**ABET**

**Self-Study Report**

**for the**

**Bachelor of Science Degree in**

****

**at**

**California State University, Los Angeles**

**Los Angeles, CA**

**July 1, 2012**

**CONFIDENTIAL**

The information supplied in this Self-Study Report is for the confidential use of ABET and its authorized agents, and will not be disclosed without authorization of the institution concerned, except for summary data not identifiable to a specific institution.

CSULAecstSTARsloganCSLAgreyOUT.ai­­­

Table of Contents

[**BACKGROUND INFORMATION 7**](#Background)

A. Contact Information 8

B. Program History 10

C. Options 10

D. Organizational Structure 10

E. Program Delivery Modes 10

F. Program Locations 10

G. Deficiencies, Weaknesses or Concerns from Previous Evaluation

and the Actions taken to Address Them 10

H. Joint Accreditation 14

[**CRITERION 1. STUDENTS 15**](#Criterion1)

A. Student Admissions 15

B. Evaluating Student Performance 16

C. Transfer Students and Transfer Courses 18

D. Advising and Career Guidance 19

E. Work in Lieu of Courses 20

F. Graduation Requirements 21

G. Transcripts of Recent Graduates 23

[**CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES 2**](#Criterion2)**4**

A. Mission Statement 24

B. Program Educational Objectives 26

C. Consistency of the Program Educational objectives with the Mission Statement 26

D. Program Constituencies 31

E. Process for Revision of the Program Educational Objectives 32

[**CRITERION 3. STUDENT OUTCOMES 3**](#Criterion3)**6**

A. Student Outcomes 36

B. Relationship of Student Outcomes to Program Educational Objectives 36

C. Process for the Establishment and Revision of the Student Outcomes 38

**[CRITERION 4. CONTINUOUS IMPROVEMENT 43](#Criterion4)**

A. Program Educational Objectives 43

B. Student Outcomes 47

B-1. Assessment measures on CSNS 48

B-2. Assessment data collection on CSNS 60

B-3. Analysis of Student Learning Outcomes 63

B-4. Relationship between Student Learning Outcomes and ABET criteria 93

C. Continuous Improvement 96

D. Additional Information 99

D-1. Assessment Reports 99

D-2. Constituent Comments 100

**[CRITERION 5. CURRICULUM 104](#Criterion5)**

A. Program Curriculum 104

A-1. Course requirements 104

A-2. Course prerequisite(s) requirements 108

A-3. Prerequisite flowchart 110

A-4. Capstone courses 111

A-5. Course coverage of required areas 112

A-6. Relationship between courses and Student Learning Outcomes 114

B. Course Syllabi and Course Journals 115

[**CRITERION 6. FACULTY 11**](#Criterion6)**7**

A. Faculty Qualifications 117

B. Faculty Workload 117

C. Faculty Size 118

D. Professional Development 119

E.. Authority and Responsibility of Faculty 120

[**CRITERION 7. FACILITIES 12**](#Criterion7)**7**

A. Offices, Classrooms and Laboratories 127

B. Computing Resources 128

C. Guidance 128

D. Maintanance and Upgrading of Facilities 129

E. Library Services 129

F. Overall Comments on Facilities 130

[**CRITERION 8. INSTITUTIONAL SUPPORT 131**](#Criterion8)

A. Leadership 131

B. Program Budget and Financial Support 131

C. Staffing 132

D. Faculty Hiring and Retention 132

E. Support of Faculty Professional Development 134

[**PROGRAM CRITERIA 1**](#PrgmCrit)**35**

[**APPENDICES 1**](#AppendixA)**36**

[Appendix A – Course Syllabi 13](#AppendixA)6

[Appendix B – Faculty Vitae 2](#AppendixB)36

[Appendix C – Equipment 2](#AppendixC)68

[Appendix D – Institutional Summary](#AppendixD) 275

[**Signature Attesting to Compliance**](#Compliance) **276**

# 

List of Figures

Figure 1.1: Articulation Agreements by Major on Assist. 19

Figure 1.2: Academic Requirements Report on GET 22

Figure 1.3: Degree Awarded on GET 23

Figure 2.1: Consistency of PEO#1 with missions of University and College 27

Figure 2.1: Consistency of PEO#2 with missions of University and College 28

Figure 2.3: Consistency of PEO#3 with missions of University and College 29

Figure 2.4: Mission-Objectives Pyramid Structure 30

Figure 2.5: Assessment Process 32

Figure 2.6: Importance of Program Educational Objectives 35

Figure 3.1: Program Educational Objectives-Student Learning Outcomes Map 38

Figure 3.2: Importance of Student Learning Outcomes 42

Figure 4.1: Satisfaction of Program Educational objective #1 45

Figure 4.2: Satisfaction of Program Educational Objective #2 46

Figure 4.3: Satisfaction of Program Educational Objective #3 47

Figure 4.4: Student Learning Outcomes – Measures Map 60

Figure 4.10: MFT Assessment Indicator Results (2006-2011) 63

Figure 4.11: MFT Assessment Indicator Results (Quarterly) 64

Figure 4.12: CS490 Assessment Indicator Results (Quarterly) 64

Figure 4.13: Survey Results 65

Figure 4.14: MFT Assessment Indicator Results (2006-2011) 66

Figure 4.15: MFT Assessment Indicator Results (Quarterly) 66

Figure 4.16: CS490 Assessment Indicator Results (Quarterly) 67

Figure 4.17: Survey Results 67

Figure 4.18: CS490 Assessment Indicator Results (Quarterly) 68

Figure 4.19: Survey Results 69

Figure 4.20: MFT Assessment Indicator Results (2006-2011) 70

Figure 4.21: MFT Assessment Indicator Results (Quarterly) 70

Figure 4.22: CS490 Assessment Indicator Results (Quarterly) 71

Figure 4.23: Survey Results 71

Figure 4.24: MFT Median Results 73

Figure 4.25: CS490 Results 73

Figure 4.26: MFT-GPA Correlation Results 74

Figure 4.27: CS337 Skill Evaluation 75

Figure 4.28: CS491/CS496 Skill Evaluation 75

Figure 4.29: CS337 Rubric Evaluation (SE-Requirements) 76

Figure 4.30: CS496A Rubric Evaluation (SE-Requirements) 76

Figure 4.31: CS437 Rubric Evaluation (SE-Design) 77

Figure 4.32: CS496B Rubric Evaluation (SE-Design) 77

Figure 4.33: Survey Results 78

Figure 4.34: CS437 Skill Evaluation 79

Figure 4.35: CS491B Skill Evaluation 79

Figure 4.36: CS437 Rubric Evaluation (SE-Team Work) 80

Figure 4.37: CS496C Rubric Evaluation (SE- Team Work) 80

Figure 4.38: CS437 Rubric Evaluation (SE- Implementation) 81

Figure 4.39: CS496C Rubric Evaluation (SE- Implementation) 81

Figure 4.40: Survey Results 82

Figure 4.41: CSCS337/437 Skill Evaluation 83

Figure 4.42: CS491/CS496 Skill Evaluation 83

Figure 4.43: CS337/CS437 Skill Evaluation 84

Figure 4.44: CS491/CS496 Skill Evaluation 84

Figure 4.45: CS437 Rubric Evaluation (Oral Communication) 85

Figure 4.46: CS496C Rubric Evaluation (Oral Communication) 85

Figure 4.47: CS437 Rubric Evaluation (Written Communication) 86

Figure 4.48: CS496C Rubric Evaluation (Written Implementation) 86

Figure 4.49: Survey Results 87

Figure 4.50: TECH250 Evaluation 88

Figure 4.51: Survey Results 88

Figure 4.52: CS301 Evaluation 90

Figure 4.53: TECH250 Evaluation 90

Figure 4.54: Survey Results 91

Figure 4.55: CS101 Evaluation 92

Figure 4.56: CS301 Evaluation 92

Figure 4.57: Survey Results 93

Figure 4.58: ABET criteria – Student Learning Outcomes Map 94

Figure 5.1: Flow Chart of required courses 110

List of Tables

Table 0.1: Summary of activities 9

Table 1.1: Grading Symbols 17

Table 2.1: Program Educational Objectives review timeline 34

Table 3.1: Student Learning Outcomes review timeline 40

Table 4.1: Skill Evaluations 51

Table 4.2: Team Work Rubric 52

Table 4.3: Oral Communication scoring rubric 53

Table 4.4: Written Communication Scoring Rubric 54

Table 4.5: Software Engineering - Requirements 55

Table 4.6: Software Engineering – Design 56

Table4.7: Software Engineering – Implementation 57

Table 4.8: Course Assignments 59

Table 4.9: Data Collection 62

Table 4.10: ABET criteria – Student Learning Outcomes Map 95

Table 4.11: Examples of Continuous Improvement 99

Table 5.1: Curriculum (Required courses) 106

Table 5.2: Curriculum (Electives) 107

Table 5.3: Curriculum (with prerequisites) 109

Table 5.4 Student Learning Outcomes: Courses Map 115

Table 6-1: Faculty Qualifications 123

Table 6-2: Faculty Workload Summary 125

Table 6.3: Course Coordinators 126

Table A.1: Courses Table 137

Table B.1: Faculty Table 236

Table C.1: Computer Laboratory Classrooms 269

Table C.2: Mediasite Classrooms 269

Table C.3: Servers 271

Table C.4: Computer Science Software 272

Table C.5: Five Year Maintenance Plan 274

# BACKGROUND INFORMATION

1. **Contact Information**

* **Raj S Pamula**

Professor and Chair

Department of Computer Science

5151 State University Drive

Los Angeles, CA 90032

Phone: 323-343-6690

Fax: 323-343-6672

e-mail: [rpamula@calstatela.edu](mailto:rpamula@calstatela.edu)

* **Chengyu Sun**

Associate Professor and Assessment Coordinator

Department of Computer Science

5151 State University Drive

Los Angeles, CA 90032

Phone: 323-343-6690

e-mail: [csun@calstatela.edu](mailto:csun@calstatela.edu)

* **Russ Abbott**

Professor and Assessment Coordinator

Department of Computer Science

5151 State University Drive

Los Angeles, CA 90032

Phone: 323-343-6690

e-mail: [rabbott@calstatela.edu](mailto:rabbott@calstatela.edu)

* **Benjamin Lee**

Faculty Director & ABET coordinator

College of Engineering, Computer Science and Technology

5151 State University Drive

Los Angeles, CA 90032

Phone: 323-343-4450

e-mail: [blee10@calstatela.edu](mailto:blee10@calstatela.edu)

* **Keith Moo-Young**

Dean

College of Engineering, Computer Science and Technology

5151 State University Drive

Los Angeles, CA 90032

Phone: 323-343-4450

e-mail: [kmooyou@calstatela.edu](mailto:kmooyou@calstatela.edu)

**Additional information:**

* Department website

<http://cs.calstatela.edu>

* Department assessment information

<http://csns.calstatela.edu/wiki/content/assessment/>

* College website

<http://www.calstatela.edu/academic/ecst/>

* University website

<http://www.calstatela.edu>

1. **Program History**

The undergraduate Bachelor of Science Computer Science degree program was housed in the Department of Mathematics and Computer Science until Fall 2001, when the Department of Computer Science was formed. The following is a brief chronology of the evolution of the Department of Computer Science from 2001-2006,

* Three major revisions to the undergraduate curriculum (CS BS) were implemented in 2002, 2003 and 2005.
* A new graduate program (CS MS) was implemented in 2003.
* The undergraduate (CS BS) received its first ABET accreditation after the ABET visit in October 2006.

The history of activities that have taken place in the Department since the last ABET accreditation visit in Fall 2006 are summarized below in Table 0.1.

| **Quarter/Year** | **Notes** |
| --- | --- |
| Fall 2006 | CSNetwork Services, or CSNS, was developed by Dr. Sun and introduced in a number of courses as a basic Learning Management System. CSNS is a web-based system that provides a number of services to facilitate teaching, learning, and student interactions. Information about CSNS is described under Criterion 4. |
| Winter 2008 | Major undergraduate program modifications were implemented as a result of the Department’s assessment process. They included the following: (i) Modification of CS245 (ii) Modification of CS332 (iii) Addition of CS337 (iv) Modification of CS491 (v) Modification of Physics requirements (vi) Modification of General Education requirements. (See Report2008 - Assessment Report, described in Section D under Criterion 4.) |
| Fall 2008 | A more detailed assessment process was implemented with the addition of assessment components to CSNS. (See Report2010 -Assessment Report described in Section D under Criterion 4.) |
| Spring 2009 | A Blended BS/MS program in computer science was implemented. It provides an accelerated route for academically superior upper-division students in the BS degree program to complete the MS degree program while simultaneously completing the BS requirements. |
| Fall 2010 | Undergraduate program modifications were implemented as a result of the Department’s assessment process. CS491AB was replaced with CS496ABC. |
| 2009-2012 | The course modifications were implemented for CS447 (Computer Networks Configuration and Management ) and CS450 (Computer Graphics).  The following new courses were added, increasing the number of choices for electives: CS454 (3D Computer Game Programming) CS454 (User Interfaces and Interactions), CS454 (Mobile Computing), CS470 (Computer Network Protocols), CS480 (Cryptography and Information Security). |
| 2010-2011 | A new Bachelor of Arts was proposed. It is currently undergoing consultation within the university. This program would be ABET accreditable under the Information Technology Criteria. |
| 2011-2012 | Preparations were completed for the next scheduled ABET re-accreditation visit in Fall 2012. |

**Table 0.1: Summary of activities**

1. **Options**

The Computer Science undergraduate program (CS BS) has no options, tracks, or concentrations.

1. **Organizational Structure**

The organization charts for the University President, Provost and Vice President for Academic Affairs Division, Academic Affairs Management Group, and College of Engineering, Computer Science and Technology are described in Section 3 of Appendix D.

1. **Program Delivery Modes**

The delivery mode used by the undergraduate Computer Science program (CS BS) at California State University, Los Angeles is the traditional on-campus lecture/laboratory mode. Courses are scheduled throughout the day on both weekdays and weekends. A significant number of courses in the program use CSNS as a learning management system to supplement the traditional teaching methods. Some courses are videotaped to provide a virtual classroom for students to watch at their convenience.

1. **Program Locations**

.

Undergraduate Computer Science program (CS BS) is offered only at Los Angeles campus.

1. **Deficiencies, Weaknesses or Concerns from Previous Evaluation(s) and the Actions Taken to Address Them**

The ABET final Statement after the 2006 visit had not indicated any department deficiencies or weaknesses. However, there were four concerns and three observations, all of which are addressed below.

***Concern #1:*** *(Standard I-3). Only two courses, CS437 and CS491AB contain direct measures of program outcomes. This can limit the ability of the program to obtain meaningful data relative to the achievement of outcomes.*

Corrective action: In Fall 2008, our Department adopted a more detailed assessment process after developing Computer Science Network Services (CSNS). The assessment measures now contain a number of direct measures with well-defined data collection techniques on CSNS. Many direct measures are employed in a number of courses (CS101, CS301, CS337, CS437, CS490, CS496A, and CS496C). Each outcome is now assessed through many measures in these courses. Direct measures include skill evaluations, assignments, exams, and grading rubrics. [See Section B-1 under Criterion 4, Tables 4.1-4.9]

Skill Evaluation is one of the direct measures that allows faculty to directly observe a student's demonstration of a particular skill. Prior to 2011, we employed a summative assessment of skills on a 5 point scale. This practice was modified after the faculty in the Assessment Committee attended an ABET sponsored Faculty Workshop on Sustainable Assessment Processes. The new process now defines performance indicators backed by well-defined rubrics that define the expectation of students [See Tables 4.2-4.9]. The rubrics were developed in Spring 2011 and employed in the 2011-2012 assessment cycle.

CSNS thus provides support to collect assessment data from various measures both at the course level and at the program level. By building assessment elements into everyday teaching and learning, we ensure that program assessment is a rigorous, systematic, and a continuous process. These improved processes are described in Section 4.

Effective Date: Fall 2008 (refined in Spring 2011)

***Concern #2:*** *(Standard IV6) Although core materials do provide basic coverage of computer organization in EE444,such late coverage of computer organization may affect students’ ability to gain a full conceptual understanding of other areas of computer science that depend on knowledge of computer organization, such as programming languages, data structures, and algorithms.*

Corrective action: CS245, a lower division core requirement, has been modified from “Using Operating Systems and Networks: to “Introduction to Computer Organization, Operating Systems and Networks”. Topics in CS245 that address Computer Organization now include the following:

* Introduction to Computer Organization and Assembly language
* Binary arithmetic & data types
* Logic operations, Boolean algebra
* Hardware basics: transistors, logic gates.
* Logic circuits & Memory
* von Neumann model & the Instruction cycle.

In addition, Computer Organization is introduced in CS101. Topical coverage includes Hardware evolution and von Neumann computer architecture model.

Thus Computer Organization is currently covered at three levels: CS101, CS245 and EE444.

As described in Section C under Criterion 4, these actions have contributed to the improved results in the evaluation of Student Learning Outcome #4.

Effective Date: Winter 2008

***Concern #3:*** *(Standard IV-7) Although core materials do provide some coverage of theoretical foundations in some courses, the teaching of CS386 late in the curriculum may affect students ability to gain a full conceptual understanding of areas of computer science that would benefit from knowledge of theoretical foundations.*

In our ABET 2006 ABET Self Study, we misread the standard “theoretical foundations” as only addressing “computer theory” pertaining to abstract computing that is described by a course in “automata theory and formal languages”. As a result, our Self Study report included CS386 as the only course covering the standard for “theoretical foundations”.

During the ABET 2006 visit, we found out that this standard is much broader and includes theoretical foundations of all other computing topics that is typically covered throughout the curriculum.

Theoretical foundations are covered in a broad number of core courses as described below:

* Fundamentals of computer science (CS101)
* Database systems (CS122)
* Programming fundamentals and Object oriented design (CS201-CS203)
* Discrete mathematics and discrete structures (MATH248)
* Linear algebra and Matrix theory (MATH255)
* Computer organization (CS245, EE444)
* Algorithm analysis (CS312)
* Theory and structure of programming languages (CS 332F and CS 332L)
* Automata theory (CS386)
* Programming languages (CS201-CS203, CS332FLC)
* Operating systems (CS245, CS440)
* Software design (CS337)

Effective Date: Spring 2007

***Concern #4:*** *(Standard IV-9) A large number of programming-oriented courses in the core provides a limited foundation for students to fully build on the core for depth in advanced course work.*

Corrective action: (i) Modification of CS245, (ii) Expanding a one quarter CS437 course into a two-quarter CS337-CS437 sequence, (iii) Expanding a two-quarter CS491AB sequence into a three-quarter CS496ABC sequence.

As indicated by the flow chart of program requirements in Figure 5.1, the courses are structured over many levels and indicated as rows of requirements. Beginning lower division programming-oriented courses (CS201-CS203) and beginning systems courses (CS120, 122, CS245) are marked by Row1 to Row3.

The advanced required CS courses are indicated in Row4-Row7. These consist of upper division CS3xx and CS4xx level classes that build on the above mentioned beginning classes for depth in advanced course work.

The advanced requirements culminate in two capstone courses: CS490 and CS496ABC. Students in CS496ABC build on their core knowledge and implement a large group project under the supervision of an instructor. This is a three-quarter laboratory project sequence usually taken during their senior year. Students in CS490 take a standardized Major Field Test (resembling the Computer Science subject test in the Graduate Record Exam) to measure the critical knowledge and understanding obtained by students in the core areas of Computer Science.

The current curriculum lays a solid foundation in the field of Computer Science which is wide and deep. Students gain additional exposure to advanced topics by taking 24 units (six upper division courses) of upper division electives. These electives would be chosen from a variety of courses as listed in Table 5.2. In addition, Special Topics elective courses are introduced almost every year. The electives are to be taken during Row4 to Row 7 (see Figure 5.1) and will help solidify the depth and breadth of the student’s undergraduate experience.

The course requirements to meet this Standard are described in detail in Section A under Criterion 5.

Effective Date: Winter 2008 (CS 337, CS437, CS491) and Fall 2010 (CS496).

***Observation #1:*** *The department might want to consider increasing the number of credit hours for CS491AB to ensure that the student gets enough credit hours for implementing the entire project individually using the software life cycle. The number of credit hours in CS 491AB (2+2) seems to be inconsistent with the five credit hours in CS437 where the software lifecycle is being taught.*

Action: CS491AB (2+2) was changed to CS491AB (3+3) in Winter2008 as a result of this observation. This two-quarter senior design project sequence was restructured again to three-quarter CS496 (2+2+2) sequence in Fall 2010. As a result of these changes, students devote extra units and extra time to complete their projects. Many of the projects are sponsored by industry. (See Section C#1 under Criterion 4).

Effective Date: Winter 2008 (CS491) and Fall 2010 (CS496)

***Observation #2:*** *The department may wish to consider offering a group project in one of CS491AB that will offer opportunities to students to use their skills from CS437 to work on a real life project to work as a team.*

Action: In the newly designed CS496ABC (modification of CS491AB) sequence, all the projects are group projects. Most are sponsored by industry. Students work in groups on yearlong real-life projects. (See CS496 projects at <http://csns.calstatela.edu/projects.html>). CS437 was modified and expanded into a two quarter CS337-CS437 sequence. CS337-CS437 also requires a group project.

CS337 was made a pre-requisite to CS496A. This modification made a positive impact as the students, working as a team, use their skills from CS337 to work on yearlong, industry sponsored projects in CS496.

Effective Date: Fall 2010

***Observation #3:*** *Although Standard IV-13 is satisfied, the department might consider adopting the new sequence of physics courses that have been introduced in fall quarter 2006, as these courses seem to be more appropriate for computer science students.*

Action: The undergraduate program was modified to allow for the possibility of any three quarter lab sequence PHYS 101-103 (12 units) or PHYS 211-213 (15 units). We agree that the PHYS 211-213 sequence using the calculus approach may be more suitable than the analytical approach used in PHYS101-103 sequence for our students. The proposed solution would give students the choice of either PHYS sequence/approach based on their interest. It also addresses the problem with transfer students who can now continue with the sequence they have started with at a transfer institution.

Effective Date: Winter 2008

1. **Joint Accreditation**

The Computer Science undergraduate program (CS BS) is not jointly accredited nor is it seeking joint accreditation by any commission other than ABET Computing Accreditation Commission (CAC).

**GENERAL CRITERIA**

# CRITERION 1. STUDENTS

1. **Student Admissions**

Requirements for admission to California State University (CSU), Los Angeles is available at CSUMentor (<http://www.csumentor.edu/>) an online resource designed to help students and their families learn about the CSU system, select a CSU campus to attend, plan to finance higher education, and apply for admission. Admissions information is also available on Golden Eagle Territory (GET, <https://get.calstatela.edu/>), a student portal used at California State University, Los Angeles.

Typically students are admitted in the following three categories: (i) First time freshmen after graduating from a High School (ii) Upper division transfer student from a 2-year or a 4-year College. (iii) Lower division transfer student from a 2-year or a 4-year College. A brief summary of admission requirements for each of the categories is indicated below:

1. Criteria for admission of first-time freshmen are established in compliance with a California State mandate that CSU institutions admit the top one-third of California high school graduates. To qualify for admission as a first-time freshman, an applicant must graduate from High School and meet the following standards in each of the following areas:

* Specific high school courses
  + The CSU requires a minimum 15-unit pattern of courses (Math, Science, English, History, and other college preparatory classes) for admission as a first-time freshman. Each unit is equal to a year of study in a subject area. A grade of C or higher is required for each course used to meet any subject requirement.)
* Grades in specified courses and test scores
  + Grades earned in high school are the most important factor in CSU admission decisions. Test scores are required unless a student has a grade point average above 3.0 andis a resident of California. The CSU uses a calculation called an eligibility index that combines a student’s high school grade point average with the score the student earns on either the SAT or ACT tests. (To calculate the eligibility index multiply the grade point average by 800 and add the combined score on the SAT, using the combined scores earned on the critical reading and math sections. The CSU does not use the SAT Writing score for admission purposes. For the ACT, multiply the grade point average by 200 and add ten times the ACT composite score.) A California high school graduate (or a resident of California for tuition purposes), needs a minimum eligibility index of 2900 using the SAT combined score for critical reading and math sections or 694 using the ACT.

1. Criteria for admission as an upper division transfer student into the CSU system considers the following three factors:

* College grade requirements
  + Students must have a minimum of 90 quarter units. The overall grade point average must be at least 2.0 (2.40 for California nonresidents.) The GPA is calculated using all transfer units attempted.
* College coursework completed
  + Students need to complete a minimum of 30 semester or 45 quarter units of the General Education Breath Requirement or the Intersegmental General Education Transfer Curriculum. Within either pattern, the highest-priority classes are the three courses in the English language - oral communications, English composition and critical thinking - along with a general education course in mathematics.
* Good standing at the last college or university attended
  + Students must be in good standing at the prior college. In simple terms, good standing means students are eligible to reenroll at the most recently attended college or university.

1. Criteria for admission as a lower division transfer (less than 90 quarter units in transferable work) student into the CSU system considers the following three factors (The evaluation is a combination of the factors described in I and II above):

* High school requirements
* College courses and grades,
* Good standing at the last college or university attended

1. **Evaluating Student Performance**

Students are continuously evaluated from the time they apply for admission to the University until they graduate. These evaluations occur in every situation in which a student must attain a satisfactory ("passing") result to progress toward graduation.

Faculty members are primary evaluators of students in the University. These faculty generally rely on standard evaluation tools such as exams, quizzes, homework assignments, design projects, computer assignments, oral presentations, written assignments, and a University-required final examination to measure student performance in their courses. The ongoing evaluation of students becomes integrated throughout the curriculum and decentralized by virtue of the fact that each professor chooses his or her own ways to evaluate students (except, of course, for the mandatory final exam).

Student performance is evaluated by a grade, as indicated in Table 1.1, in each course.

| **Grade Symbol** | **Explanation** | **Grade Points Earned** |
| --- | --- | --- |
| A | Superior | 4.0 per unit value of course |
| A- | Outstanding | 3.7 per unit value of course |
| B+ | Very good | 3.3 per unit value of course |
| B | Good | 3.0 per unit value of course |
| B- | Better than average | 2.7 per unit value of course |
| C+ | Above average | 2.3 per unit value of course |
| C | Average | 2.0 per unit value of course |
| C- | Below average | 1.7 per unit value of course |
| D+ | Weak | 1.3 per unit value of course |
| D | Poor | 1.0 per unit value of course |
| D- | Barely passing | 0.7 per unit value of course |
| F | Nonattainment | 0.0 per unit value of course |

**Table 1.1: Grading Symbols**

The Golden Eagle Territory (<https://get.calstatela.edu/>) provides access to services and information tailored to each student. It offers registration and other online services such as student account and transcript information. Prerequisites are strictly enforced on GET. It is very uncommon that this is overridden. The GET registration system will prevent students from registering for a class without meeting prerequisite(s). The prerequisites can be met by any articulated transfer course. If the student has met the prerequisite(s) by any non-articulated transfer course, he/she must meet with his/her advisor. Once the prerequisite course is verified, appropriate paperwork is sent to the evaluator office to indicate the course articulation which would then be reflected on GET for the student. The student would then be permitted to register for the class.

As students progress through the Computer Science program, they are monitored in a number of ways. First, the University Office of the Registrar flags any student who fails to maintain "good academic standing" (determined by the criterion that all students must maintain a minimum overall GPA of 2.0). Students whose GPA drops below 2.0 are automatically placed on academic probation. A letter is sent by the University to all students on probation requiring them to meet with their faculty advisor to discuss a remedial course of action. Students whose grade point deficiency becomes excessive are disqualified and can only be reinstated with the approval of the Department chair.

Finally, all students are formally evaluated two quarters before their anticipated graduation. The pre-graduation evaluation process ensures that students have satisfied all curricular requirements and have achieved at least a 2.0 GPA in the following four areas: (1) all units attempted (including units accepted as transfer units); (2) all units attempted at CSULA; (3) all courses used to meet the General Education requirements; and (4) all courses required for the major. (See <http://ecatalog.calstatela.edu/>, Procedures and Regulations).

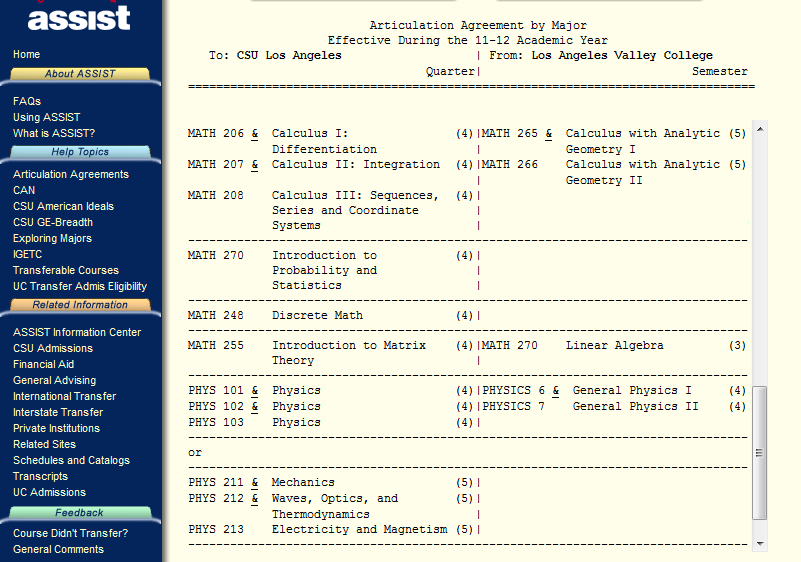
1. **Transfer Students and Transfer Courses**

Undergraduate transfer students’ previous college work is evaluated in terms of its relevance to Cal State L.A. course offerings and degree requirements. This evaluation identifies general education and graduation course requirements met by transfer courses and credit to be accepted in satisfaction of unit requirements. The policies of the transfer institution(s) are followed when grade point averages of transfer course work are computed. If such policies cannot be determined, Cal State L.A. policies are followed. Established articulation agreements are followed when evaluating college transcripts.

For California Community Colleges, comparable course-to-course articulation agreements are adjudicated by the Department Chair for lower division major preparation. General Education courses from the California Community Colleges are articulated through system-wide agreements. Grades of “C minus” or higher are transferable. All comparable course articulation for any of the California Community Colleges is available at <http://www.assist.org>, an online student-transfer information system that shows how course credits earned at one public California college or university can be applied when transferred to another. ASSIST is the official repository of articulation for California’s public colleges and universities and provides the most accurate and up-to-date information about student transfer in California.

A snapshot of articulated courses between Los Angeles Valley College and California State University for the Computer Science major is shown in Figure 1.1.

Upon admission, evaluators in the Admissions Office post all articulated courses on student records on GET. To transfer a non-articulated course from a college, the academic advisor evaluates the transfer course by examining the course syllabus and course descriptions. Once the course is verified, appropriate paperwork is sent to the evaluator office to indicate the course articulation, which would then be reflected on GET.



**Figure 1.1: Articulation Agreements by Major on Assist.**

1. **Advising and Career Guidance**

Admitted freshmen and transfer students attend university-sponsored advising sessions, which are attended by the Department chair and program advisors. Students are informed of the degree requirements, course pre-requisites, laboratory access, computer-related student clubs, and advising process. Students are given information about the program requirements that are posted on the online university catalog and the Department website. Students are also given information in the form of a Undergraduate Student Handbook, which is also posted on the Department website.

For a comprehensive overview of Higher Education every student is required to take CS101 within the first two quarters. Topics covered include : University rules and regulations; General Education requirements; Major requirements; Evaluate transfer units; Sample quarterly planners; Individualized quarterly planners; Graduation checks; Each student is required to present an individualized quarterly planner and to get feedback from the instructor. At the end of the course, students complete a self-evaluation and formulate a quarterly planner to complete all the remaining requirements.

Student advising is facilitated in a number of ways:

* Advisor/staff office visitation: The advisor’s and the Department chair’s office hours are posted in the Department office every quarter. Students meet with the faculty advisor to evaluate class work to date, to discuss issues (if any) impacting their present load, to resolve any GPA issues, and to plan subsequent classes. The advisor and student go over the student’s degree progress using GET and CSNS.
* Email advisor: Students can seek to get clarifications from the advisor or staff concerning any particular queries. Email is often used by students who need immediate clarification.
* Forums on Advisement on CSNS: Various topics exist that help clarify generic questions posed by the students. These are replied to either by other students or the advisor and are monitored by the Department staff and advisor.
* Students and faculty often communicate on discussion pages of the CS wiki.
* Students can get general advise from the Department staff who is very knowledgeable about course prerequisites, curricular requirements, course scheduling, and university regulations. The staff has access to GET and CSNS systems.
* The College advisement center provides an array of services to incoming freshman, and second year students in all majors.  These services include: academic advisement through professional staff and peer advisors, the engagement of freshman in learning communities and academic excellence workshops, and assistance with scholarship and internship opportunities.

Students are encouraged to seek the advice of any faculty in evaluating career choices. Students get information regarding career choices from job fairs (conducted by the college as a part of ECST week activities and a few times each year by the University Career Center) and guest speakers (conducted by ACM student chapter and ECST).

1. **Work in Lieu of Courses**

Credit for certain nontraditional learning experiences may be used in determining eligibility and credit allowances of undergraduate applicants. These nontraditional credit allowances are listed and described below. A more detailed analysis of each category appears in the catalog (<http://ecatalog.calstatela.edu/>) under Evaluation of Transfer Credit.

#### Categories and Credit Allowances:

* Baccalaureate-level course credit certified by the Defense Activities in Nontraditional Education Support (DANTES) yields a maximum of 36-quarter units.
* Advanced Placement examinations of the CEEB with scores of 3, 4, or 5 yield 9-quarter units for each examination.
* For credit allowance for standardized external examinations, including CLEP, Advanced Placement, the CSU English Equivalency Exam, the American Chemical Society Cooperative Examination, see the explanations later in this chapter.
* For basic training military service, 9 quarter units of lower division elective credit are awarded for 1 year or longer of active duty with an honorable discharge, and 4½ quarter units for 6 months to one year with an honorable discharge.
* Civilian and military courses and schools recommended for credit by the American Council on Education’s Commission on Educational Credit and Credentials yield a maximum of 12 quarter units.

Students can also request a “Credit by Exam” to show competency and get course credit. In the Computer Science Department, this process is very rare and is only allowed if the advisor is relatively certain that the student is competent in the material in beginning lower division Computer Science courses. The student is given a comprehensive exam covering the course material and the grade is recorded on the GET system.

1. **Graduation Requirements**

Certifying that a student has met all graduation requirements, and therefore has complied with ABET criteria (both general and program specific), is a simple but functional process as described below:

* + - * 1. Students initiate this process by filling out an Application for Graduation (degree check) on a form available at the Cal State L.A. Graduation Office website, academic department/division/school, and Enrollment Services in Administration 146. These forms are available five days prior to the application filing period. (Filing periods are published in the Graduation information section of the Schedule of Classes.)
        2. The application is to be filed two quarters prior to the end of the term of their expected graduation. Students are able to access their Academic Requirements Report at any time thorough their college career. The Academic Requirements Report (an audit report generated on GET) gives a visually clear picture of the requirements; completed requirements; transfer credits etc. as shown in Figure 1.2. Students thus have a general idea of the graduating quarter and discuss the program requirements with the advisor.
        3. Students submit their completed application to the Department for processing. The undergraduate faculty advisor discusses the Academic Progress Report with the student which indicates clearly all the requirements that have been completed and flags the remaining requirements. The student and advisor draw up a plan to meet the remaining requirements in the coming two quarters.



**Figure 1.2: Academic Requirements Report on GET**

* + - * 1. The Department staff updates the standing of the student to Graduation Check standing on CSNS. This allows the staff to check on the student’s remaining requirements in the coming quarters. The Department staff then forwards the graduation application to the Graduation Office for an official audit.
        2. The Graduation Office of the University Registrar’s Office has the sole authority to audit and certify that a candidate for graduation has fulfilled her/his approved program requirements. Graduation check audit results are sent by surface mail to the students prior to their final anticipated quarter. Students who are enrolled in the quarter they expect to graduate but do not complete all degree requirements will have their graduation application automatically transferred to the following quarter.
        3. Students who do not finish their requirements by the end of the second quarter must reapply and restart the graduation application procedure.
        4. Degree dates are posted at the end of the quarter in which all requirements are met. On completion of the degree requirements, the transcripts on GET indicate the following:

Degree : Bachelor of Science

Confer Date : xxxx-xx-xx

Plan : Computer Science

An example is shown below in Figure 1.3



**Figure 1.3: Degree Awarded on GET**

* + - * 1. After the degree is posted on GET, CSNS imports the list of all graduated students and updates their standing to an alumni standing.

1. **Transcripts of Recent Graduates**

A random selection of transcripts will be provided to the visiting ABET team prior or during the site visit as requested by the Team Chair. Academic Requirements Reports for those students can be generated at the time of the visit.

# CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES

1. **Mission Statement**

**A-1.Mission Statement and Institutional Learning Goals of the California State University, Los Angeles**

Mission Statement

Cal State L.A., a member of the California State University (CSU) system, offers excellent and innovative educational opportunities to an urban student population that reflects the diversity of the Los Angeles basin, including:

* Preparing students to appreciate, engage, enhance and transform the social, cultural, civic, and workplace structures of American and global societies;
* Providing students with the capabilities, skills, and opportunities to take full advantage of life-long learning, including graduate and professional studies, and opportunities to participate in research, scholarly, and creative activities;
* Offering students tools for personal and academic achievement, economic mobility, and healthier lives;
* Serving as a gateway among the Cal State L.A. community, the greater Los Angeles community, and world community for shared educational and cultural life;
* Providing high quality professional services to all constituencies of the University.

Institutional Learning Goals

California State University, Los Angeles students expand and deepen their interdisciplinary and general understanding of the world, enhance their critical skills, and take responsibility for a lifetime of learning, and as graduates become individuals who engage, enhance, and contribute to democratic society. The following are the institutional learning goals of CSULA:

* *Knowledge: Mastery of content and processes of inquiry*  
  CSULA graduates have a strong knowledge base in their academic major and can use powerful processes of inquiry in a range of disciplines. They engage contemporary and enduring questions with an understanding of the complexities of human cultures and the physical and natural world and are ready to put their knowledge into action to address contemporary issues.
* *Proficiency: Intellectual skills*CSULA graduates are equipped to actively participate in democratic society. They are critical thinkers who make use of quantitative and qualitative reasoning. They have the ability to find, use, evaluate and process information in order to engage in complex decision-making. They read critically, speak and write clearly and thoughtfully and communicate effectively.
* *Place and Community: Urban and global mission*  
  CSULA graduates are engaged individuals who have contributed to the multi-lingual and multiethnic communities that constitute Los Angeles and the world of the future. They are aware of how their actions impact society and the environment, and they strive to make socially responsible decisions. They are community builders sensitive to the needs of diverse individuals and groups and committed to renewing the communities in which they live.
* *Transformation: Integrative learning*   
  CSULA graduates integrate academic learning with life. They engage in community, professional, creative, research and scholarly projects that lead to changes in their sense of self and understanding of their worlds. Graduates integrate their knowledge, skills and experience to address complex and contemporary issues and act ethically as leaders for the 21st century.

**A-2.Mission Statement and Goals of the College of Engineering, Computer Science, and Technology**

Mission Statement

*To provide our students with innovative learning experiences and service opportunities and to graduate well educated professionals who are prepared to meet the challenges of a rapidly changing world.*

Goals

* *To design a world-class curriculum enabling our students for life long learning and adapting to an ever changing technological environment,*
* *To become the most responsive graduate engineering, computer science, and technology program in Los Angeles that meets the needs of the working professional,*
* *To demonstrate internationally recognized leadership in applied research, advance prototyping, and design in strategic areas that are fully integrated into the bachelor’s and master’s educational programs,*
* *To humanize engineering, computer science, and technology through our community service and globalization activities that enable our engineering community to reflect the 21st century workforce,*
* *To be consistently ranked as one of the top 20 Engineering programs in the Nation.*

**A-3. Mission Statement of the Department of Computer Science**

*To graduate well educated computer scientists who are prepared to meet the challenges of a rapidly changing, increasingly complex world.*

This mission is concise and supports the University and College’s missions.

1. **Program Educational Objectives**

Program Educational Objectives are broad statements that will describe what graduates are expected to attain within a few years of graduation. The Program Educational Objectives provide the curricular guidelines with respect to the program.

The existing Program Educational Objectives of the undergraduate program in Computer Science at California State University, Los Angeles are:

1. *Students who enter the workforce will have established themselves as effective professionals by having solved real problems through the use of their computer science knowledge and their communication, critical thinking, and problem solving skills.*
2. *Students who continue in academia will have been successful in pursuing advanced degrees and in demonstrating their ability to master advanced areas of computer science.*
3. *Students will have demonstrated their ability to adapt to a rapidly changing environment by having learned and applied new skills and new technologies.*

The Program Educational Objectives are published at:

1. Department web site <http://www.calstatela.edu/academic/ecst/cs/objectives.php>
2. Program assessment documentation web site http://csns.calstatela.edu/wiki/content/assessment/undergrad/
3. Computer Science Department office and all faculty offices.
4. **Consistency of the Program Educational Objectives with the Mission Statement**

The mission of the Department supports those of the College and the University. The Program Educational Objectives are consistent with the mission statements of the department, college and the university. In particular:

The **first Program Educational Objective** “*Students who enter the workforce will have established themselves as effective professionals… “*  is consistent with Department’s Mission statement “*To graduate well educated computer scientists.. “,*  the College’s Mission statement “*graduate well educated professionals…”*  and the University Mission Statement “*Offering students tools for personal and academic achievement, economic mobility, and healthier lives*”.

The **second Program Educational Objective** “*Students who continue in academia will have been successful in pursuing advanced degrees ...“* is consistent with Department’s Mission statement “…*who are prepared to meet the challenges..”,*  the College’s Mission statement “…*who are prepared to meet the challenges…”*  and the University Mission Statement *“Providing students with the capabilities, skills, and opportunities to take full advantage of life-long learning, including graduate and professional studies, and opportunities to participate in research, scholarly, and creative activities*”.

The **third Program Educational Objective** “*Students will have demonstrated their ability to adapt to a rapidly changing environment…“* is consistent with Department’s Mission statement “..*meet the challenges of a rapidly changing..”,*  the College’s Mission statement “*meet the challenges of a rapidly changing…”*  and the University Mission Statement *“Preparing students to appreciate, engage, enhance and transform the social, cultural, civic, and workplace structures of American and global societies”.*

The relationships described in Figures 2.1-2.3 further indicate how the University and College mission statements are consistent to the established Program Educational objectives of BSCS program.

**University's Mission:**

Offering students tools for personal and academic achievement, economic mobility, and healthier lives

**Specific goals:**

* *Proficiency: Intellectual skills*CSULA graduates are equipped to actively participate in democratic society. They are critical thinkers who make use of quantitative and qualitative reasoning. They have the ability to find, use, evaluate and process information in order to engage in complex decision-making. They read critically, speak and write clearly and thoughtfully and communicate effectively.
* *Knowledge: Mastery of content and processes of inquiry*  
  CSULA graduates have a strong knowledge base in their academic major and can use powerful processes of inquiry in a range of disciplines. They engage contemporary and enduring questions with an understanding of the complexities of human cultures and the physical and natural world and are ready to put their knowledge into action to address contemporary issues.

**College's Mission:**

To provide our students with an innovative learning experiences and service opportunities and to graduate well educated professionals who are prepared to meet the challenges of a rapidly changing world.

**Specific goals:**

* To become the most responsive graduate engineering, computer science, and technology program in Los Angeles that meets the needs of the working professional,
* To humanize engineering, computer science, and technology through our community service and globalization activities that enable our engineering community to reflect the 21st century workforce.

**Program Educational Objective #1**

*Students who enter the workforce will have established themselves as effective professionals by having solved real problems through the use of their computer science knowledge and their communication, critical thinking, and problem solving skills.*

**Figure 2.1: Consistency of PEO#1 with missions of University and College**

**University's Mission:**

Providing students with the capabilities, skills, and opportunities to take full advantage of life-long learning, including graduate and professional studies, and opportunities to participate in research, scholarly, and creative activities

**Specific goals:**

* *Proficiency: Intellectual skills*  
  CSULA graduates are equipped to actively participate in democratic society. They are critical thinkers who make use of quantitative and qualitative reasoning. They have the ability to find, use, evaluate and process information in order to engage in complex decision-making. They read critically, speak and write clearly and thoughtfully and communicate effectively.
* *Knowledge: Mastery of content and processes of inquiry*  
  CSULA graduates have a strong knowledge base in their academic major and can use powerful processes of inquiry in a range of disciplines. They engage contemporary and enduring questions with an understanding of the complexities of human cultures and the physical and natural world and are ready to put their knowledge into action to address contemporary issues.

**College's Mission:**

To provide our students with an innovative learning experiences and service opportunities and to graduate well educated professionals who are prepared to meet the challenges of a rapidly changing world.

**Specific goals:**

* To become the most responsive graduate engineering, computer science, and technology program in Los Angeles that meets the needs of the working professional
* To demonstrate internationally recognized leadership in applied research, advance prototyping, and design in strategic areas that are fully integrated into the bachelor’s and master’s educational programs

**Program Educational Objective (PEO) #2**

*Students who continue in academia will have been successful in pursuing advanced degrees and in demonstrating their ability to master advanced areas of computer science*

**Figure 2.1: Consistency of PEO#2 with missions of University and College**

**University's Mission:**

Preparing students to appreciate, engage, enhance and transform the social, cultural, civic, and workplace structures of American and global societies

**Specific goals:**

* *Place and Community: Urban and global mission*  
  CSULA graduates are engaged individuals who have contributed to the multi-lingual and multiethnic communities that constitute Los Angeles and the world of the future. They are aware of how their actions impact society and the environment, and they strive to make socially responsible decisions. They are community builders sensitive to the needs of diverse individuals and groups and committed to renewing the communities in which they live.
* *Transformation: Integrative learning*   
  CSULA graduates integrate academic learning with life. They engage in community, professional, creative, research and scholarly projects that lead to changes in their sense of self and understanding of their worlds. Graduates integrate their knowledge, skills and experience to address complex and contemporary issues and act ethically as leaders for the 21st century.

**College's Mission:**

To provide our students with an innovative learning experiences and service opportunities and to graduate well educated professionals who are prepared to meet the challenges of a rapidly changing world.

**Specific goals:**

* To design a world-class curriculum enabling our students for life long learning and adapting to an ever changing technological environment,
* To humanize engineering, computer science, and technology through our community service and globalization activities that enable our engineering community to reflect the 21st century workforce.

**Program Educational Objective (PEO) #3**

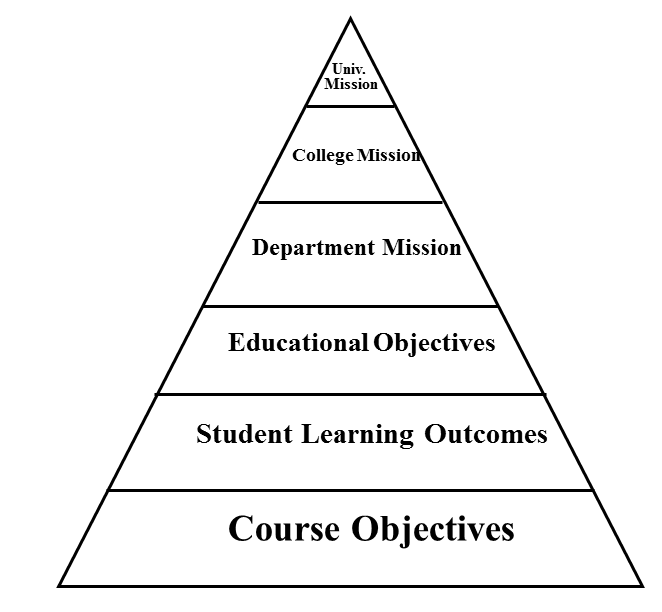
*Students will have demonstrated their ability to adapt to a rapidly changing environment by having learned and applied new skills and new technologies.*

**Figure 2.3: Consistency of PEO#3 with missions of University and College**

The University mandates a regular periodic Program Review for every degree program. During the most recent review of the Computer Science programs (June 2010), the University Program Review Committee issued the following commendation to the department:

*“For well-articulated mission, goals and objectives that align well with those of the College and University”.* (See [University Program Review Report (Spring 2010)](http://csns.calstatela.edu/download.html?fileId=2660027), posted at <http://csns.calstatela.edu/wiki/content/assessment/documents/>)

The consistency of Mission statements, Program Educational Objectives, Student Learning Outcomes, and Course Objectives can be best described by the pyramid shown in Figure 2.4. Course Objectives (described in Section A under Criterion 5) contribute to the satisfaction of Student Learning Outcomes (described in Section A under Criterion 3), which in turn contribute to the satisfaction of Program Educational Objectives (described in Section B under Criterion 2). Satisfaction of Objectives leads to the accomplishment of the Missions of the department, college, and the university.



**Figure 2.4: Mission-Objectives Pyramid Structure**

1. **Program Constituencies**

The program’s constituencies are those with a vested interest in the capabilities of our graduates. The constituencies of the undergraduate program are:

* + 1. *Students*: As prospective products of the program the current Computer Science majors represent an important constituency. They take part in the assessment of courses; their input is important for course and curriculum improvement. They also provide input through surveys, conducted at the time of graduation, which help the program assess its Student Learning Outcomes. The Student Learning Outcomes support the Program Educational Objectives, which help prospective students decide whether they wish to pursue a career in Computer Science.
    2. *Alumni*: Alumni are the product of the program. They reflect and represent the success of the program. Alumni input concerning the Program Educational Objectives is sought via surveys.
    3. *Employers*: Employers of our graduates are essential in providing input regarding the skills our students must possess to be employed. Industry representatives who are the immediate supervisors to our alumni are asked for feedback concerning Program Educational Objectives. Their guidance helps us determine if the graduates of the program will fit the needs of business and industry.
    4. *Industry Advisory Board*: The Industry Advisory Board (IAB) consists of industry leaders whose advice is critical in preparing our students for the technical and practical demands of their future employers. Some Industry Advisory Board members are also our alumni and/or employers of our graduates. This board provides feedback and suggestions for formulating our Program Educational Objectives. The IAB meets once a year (<http://csns.calstatela.edu/wiki/content/assessment/iab/>). Meetings are usually held during the Spring quarter to synchronize with the presentation of undergraduate senior design project presentations. A list of questions concerning Objectives, Learning Outcomes, and Courses is sent to the IAB ahead of the annual meeting to encourage IAB members to think about the relationship between their companies and our program. A formal survey is taken to record feedback.
    5. *Faculty*: The faculty design and deliver the undergraduate curriculum. Their responsibility is to ensure that the curriculum is updated to reflect changing technologies and the skills expected of our graduates. An annual faculty retreat <http://csns.calstatela.edu/wiki/content/assessment/retreat_presentations/> is conducted during the Winter quarter. We use the retreat as an opportunity to discuss a broad range of issues regarding the Objectives, Learning Outcomes, and Courses of the undergraduate and the graduate curriculum.

Besides these constituencies the program adheres to the following frameworks.

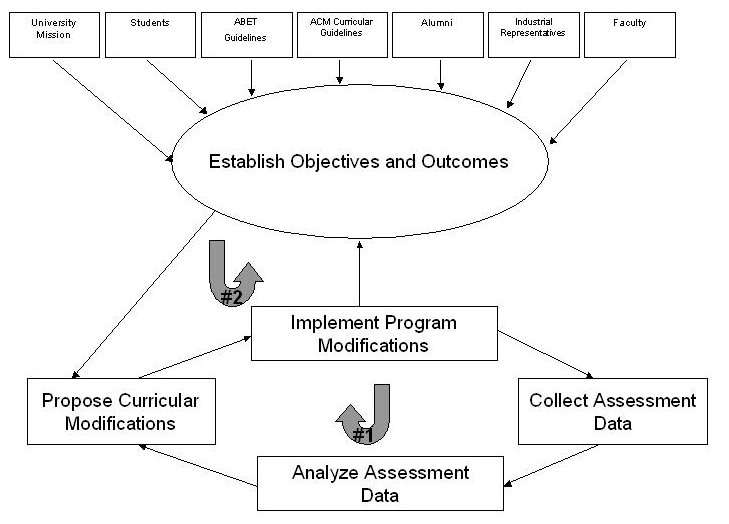
*Mission*: The missions of the university, college and the department are important constituents in establishing Program Educational Objectives and Student Learning Outcomes.

*ABET guidelines:* As the external accreditation agency, ABET assures the program’s effectiveness by setting accreditation standards. These standards provide an important input in guiding the Program Educational Objectives and Student Learning Outcomes.

*ACM curricular guidelines:* ACM/IEEE curriculum guidelines have provided important advice with respect to the computing curricula. There is an ongoing discussion on curriculum that provides input into redesigning the programs.

1. **Process for Revision of the Program Educational Objectives**

Our Department established an assessment process for the undergraduate degree program which describes the process for revision of Program Educational Objectives. As described in Figure 2.5, this assessment process is a two-loop process. The outer loop, Loop #2, shows the periodic process involved in establishing, assessing and revising the Program Educational Objectives. This process employs the feedback from various constituencies that have an interest in the outcome of the program. Loop #2 is relatively a slow loop where assessment occurs usually once every five years, or sooner based on assessment data.



**Figure 2.5: Assessment Process**

Program Educational Objectives were originally developed in the 2000-2001 academic year. These were later revised in 2004-2005 academic year and were in effect in the last ABET CAC visit in Fall 2006. Our Program Educational Objectives were revised again in 2010-2011 academic year.

The actual mechanism for revision of Program Educational Objectives is as follows:

1. Our Assessment Committee (Raj Pamula, Chengyu Sun, Russ Abbott) develops revised Program Educational Objectives by considering the following input:
   1. Existing Program Educational Objectives and Student Learning Outcomes
   2. Mission statements (from the University, College and Department)
   3. ACM (Association of Computing Machinery) curricular guidelines
   4. ABET accreditation standards.
   5. Annual feedback from surveys
      1. Comments from the constituencies (Alumni, Employers, Faculty and IAB) from annual surveys (conducted during Loop #1 described in Figure 2.5) to an open-ended question: *“enter any suggestions that you think would improve the program”*
   6. Annual discussions from IAB meetings
      1. Comments from a broad range of discussion concerning Objectives, Learning Outcomes and Courses
      2. Responses to questions sent to the IAB ahead of the annual meeting. (See IAB - Section D under Criterion 2)
   7. Annual faculty retreat
      1. Comments from a broad range of discussion concerning Objectives, Learning Outcomes and Courses that drives the undergraduate curriculum. (See Faculty - Section D under Criterion 2)
2. Any revisions to Program Educational Objectives as proposed by the Assessment Committee are considered for adoption during the annual faculty retreat.
3. Program Educational Objectives are evaluated for importance by conducting expanded surveys from Alumni, Faculty, Employers and the IAB at least once every five years. Two specific requests are made of the constituencies:
   1. *To rate the importance of the Program Educational Objective*
   2. *To suggest possible additions, deletions or modifications for any of the Program Educational Objectives.*

The timeline for the review process of Program Educational Objectives is outlined in Table 2.1.

|  |  |
| --- | --- |
| **Date** | **Activities** |
| 2006-2012  (Winter/Spring Quarter) | Annual constituency (students, alumni, employers, IAB and faculty) surveys are conducted |
| 2006-2012  (Winter Quarter) | Annual faculty retreats |
| 2006-2012  (Spring Quarter) | Annual IAB meetings |
| Winter 2010 | Assessment Committee proposed revised Program Educational Objectives.  Revised Program Educational Objectives were adopted by the faculty. |
| Fall 2011 | Constituency (alumni, employers, IAB and faculty) surveys are conducted to validate the importance of Program Educational Objectives |

**Table 2.1: Program Educational Objectives review timeline**

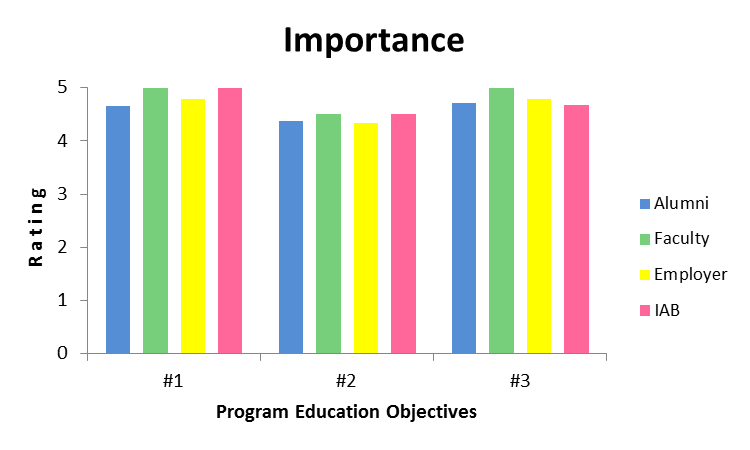
Prior to Winter 2010, our Program Educational Objectives were described as follows:

1. *Graduating students have the knowledge and skills to pursue a career in industry and/or continue their education in graduate programs.*
2. *Graduates have the knowledge and skills that enable them to participate in life-long learning and to adapt to an ever-changing technological environment.*

As described in Table 2.1, the Program Educational Objectives were revised in Winter 2010 (See Section B under Criterion 2 for a list of the revised Program Educational Objectives). The major revisions are as follows:

* The revised Program Educational Objectives sought to separate our graduates into two categories: (i) those who will enter the work force immediately after graduation and (ii) those who will pursue advanced degrees in graduate programs.
* The revised Program Educational Objectives sought to emphasize the notion of “problem solving”, “critical thinking” and “communication skills” as per the feedback from various constituency surveys.
* The revised Program Educational Objectives are closely tied to the revised Student Learning Outcomes.

Constituency Surveys were conducted in Fall 2011 to validate the importance of the revised Program Educational Objectives. This step completes the process of revising Program Educational Objectives and obtains final approval from all the constituencies.



**Figure 2.6: Importance of Program Educational Objectives**

As shown in figure 2.6, all three Program Educational Objectives are deemed important. Assuming all constituencies are weighted equally, all the Program Educational Objectives have received a ranking higher than 4.0/5.0.

* Program Educational Objective #1 received an average ranking of 4.86/5;
* Program Educational Objective #2 received an average ranking of: 4.43/5;
* Program Educational Objective #3 received an average ranking of: 4.79/5;

Thus, the Program Educational Objectives are acceptable to the program constituencies. The assessment process to ensure the achievement of the Program Educational Objectives will be described in Section A under Criterion 4.

In addition, comments from constituency members have been very positive. The qualitative data collected in the IAB meetings reinforces the above results and indicates that the Program Educational Objectives are aligned to the needs of the industry. A few pertinent comments from various constituency members are given in Section D-2 under Criterion 4.

# CRITERION 3. STUDENT OUTCOMES

**A.     Student Outcomes**

Student Learning Outcomes are specific skills that students will possess at the end of the degree program. Student Learning Outcomes provide curricular guidelines with respect to the program. The Student Learning Outcomes of the undergraduate program in Computer Science at California State University, Los Angeles are:

1. *Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*
2. *Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*
3. *Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*
4. *Students will have a fundamental understanding of computer systems.*
5. *Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*
6. *Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*
7. *Students will be able to communicate effectively orally and in writing.*
8. *Students will have the knowledge, skills, and attitudes for lifelong self-development.*
9. *Students will have the ability to analyze the local and global impact of computing on individuals and society.*
10. *Students will have a fundamental understanding of social, professional, ethical, legal, and security issues in computing.*

The Student Learning Outcomes are published at:

1. Department web site http://www.calstatela.edu/academic/ecst/cs/outcomes.php
2. Program assessment documentation site http://csns.calstatela.edu/wiki/content/assessment/undergrad/
3. Computer Science Department office and all faculty offices.

**B.     Relationship of Student Outcomes to Program Educational Objectives**

The Program Educational Objectives as outlined under Criterion 2 are brought to fruition through the Student Learning Outcomes as outlined under Criterion 3. i.e., the Student Learning Outcomes lay the foundation for our graduating student in achieving the professional goals set out in the Program Educational Objectives. Each Program Educational Objective is closely related to one or more Student Learning Outcome as shown in Figure 3.1 and also described below:

**Program Educational objective #1**: *Students who enter the workforce will have established themselves as effective professionals by having solved real problems through the use of their computer science knowledge and their communication, critical thinking, and problem solving skills.*

The following Student Learning Outcomes are largely concerned with the development of the theoretical understanding and skills that are necessary to succeed as software professional. Thus, students that are successful at attaining these outcomes should also find success in the workforce.

1. *Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*
2. *Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*
3. *Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*
4. *Students will have a fundamental understanding of computer systems.*
5. *Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*
6. *Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*
7. *Students will be able to communicate effectively orally and in writing.*
8. *Students will have the knowledge, skills, and attitudes for lifelong self-development.*

**Program Educational objective #2**: *Students who continue in academia will have been successful in pursuing advanced degrees and in demonstrating their ability to master advanced areas of computer science.*

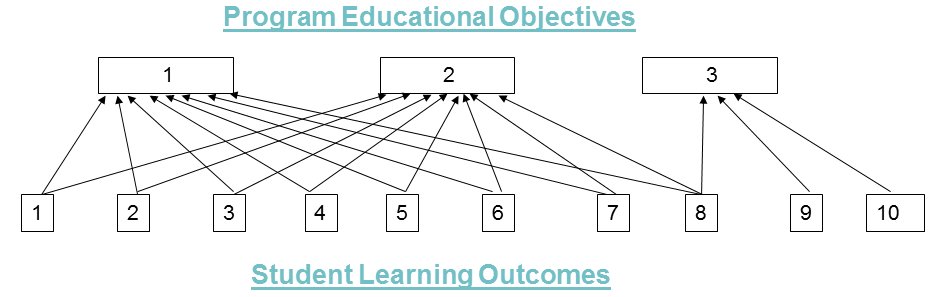
The following Student Learning Outcomes are largely concerned with the development of the theoretical understanding and skills that are necessary to succeed as a graduate student.

1. *Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*
2. *Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*
3. *Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*
4. *Students will have a fundamental understanding of computer systems.*
5. *Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*
6. *Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*
7. *Students will be able to communicate effectively orally and in writing.*
8. *Students will have the knowledge, skills, and attitudes for lifelong self-development.*

**Program Educational objective #3**: *Students will have demonstrated their ability to adapt to a rapidly changing environment by having learned and applied new skills and new technologies.*

The following Student Learning Outcomes are largely concerned with the abilities to be successful throughout their career. Attainment of these outcomes provides graduates with the necessary views of the world that will allow them to adapt to diverse environments.

1. *Students will have the knowledge, skills, and attitudes for lifelong self-development.*
2. *Students will have the ability to analyze the local and global impact of computing on individuals and society.*
3. *Students will have a fundamental understanding of social, professional, ethical, legal, and security issues in computing.*



**Figure 3.1: Program Educational Objectives-Student Learning Outcomes Map**

**C.    Process for the Establishment and Revision of the Student Outcomes**

Our Department established an assessment process for the undergraduate degree program which describes the process for establishment and revision of Student Learning Outcomes. Revision of Student Learning Outcomes follows the same process as the revision of Program Educational Objectives as described earlier in Section E under Criterion 2. The outer loop, Loop #2 in Figure 2.5, shows the periodic process involved in establishing, assessing and revising the Student Learning Outcomes. This process employs feedback from the various constituencies who have an interest in the outcome of the program. Loop #2 is relatively a slow loop where assessment occurs usually once every five years, or sooner based on assessment data.

Our Student Learning Outcomes were originally established in the 2000-2001 academic year. These were later revised in 2004-2005 academic year and were in effect in the last ABET CAC visit in Fall 2006. Our Student Learning Outcomes were revised again in Winter 2009.

The actual mechanism for revision of Student Learning Outcomes is as follows:

1. Our Assessment Committee (Raj Pamula, Chengyu Sun, Russ Abbott) develops revised Student Learning Outcomes by considering the following input:
   1. Existing Program Educational Objectives and Student Learning Outcomes
   2. ACM (Association of Computing Machinery) curricular guidelines
   3. ABET accreditation standards.
   4. Annual feedback from surveys
      1. Comments from the constituencies (Alumni, Employers, Faculty and IAB) from annual surveys (conducted during Loop #1 described in Figure 2.5) to an open-ended question: *“enter any suggestions that you think would improve the program”*
   5. Annual discussions from IAB meetings
      1. Comments from a broad range of discussion concerning Objectives, Learning Outcomes and Courses
      2. Responses to questions sent to the IAB ahead of the annual meeting. (See IAB - Section D under Criterion 2)
   6. Annual faculty retreat
      1. Comments from a broad range of discussion concerning Objectives, Learning Outcomes and Courses that drives the undergraduate curriculum. (See Faculty - Section D under Criterion 2)
2. Any revisions to Student Learning Outcomes as proposed by the Assessment Committee are considered for adoption during the annual faculty retreat
3. Student Learning Outcomes are evaluated for importance by conducting expanded surveys from Alumni, Faculty, Employers and the IAB at least once every five years. Two specific requests are made of the constituencies:
   1. *To rate the importance of the Student Learning Outcome*
   2. *To suggest possible additions, deletions or modifications of the Student Learning Outcomes.*

The timeline for review process of Student Learning Outcomes is outlined in Table 3.1.

|  |  |
| --- | --- |
| **Date** | **Activities** |
| 2006-2012  (Winter/Spring Quarter) | Annual constituency (students, alumni, employers, IAB and faculty) surveys are conducted. |
| 2006-2012  (Winter Quarter) | Annual faculty retreat |
| 2006-2012  (Spring Quarter) | Annual IAB meeting |
| Winter 2009 | Assessment Committee proposed revised Student Learning Outcomes.  Revised Student Learning Outcomes were adopted by the faculty. |
| Fall 2011 | Constituency (students, alumni, employers, IAB and faculty) surveys conducted to validate the importance of Student Learning Outcomes |

**Table 3.1: Student Learning Outcomes review timeline**

Prior to Winter 2009, our Student Learning Outcomes were as follows:

1. *Students will have a broad understanding of computing at all levels of abstraction.*
   1. *Graduating students will have a strong foundation in the design, analysis, and application of many types of algorithms.*
   2. *Graduating students will have a fundamental understanding of Computer Architecture and Operating Systems.*
   3. *Graduating students will have a fundamental understanding of Automata Theory.*
   4. *Graduating students will have a fundamental understanding of Programming language Paradigms.*
   5. *Graduating students will be able to demonstrate fluency in at least one programming language and acquaintance with several more.*
   6. *Graduating students will have fluency in at least one Operating System and acquaintance with several more.*
2. *Students have had the opportunity to focus in depth on selected areas of Computer Science.*
3. *Students will have the training to design and implement a large software system and will have the ability to work both individually and collaboratively.*
4. *Students will have sufficient oral and written communication skills*.
5. *Students will have the skills to pursue careers in industry and/or continue their education in graduate programs.*
6. *Students will have the skills to adapt to the evolving technologies in Computer Science.*

As described in Table 3.1, the Student Learning Outcomes were revised (See Section 3A for a list of the revised Student Learning Outcomes) in 2009. The revised Student Learning Outcomes are ensured of having appropriate assessment measures tied to the underlying curriculum.

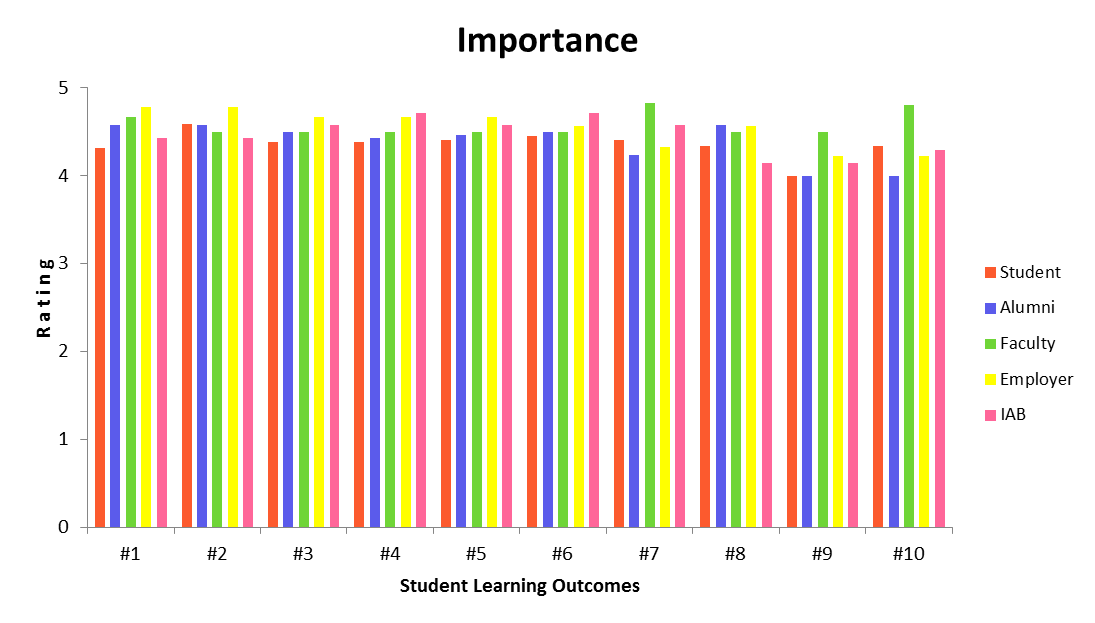
The major reasons are described below:

* At the time when the Student Learning Outcomes were revised in 2006, the ABET CAC criteria did not indicate a set of standard program outcomes similar to the Engineering Criteria. However, in 2007/2008, CAC adopted a set of standard “a..k” criteria. As ABET is an important program constituency, we decided to restructure the Student Learning Outcomes to meet all the ABET CAC “a..k” criteria.
* The revised Student Learning Outcomes spell out the skill areas of “computing” and list them as independent outcomes, which were earlier listed as #1 (“a..f”).
* The revised Student Learning Outcomes also avoid confusion with the revised “a..k” general criteria established by the Computing Accreditation Commission of ABET.
* The revised Student Learning Outcomes reflect feedback from all constituencies.

Constituency Surveys were conducted to validate the importance of Student Learning Outcomes. This step completes the process of revising Student Learning Outcomes and obtains final approval from all the constituencies.

As shown in figure 3.2, all ten Student Learning Outcomes are deemed important. Assuming all constituencies are weighted equally, all the Student Learning Outcomes have received a ranking higher than 4.0/5.0:

* Student Learning Outcomes #1 received a ranking of 4.55/5;
* Student Learning Outcomes #2 received a ranking of 4.57/5;
* Student Learning Outcomes #3 received a ranking of 4.52/5;
* Student Learning Outcomes #4 received a ranking of 4.54/5;
* Student Learning Outcomes #5 received a ranking of 4.52/5;
* Student Learning Outcomes #6 received a ranking of 4.54/5;
* Student Learning Outcomes #7 received a ranking of 4.47/5;
* Student Learning Outcomes #8 received a ranking of 4.42/5;
* Student Learning Outcomes #9 received a ranking of 4.17/5;
* Student Learning Outcomes #10 received a ranking of 4.33/5;



**Figure 3.2: Importance of Student Learning Outcomes**

Thus, the Student Learning Outcomes are acceptable to the program constituencies. The assessment process to ensure the achievement of the Student Learning Outcomes will be described in Section B under Criterion 4.

In addition, comments from constituency members have been very positive. The qualitative data collected in the IAB meetings reinforces the above results and indicates that the Student Learning Outcomes are aligned to the needs of the industry. A few pertinent comments from various constituency members are given in Section D-2 under Criterion 4.

# CRITERION 4. CONTINUOUS IMPROVEMENT

Continuous improvement is best accomplished when the assessment process is supported by a system that is closely tied to underlying course level instruction. This section describes CSNS, a system that was developed by a faculty member to meet this objective.

The Computer Science Network Services (CSNS) is a path-breaking web-based system that provides a number of services that facilitate teaching, learning, student administration, and program assessment. CSNS was developed by Dr. Chengyu Sun who is also the assessment coordinator for the Computer Science programs. CSNS has as a goal to combine learning management, student administration, and program assessment into one user friendly system.

CSNS is unique in its breadth of coverage.

* Some of the established Learning Management Systems, such as Blackboard <http://www.blackboard.com/>) or Moodle (<http://www.moodle.com/>), are very course-centric in their approach and focus primarily on course-level content creation and management. Blackboard does not provide any sort of assessment tools.
* Some of the established Assessment Management Systems, such as TaskStream (<https://www.taskstream.com>), provide electronic portfolio and assessment management tools. But TaskStream does not integrate or provide services for Learning Management.

Furthermore both Blackboard and TaskStream are subscription services while CSNS is a homegrown product. CSNS offers a set of general-purpose yet tightly integrated tools that provides more services than the combined capabilities of Blackboard and TaskStream.

Besides basic course management, CSNS services include forums, surveys, mailing lists, a wiki, news, an online file manager, stored queries, and a charting capability. These components are well integrated with the CSNS core functionality. For instance, each course has a forum associated with it. Since the instructor and the students of the course are automatically subscribed to the forum they will be notified whenever a new entry is posted. On the other hand, each of these components can also be used as a standalone service that is not directly related to teaching and learning. For example, users may use the wiki to create their own home pages. The flexibility of these components, especially when they are used together, creates many valuable usage scenarios. For example, the assessment process itself is documented on the wiki (<http://csns.calstatela.edu/wiki/content/assessment/>).

CSNS went online in Spring 2006, and by Spring 2011 it has served over 1500 students and faculty in more than 300 classes.

CSNS also performs tasks needed to implement the assessment process. CSNS provides two types of support for program assessment. At the course level, a number of assessment artifacts such as a course journal, skill evaluations, and key assignments can be collected whenever a course is taught. By building assessment elements into everyday teaching and learning, we ensure that program assessment is a rigorous, systematic, and continuous process. In addition to course-level assessment artifacts, CSNS also supports assessment instruments such as opinion surveys, web portfolios, and the Major Field Test—an external test administrated by the Educational Testing Service (ETS, <http://www.ets.org/>).

The CSNS-based assessment process has been demonstrated during both (i) our University-level program review (held once every five years) and (ii) university reaccreditation by WASC ([Western Association of Schools and Colleges](http://www.wascsenior.org/)). In both evaluations CSNS received enthusiastic commendations.

The methodologies used to evaluate, analyze, document, and maintain both the Program Educational Objectives and Student Learning Outcomes using CSNS are described in the sections below.

1. **Program Educational Objectives**

Assessment and evaluation of the extent to which the program educational objectives are met is held once every five years (or sooner) based on trends observed during the annual assessment. As described under Criterion 2 (Section E), the annual assessment process consists of constituency surveys, IAB meetings and annual faculty retreats.

Since the Program Educational Objectives are strongly related to the Student Learning Outcomes (see Figure 4-58), an indirect measure of whether the Program Educational Objectives are satisfied is the extent to which the underlying Student Learning Outcomes are satisfied. As described in Section B-3, all ten Student Learning Outcomes have been achieved.

The following annual constituency surveys are used to evaluate the extent to which the Program Educational objectives have been achieved.

* Alumni Survey
* Employers' Survey
* Faculty Survey
* IAB Survey

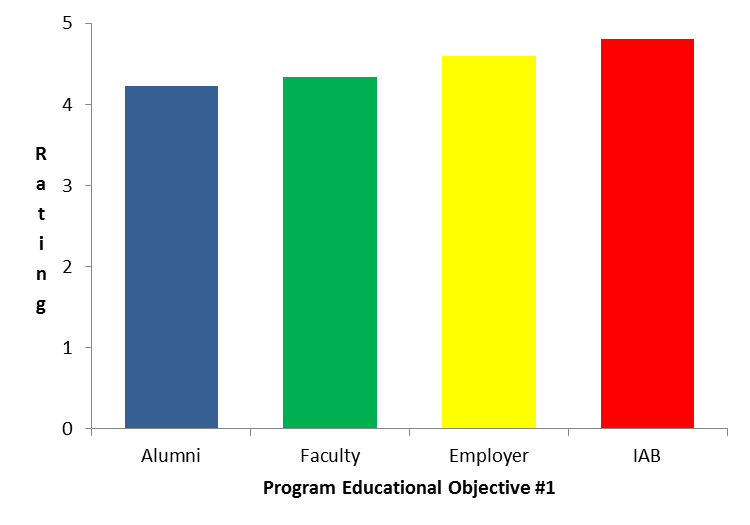
CSNS allows the assessment coordinator to administer the surveys to the appropriate alumni, faculty and IAB groups.

Copies of the surveys conducted during the 2011-2012 cycle are described in the 2012 Assessment Report (see Section D-1 below).

Following are analyses of the Program Educational Objectives.

**Program Educational Objective #1**: *Students who enter the workforce will have established themselves as effective professionals by having solved real problems through the use of their computer science knowledge and their communication, critical thinking, and problem solving skills.*

As Figure 4.1 illustrates alumni, employer, and faculty have all been satisfied by the achievement of this objective. Average survey ranking shows that alumni, faculty and alumni strongly agreed that our program is achieving this objective very well.

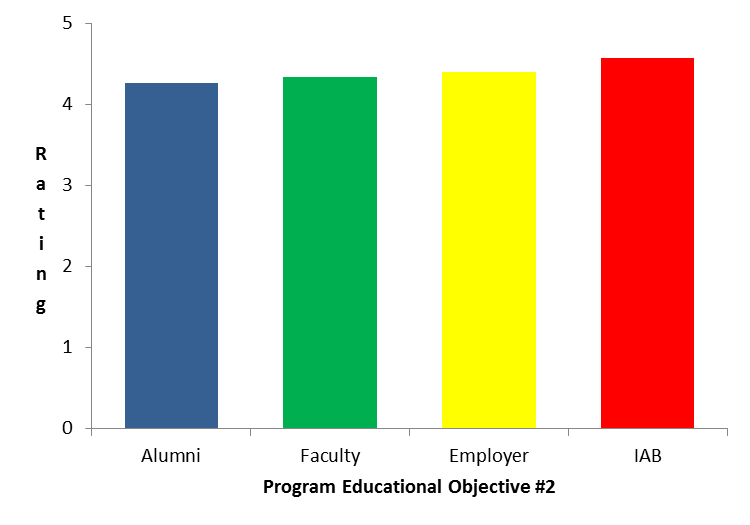


**Figure 4.1: Satisfaction of Program Educational objective #1**

Alumni responses on the annual surveys indicated that they are all currently employed in Computer Science-related industries such as Defense, IT, Consulting, Healthcare, Government, etc..

**Program Educational objective #2**: *Students who continue in academia will have been successful in pursuing advanced degrees and in demonstrating their ability to master advanced areas of computer science.*

As Figure 4.2 illustrates alumni, IAB, and faculty have all been satisfied by the achievement of this objective.



**Figure 4.2: Satisfaction of Program Educational Objective #2**

In addition, alumni were asked the following question on the annual survey:

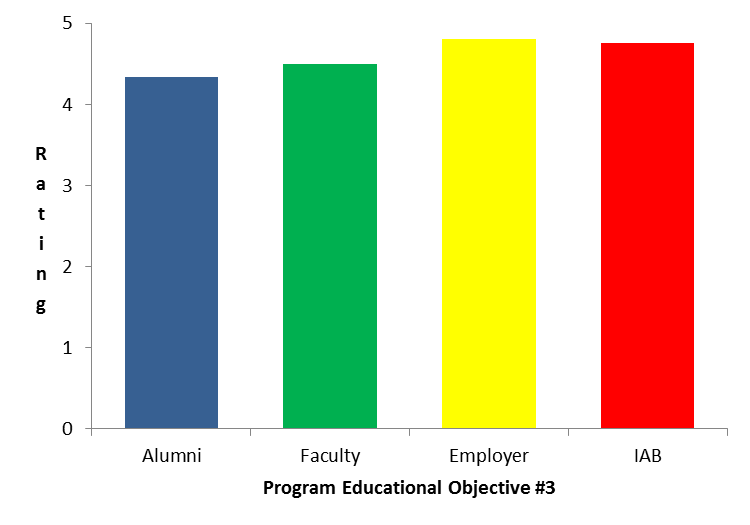
*Have you received/are you pursuing any additional professional development and/or advanced degrees after obtaining your B.S. degree?*

The responses vary from year to year. The summary analysis below is based on the self-reported data by the alumni over the past five years. Faculty experience indicates that the analysis below is fairly accurate.

* Three students (Myle Ott, Rone Lim, Daniel Firpo) are expected to complete their PhD in Computer Science in the next one year while another student (Sanmit Narveekar) is set to start his PhD in 2012.
* Approximately 20% of the respondents indicated that they have received a Master of Science (MS) degree in Computer Science.
* All “Outstanding Graduating Seniors” since 1990 (whose names are displayed on a plaque in the Department of Computer Science) have either completed or are currently enrolled in an M.S or PhD program.
* Approximately 2% of the respondents indicated that they have received a Master’s degree other than in Computer Science, such as an MBA, JD, or in another discipline.
* Approximately 10% of the respondents indicated that they have received Professional Certificates such as from CISCO, Microsoft, IBM, Oracle, SUN etc.,

**Program Educational objective #3**: *Students will have demonstrated their ability to adapt to a rapidly changing environment by having learned and applied new skills and new technologies.*

As Figure 4.3 illustrates alumni, IAB, and faculty have all been satisfied by the achievement of this objective.



**Figure 4.3: Satisfaction of Program Educational Objective #3**

The results described above confirm that our program is attaining all three Program Educational Objectives. In addition, comments from constituency members have been very positive over the last five years. The qualitative data collected in the IAB meetings reinforces the findings from constituency surveys. In all meetings since the last ABET visit, our industrial advisors agree that our program is meeting the educational objectives satisfactorily.

A few pertinent comments from alumni and IAB are included in Section D-2 under Criterion 4. However, the number of responses from the employers has been low when compared to the number of responses from the other constituencies.

1. **Student Outcomes**

As described in Figure 2.5, assessment for the undergraduate degree program is a two-loop process. The inner loop, loop #1, describes how the Student Learning Outcomes are evaluated by collecting data from various measures. Loop#1 is a fast loop where Student Learning Outcomes are assessed annually and ensure continuous improvement both at the course level and at the program level. Results of each year’s assessment measures are used to provide information to help guide refinements to the program. This process is detailed in the following sub-sections.

B-1: Assessment measures on CSNS

B-2: Assessment data collection on CSNS

B-3: Analysis of Student Learning Outcomes

B-4: ABET criteria – Student Learning Outcomes mapping

**B-1 Assessment measures on CSNS**

Assessment measures (Direct or Indirect) are detailed metrics that assist in assessing whether a specific Learning Outcome has been achieved. Direct measures provide for the direct examination or skills against measureable performance indicators; indirect measures are based on opinions expressed in surveys, during IAM meetings, and through other channels. Thus, direct measures of a learning outcome reveal what students know and can do while indirect measures suggest why performance was above or below expectations and what might be done for improvement. Both direct and indirect measures are employed for assessment purposes with all direct measures embedded in relevant courses. The four types of measures employed are Skill evaluations, Course assignments, Major Field Test, and Surveys.

* + Skill Evaluations

Skill Evaluation is a direct measure that allows faculty to directly observe a student's demonstration of a particular skill using certain performance indicators.

Prior to 2011, we employed a summative assessment of skills that were determined by the faculty. This practice was modified after the faculty in the Assessment Committee attended an ABET sponsored *Faculty Workshop on Sustainable Assessment Processes*.

The new process defines the performance indicators by well-defined rubrics that generally contain three components:

1. Dimensions: the specific Performance Indicator being evaluated
2. Scale: the possible levels of performance usually on a 5 point scale
3. Description: a specification of the required student performance for each level

Rubrics provide clear criteria for each level of performance so those who conduct the assessment can make objective decisions, and those who are being evaluated know exactly what they need to accomplish. The rubrics were developed in Spring 2011 and are being employed in the 2011-2012 assessment cycle. Skills are evaluated in certain courses as described in Table 4.1. The rubrics to evaluate the skills are described in Tables 4.2 - 4.7. CSNS provides an easy-to-use mechanism to collect the data from skill evaluations every time the course is taught, usually once or twice a year.

* + Course Assignments

Since courses contribute to the achievement of Student Learning Outcomes, data can be compiled from courses to evaluate those outcomes. These direct measures are in the form of Course Assignments such as projects, papers, exams, presentations, and portfolios. Note that we do not assess each Student Learning Outcome in each course that contributes to that outcome. Instead we choose certain courses that allow us to measure how well each Student Learning Outcome is being met.

These assignments are carefully determined by the assessment committee, and in most cases, the artifacts collected from these assignments are a part of a student’s assessment portfolio. A summary of these course assignments are described in Table 4.8.

The course management module in CSNS allows faculty to easily create assignments. Assignments require either that students upload their solutions as files or that they complete the assignment online using a web browser. Key assignments that contribute assessment data to certain Student Learning Outcomes are chosen for collection. The data is collected every time the course is taught, usually once or twice a year.

* + MFT

The Major Field Test (MFT) is designed by the Educational Testing Service (ETS, <http://www.ets.org/>) to measure the knowledge and understanding obtained by students. The MFT exam is currently utilized by over 230 institutions and more than 9,100 students. This direct measure provides comprehensive data (a standardized score and a national percentile for each student) enabling us to evaluate student performance and compare it to programs at similar institutions nationwide. The MFT also provides three indicators.

1. Assessment Indicator #1: Programming
2. Assessment Indicator #2: Discrete Structures and Algorithms
3. Assessment Indicator #3: Systems: Architecture/Operating Systems/Networking/Database

Each indicator provides the mean (average) percent correct of test questions answered in that particular subdomains/content areas for the class as a whole. These indicators are tied closely to the Student Learning Outcomes as described in Table 4.9.

The MFT is usually conducted once a year. Data reports produced by ETS are easily imported to CSNS.

* + Surveys

Surveys provide indirect measures which gather perceptions of learning, opinions about learning or reflections on learning. Surveys also provide a means to ask qualitative open ended questions. Surveys are collected from five of our constituencies - students, faculty, alumni, employers, and industry partners.

Employers’ survey is sought by contacting the immediate supervisors to our alumni.

Surveys are targeted towards determining the “importance” and “satisfaction” of Student Learning Outcomes.

Surveys on the “satisfaction” of the learning outcomes indicate how well we achieve each learning outcome. These surveys are conducted every year.

The annual surveys are expanded, usually once in five years, to determine the satisfaction and importance of both Student Learning Outcomes and Program Educational Objectives. In essence, the annual surveys provide assessment data required during Loop #1 (see Figure 2.5 described under Criterion 2) activities while the expanded surveys provide additional assessment data required during Loop #2 activities.

A list of all surveys (conducted in 2011-2012) from various constituencies is described in Assessment Report 2012 (see Section D-1 under Criterion 4)

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| **Skill** | **Performance**  **Indicators** | **Courses** |
| **Team Work** | [Rubric](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/Team_Work/)  (Table 4.2) | CS337, CS437, CS496A, CS496C |
| **Oral Communication** | [Rubric](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/Oral_Communication/)  (Table 4.3) | CS337, CS437, CS496A, CS496C |
| **Written Communication** | [Rubric](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/Written_Communication/)  (Table 4.4) | CS437, CS496C |
| **Software Engineering - Requirements** | [Rubric](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/Requirements/)  (Table 4.5) | CS337, CS496A |
| **Software Engineering – Design** | [Rubric](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/Design/)  (Table 4.6) | CS437, CS496B |
| **Software Engineering - Implementation** | [Rubric](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/Implementation/)  (Table 4.7) | CS437, CS496C |

**Table 4.1: Skill Evaluations**

# 

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Performance**  **Indicator** | **1**  **Poor** | **2**  **Insufficient** | **3**  **Satisfactory** | **4**  **Good** | **5**  **Excellent** |
| **Participation** | Does not provide any ideas when participating in the group and in classroom discussion. Refuses to participate. | Rarely provides useful ideas when participating in the group and in classroom discussion. May refuse to participate. | Sometimes provides useful ideas when participating in the group and in classroom discussion. A satisfactory group member who does what is required. | Usually provides useful ideas when participating in the group and in classroom discussion. | Routinely provides useful ideas when participating in the group and in classroom discussion. |
| **Problem-solving** | Pretends to solve problems; Causes disruption to others work. | Does not try to solve problems or help others solve problems. Lets others do the work. | Does not suggest or refine solutions, but is willing to try out solutions suggested by others. | Refines solutions suggested by others. | Actively looks for and suggests solutions to problems. |
| **Attitude** | Is always  publicly critical of the project or the work of other members of the group. Has a negative attitude towards every aspect. | Is often publicly critical of the project or the work of other members of the group. Is often negative about the task(s). | Is occasionally publicly critical of the project or the work of other members of the group. Usually has a positive attitude about the task(s). | Is rarely publicly critical of the project or the work of others. Often has a positive attitude about the task(s). | Is never publicly critical of the project or the work of others. Always has a positive attitude about the task(s). |
| **Contribution** | Does not complete any assigned tasks and uses others to complete his/her work.  A dis-interested team member who relies on others to complete the overall project. | Completed most of the individual tasks but did not assist other group members during the project.  A passive team member who does not care about the overall project. | Completed individual task and assisted other group members some times during the project.  A good team member but needs to try harder to complete the overall project. | Completed most of the assigned tasks. Volunteered to assist group members in finishing the  tasks.  A strong group member who tries hard to complete the project. | Completed all assigned tasks. Always assisted other group members in finishing off the tasks.  A group leader who works hard to complete the overall project. |
| **Interaction** | Does not listen to other team members. | Rarely listens to, shares with, and supports the efforts of others. | Often listens to, shares with, and supports the efforts of others | Usually listens to, shares, with, and supports the efforts of others. | Almost always listens to, shares with, and supports the efforts of others. |

**Table 4.2: Team Work Rubric**

| **Performance**  **Indicator** | **1**  **Poor** | **2**  **Insufficient** | **3**  **Satisfactory** | **4**  **Good** | **5**  **Excellent** |
| --- | --- | --- | --- | --- | --- |
| **Logical organization** | Poor organization. Introduction and main points are undeveloped. | Not well organized and audience has difficulty following the presentation; main points are unclear. | Satisfactory organization; good introduction and conclusion; transitions are somewhat sudden; points are often not made clearly. | Good organization; clear introduction and conclusion; main points well stated; few lapses in transitions; most points made clearly. | Superb organization in logical sequence that is easily understood by the audience. Clear introduction; main points well stated with good transitions; clear summary and conclusion. |
| **English**  **Language** | Audience unable to follow most of the presentation because of language difficulties. | Many grammatical errors; speaks in incomplete sentences; accent difficult to understand. | Few grammatical errors; but sentences are either incomplete or run on. Accent requires significant effort to understand. Uses few English colloquial expressions. | Few grammatical errors; some sentences are either incomplete or run on; minimal accent; speaks what would normally be considered standard English. | No grammatical errors with exceptional sentence structures; fluent and elegant English. |
| **Technical vocabulary** | Seems unsure of the technical vocabulary. | Limited vocabulary with many errors; terms often used incorrectly. | Limited vocabulary; makes errors on a few terms; but overall does not embarrass him/herself technically. | Good use of technical terms but is slightly unsure of certain terms. | Exceptional use of technical terms; explains them well when necessary; uses language that is appropriate to the audience level. |
| **Presentation**  **Aids** | Slides seem to have been cut-and pasted together; No connection between slides. | Boring slides; Speaker seems unsure of what is coming next; | Slides seem to contain the right information but no apparent effort made to create truly effective and engaging slides. | Generally good set of slides; Conveys the main points reasonably well in a traditional way. | Well rehearsed, informative, creative, and engaging slide presentation. |
| **Audience interaction** | Makes no contact with audience and seems unaware of audience reactions. | Makes occasional eye contact with audience but seems to read from prepared notes. | Makes eye contact with at least a portion of audience; style does not invite audience participation; responds only briefly to audience questions. | Maintains good eye contact with audience but some responses to audience questions are incomplete. | Makes exceptional rapport with audience and responds to questions openly, honestly, and completely. |

**Table 4.3: Oral Communication scoring rubric**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Performance  Indicator | 1  Poor | 2  Insufficient | 3  Satisfactory | 4  Good | 5  Excellent |
| Document organization | No organization | Little evidence of organization with poor transitions | Logical organization with few lapses and acceptable transitions | Logical organization that displays completeness with few lapses in transitions | Exceptional organization and provides effective transitions. |
| Section  content | Demonstrates no focus on the topic. | Demonstrates insufficient focus on the topic and provides few details. | Maintains focus on the topic and provides adequate if minimal details | Maintains good focus on the topic and provides sufficient details | Maintains exceptional focus on the topic and provides ample supporting details |
| Sentence  structure | Does not follow the rules of English grammar. | Many grammatical errors in each paragraph. | Few grammatical errors but displays limited variety in sentence length and structure. | Few grammatical errors and sentences are appropriately varied in length and structure. | No grammatical errors with exceptional, varied, and appropriate sentence structure. |
| Technical  vocabulary | Virtually no command of technical vocabulary. | Limited use of technical vocabulary and makes many errors. | Limited use of technical vocabulary but makes a few errors. | Appropriate use of technical vocabulary; makes few errors; | Exceptional use of technical vocabulary. |
| Graphical  depictions | No graphical depictions | Very few graphical depictions | Uses sufficient graphical depictions but not tied to the text and with poor aesthetics | Appropriate graphical depictions that is tied to the text but with poor aesthetics | Exceptional depiction of graphics that is also aesthetically pleasing. |

**Table 4.4: Written Communication Scoring Rubric**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | PerformanceIndicator | 1Poor | 2Insufficient | 3Satisfactory | 4Good | 5Excellent | | **Development Process** | Does not understand the waterfall development process. Not familiar with the four development phases of Analysis, Design, Implementation, and Test. | Requirement reviews and design reviews are carried out, but the relationship between the reviews and implementation is vague. | Requirement reviews and design reviews are conducted and the relationship between the reviews and implementation is established. | Requirements analysis and design, implementation, and testing are planned using available tools such as MS Project. | Requirement analysis and design, implementation, and testing are clearly planned and reasonable. All requirements defined in the analysis phase can be traceable to the design and the eventual implementation. Produces a rigorous development plan and schedule. | | **Requirements**  **Accuracy** | Most requirements are wrong, invalid or not needed. | Many requirements are either not valid or not needed. | Many requirements are valid while some requirements are not fully understood. | Most requirements are a valid need in the software. | Each and every requirement is a valid need in the software. | | **Requirements**  **Documentation** | SRS document does not address any of the requirements clearly. | SRS document is sketchy and unclear with regards to many requirements. | SRS document is somewhat clear and addresses many of the requirements in sufficient detail. | SRS document is clear and addresses most of the requirements in sufficient detail. | SRS document is clear, understandable and addresses all the requirements in sufficient detail. | |

**Table 4.5: Software Engineering - Requirements**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | PerformanceIndicator | 1Poor | 2Insufficient | 3Satisfactory | 4Good | 5Excellent | | **Program**  **Design** | Shows virtually no understanding of the use of abstraction mechanisms. | Shows some understanding of the use of abstraction mechanisms. | Understands the process of object-oriented and functional design. | Understands how to write functions that abstract out the essential elements of a function and hide representation and other lower-level issues. | Demonstrates the ability to factor out appropriate abstractions in virtually all situations. | | **Libraries and frameworks** | Does not understand the value of libraries and frameworks. | Understands the value of libraries and frameworks but rarely uses them. | Understands the value of libraries and frameworks and uses them on occasion. | Understands the value of libraries and functions and uses them most of the time. | Understands the value of libraries and functions and uses them whenever possible. | | **Design**  **Patterns** | Design patterns are either unknown or used incorrectly. | Some knowledge of design patterns, but makes little use of them. | Analysis and design contains the correct use of design patterns, but only a few patterns are known well enough to be employed. | A large number of design patterns are known and their use is understood to a large extent. | A wide variety of design patterns are correctly used to speed up the design process while creating more reliable and reusable programs. | |

**Table 4.6: Software Engineering – Design**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | PerformanceIndicator | 1Poor | 2Insufficient | 3Satisfactory | 4Good | 5Excellent | | **Programming paradigms** | Does not understand concepts from programming paradigms (object oriented and functional). | Does not quite understand concepts from programming paradigms (object oriented and functional). | Understands and uses concepts from programming paradigms (object oriented and functional) some of the time. | Understands and uses concepts from programming paradigms (object oriented and functional) most of the time. | Understands and uses concepts from programming paradigms (object oriented and functional) when appropriate. | | **Functions and methods** | Functions/methods are usually longer than 2 dozen lines of code. | Functions/methods are often longer than 2 dozen lines of code. | Functions/methods are sometimes longer than 2 dozen lines of code. | Functions/methods are generally shorter than 2 dozen lines of code. | Functions/methods are almost never more than 2 dozen lines of code. | | **Testing** | Has no concept of testing. Stubs and drivers are not considered in the development stage. Does not use a testing framework. | Understands the concept of testing. Drivers and stubs are used but not well defined. May use a testing framework but only minimally. | Aware of the importance of testing. The use of stubs and drivers is considered before implementation. Uses a testing framework consistently. | Makes testing plans, and testing is integrated with development. Stubs and drivers are written before further implementation. Writes test cases for a testing framework before writing code. | Makes testing plans, and testing is integrated with development. Stubs and drivers are written before further implementation. Writes test cases for a testing framework before writing code. Produces rigorous test reports. | |

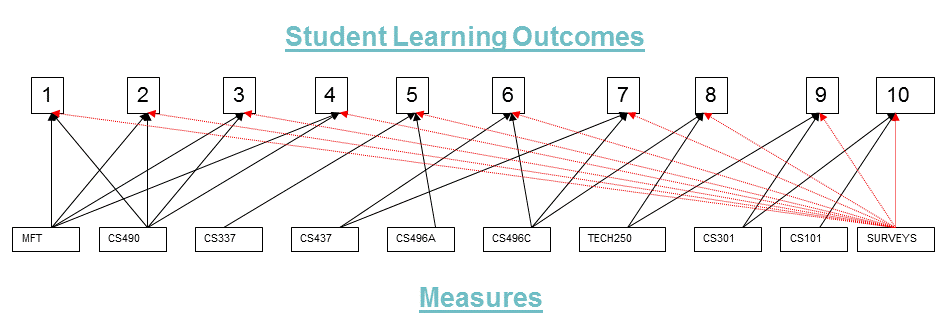
**Table4.7: Software Engineering - Implementation**

| Course | Assignments | Comments |
| --- | --- | --- |
| CS101 | Ethical Issues | Each student uploads project presentation slides for this assignment.  Measurable: Demonstrate the level of competence or knowledge. |
| CS301 | Societal Impact of Computing  Ethical Issues | Each student takes exam(s) for the two topics.  Measurable: Achievement of skills in terms of absolute levels of mastery. |
| CS337 | Project Requirement Document | Each student (or a group of students) uploads a project Requirement Document and does a formal presentation.  Measurable: Achievement of skills as evaluated by the faculty. |
| CS437 | Project Design Document  Project Documentation | Each student (or a group of students) uploads a Project Design Document and Project Report and does a formal presentation.  Measurable: Achievement of skills as evaluated by the faculty. |
| CS490 | A1 – Theory  A2 - Programming  A3 - Algorithms  A4 - Systems  Major GPA | Each student takes the MFT Assessment Tests, which consists of four topics (A1, A2, A3, and A4) related to those Student Learning Outcomes. These are referred to as Assessment Indicator #1, Assessment Indicator #2, Assessment Indicator #3 and Assessment Indicator #4.  Measurable: Achievement of skills in terms of absolute levels of mastery.  The student's Computer Science GPA.  Measurable: MFT-GPA correlation |
| CS491A/CS496A | Project Requirement Document | Each student (or a group of students) submits a Project Requirement Document at the end of this class and does a formal presentation.  Measurable: Achievement of skills as evaluated by the faculty. |
| CS496B | Project Design Document | Each student (or a group of students) submits a Project Requirement Design Document at the end of this class and does a formal presentation.  Measurable: Achievement of skills as evaluated by the faculty. |
| CS491B/CS496C | Project  Project Documentation  Project Presentation  Project Poster | Each student (or a group of students) submits the following documents and does a formal presentation.  Project\*  Each student (or a group of students) uploads a project in a zip or gzip file, presumably including everything - source code, binaries, libraries needed, documentation, and so on, to this assignment.  Project Documentation  Each student (or a group of students) uploads project documentation in a zip or gzip file. The documentation may include a design document, a project report, a user manual, and other documents related to the project.  Project Presentation  Each student (or a group of students) uploads presentation slides.  Project Poster  Each student (or a group of students) uploads a poster in the specified layout. These posters are displayed at the Department and College Industry Advisory Board meetings.  Measurable: Achievement of skills as evaluated by the faculty and the Industry Advisory Board. (If the project is good, the instructor can put the word "Good" in the Notes field of the grade form so the project can get a gold star in the project listings on CSNS) |
| TECH250 | Societal Impact of Computing  Lifelong Learning | Each student takes exam(s) and submits assignments for the two topics.  Measurable: Achievement of skills in terms of absolute levels of mastery. |

**Table 4.8: Course Assignments**

**B-2 Assessment data collection on CSNS**

CSNS is the centerpiece in implementing the assessment data collection process from the measures described in Section B-1. Each Student Learning Outcome is assessed by multiple measures as defined by the mapping in Figure 4.4. For example, SLO #4 derives data from measures implemented by MFT, CS490 course, and Surveys.



**Figure 4.4: Student Learning Outcomes – Measures Map**

Data collected from various measures employed for each Student Learning Outcome is indicated in Table 4.9. A few characteristics indicated in Table 4.9 are as follows:

* Data: Refers to the actual data collected from the indicated direct or indirect measure.
* Type: Refers to the type of the measure as one of the following (as defined earlier in Section B-1)
  + Assignment
  + MFT Skill Evaluation
  + Survey
* Target: Refers to the satisfactory level for achievement by the measure. Note that the target value is normally indicated as a number on a 5 point scale, if not specified otherwise.
* Frequency: Refers to the number of times the measure is employed in a year.

| **SLOs** | **Measure**  **Data** | **Measure**  **Type** | **Measure**  **Target** | **Measure**  **Frequency** |
| --- | --- | --- | --- | --- |
|  | CS490 Assessment Indicator #1  MFT Assessment Indicator #2  [SLO-1 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo1) | Course Assignment  MFT  Survey | 3 or higher  50th percentile or higher (based on the mean score of our students and the distribution of the mean scores of all the institutions).  50% or higher (measured in the percentage of the questions answered correctly by the students in the class).  3 or higher | 1 or 2 times a year  1 or 2 times a year  1 time a year |
|  | [CS490 Assessment Indicator #2](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs490_ai2)  [MFT Assessment Indicator #1](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/mft_ai1)  [SLO-2 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo2) | Course Assignment  MFT  Survey | 3 or higher  50th percentile or higher (based on the mean score of our students and the distribution of the mean scores of all the institutions).  50% or higher (measured in the percentage of the questions answered correctly by the students in the class).  3 or higher | 1 or 2 times a year  1 or 2 times a year  1 time a year |
|  | [CS490 Assessment Indicator #3](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs490_ai3)  MFT Assessment Indicator #2  [SLO-3 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo3) | Course Assignment  MFT  Survey | 3 or higher  50th percentile or higher (based on the mean score of our students and the distribution of the mean scores of all the institutions).  50% or higher (measured in the percentage of the questions answered correctly by the students in the class).  3 or higher | 1 or 2 times a year  1 or 2 times a year  1 time a year |
|  | [CS490 Assessment Indicator #4](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs490_ai4)  [MFT Assessment Indicator #3](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/mft_ai3)  [SLO-4 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo4) | Course Assignment  MFT  Survey | 3 or higher  50th percentile or higher (based on the mean score of our students and the distribution of the mean scores of all the institutions).  50% or higher (measured in the percentage of the questions answered correctly by the students in the class).  3 or higher | 1 or 2 times a year  1 or 2 times a year  1 time a year |
|  | [CS337 Skill Evaluation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs337_pra)  [CS496A Skill Evaluation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs491a_pra)  [SLO-5 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo5) | Skill evaluation  Skill evaluation  Survey | 3 or higher  3 or higher  3 or higher | 1 or 2 times a year  1 or 2 times a year  1 time a year |
|  | [CS437 Skill Evaluation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs437_dev)  [CS496C Skill Evaluation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs491b_dev) [SLO-6 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo6) | Skill evaluation  Skill evaluation  Survey | 3 or higher  3 or higher  3 or higher | 1 or 2 times a year  1 or 2 times a year  1 time a year |
|  | [CS337/CS437 Skill Evaluation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs337_oral)  [CS496ABC Skill Evaluation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs491_oral)  [CS337/CS437 Skill Evaluation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs337_written)  [CS496ABC Skill Evaluation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs491_written)  [SLO-7 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo7) | Skill evaluation  Skill evaluation  Skill evaluation  Skill evaluation  Survey | 3 or higher  3 or higher  3 or higher  3 or higher  3 or higher | 1 or 2 times a year  1 or 2 times a year  1 or 2 times a year  1 or 2 times a year  1 time a year |
|  | [TECH250/GE Block E Assignment](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/tech250_life)  [SLO-8 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo8) | Course Assignment  Survey | 3 or higher  3 or higher | 1 or 2 times a year  1 or 2 times a year |
|  | [CS301 Assignment](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs301_impact)  [TECH250 Assignment](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/tech250_impact)  [SLO-9 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo9) | Course Assignment  Course Assignment  Survey | 3 or higher  3 or higher  3 or higher | 1 or 2 times a year  1 or 2 times a year  1 time a year |
|  | [CS101 Assignment](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs101_ethics)  [CS301 Assignment](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs301_ethics)  [SLO-10 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo10) | Course Assignment  Course Assignment  Survey | 3 or higher  3 or higher  3 or higher | 2 or 3 times a year  1 or 2 times a year  1 time a year |
| [**1,2,3,4**](http://csns.calstatela.edu/wiki/content/assessment/undergrad/SLO/)  **(Overall)** | [MFT Median Score Percentile](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/mft_median) CS490 Tests Average  [MFT Score-Major GPA Correlation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/mft_gpa_correlation) | MFT  Course Assignment  MFT | 50th Percentile or higher  12 or higher (on a 20 point scale)  75th Percentile or higher | 1 or 2 times a year  1 or 2 times a year  1 or 2 times a year |

**Table 4.9: Data Collection**

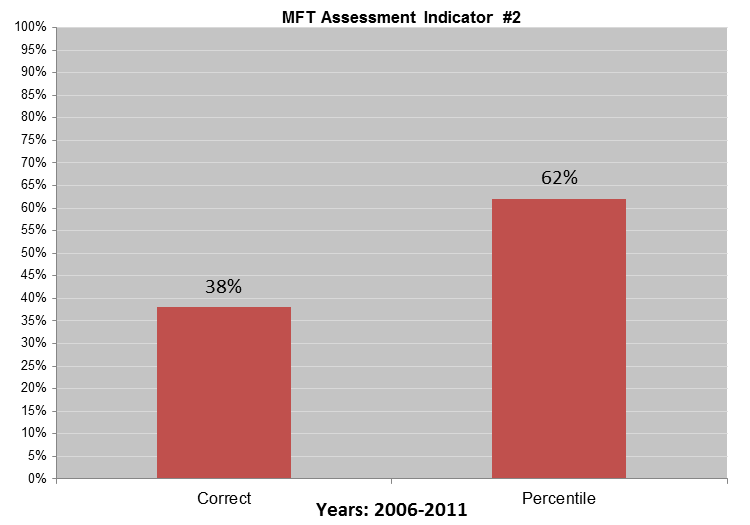
**B-3 Analysis of Student Learning Outcomes**

Data collected from various measures is analyzed to determine whether the achievement target for each Student Learning Outcome is met. Note the following:

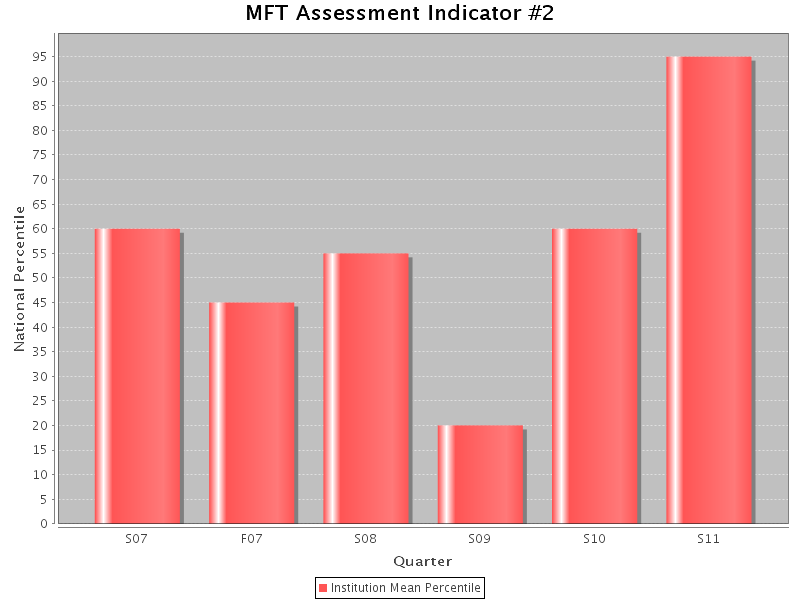
* CSNS has built in tools to query the assessment data and display the data graphically.
* Quarter labels on the X-axis refer to the individual quarter and year when the data was collected. For example, F09 refers to Fall 2009, S10 refers to Spring 2010 and W12 refers to Winter 2012.
* The number of survey responses from the employers is very low (especially in 2009 & 2010) when compared to the number of responses from the other constituencies.
* Additional analysis of data for the 2006-2008 periods is indicated in Assessment Report 2006-2008. During this period, not all assessment tools were available on CSNS.
* MFT data collected in Spring 2012 will be included in the Assessment Report 2010-2012.

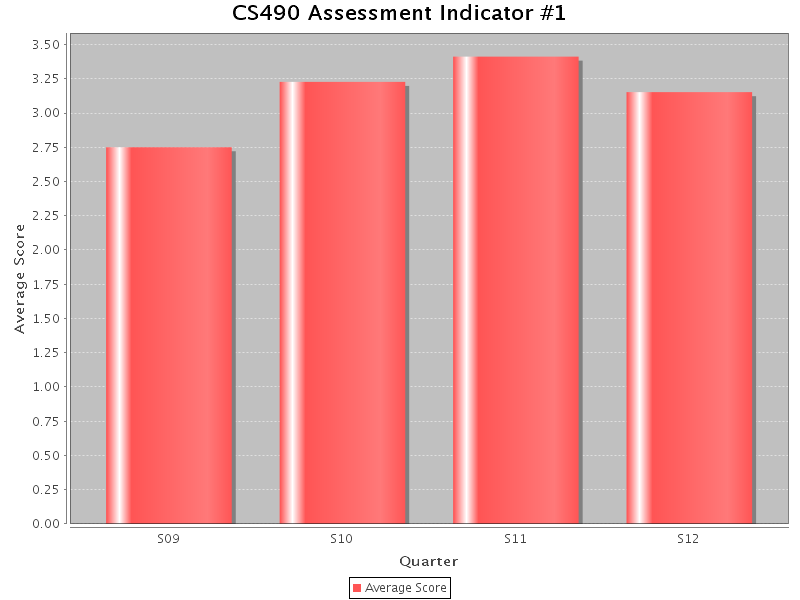
Each of the Student Learning Outcomes is analyzed below:

**Student Learning Outcome #1:** *Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*

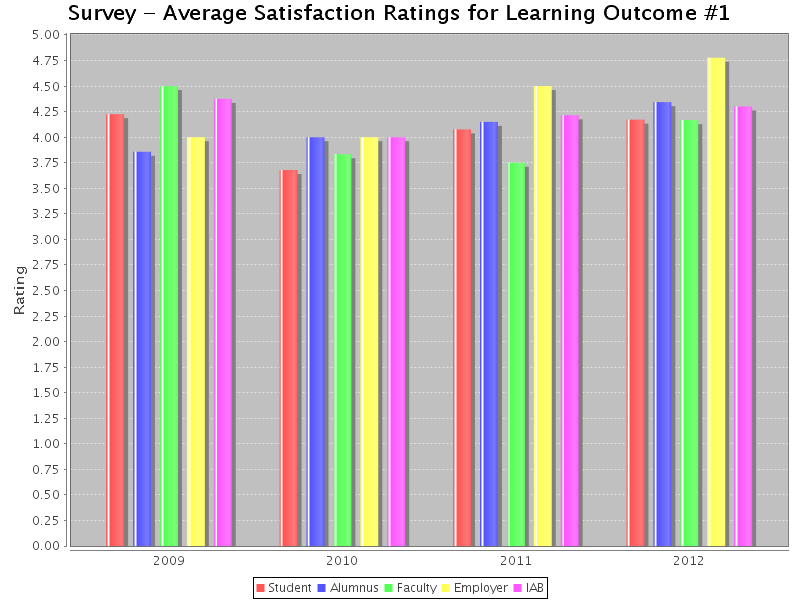
**

**Figure 4.10: MFT Assessment Indicator Results (2006-2011)**

**Figure 4.11: MFT Assessment Indicator Results (Quarterly)** 



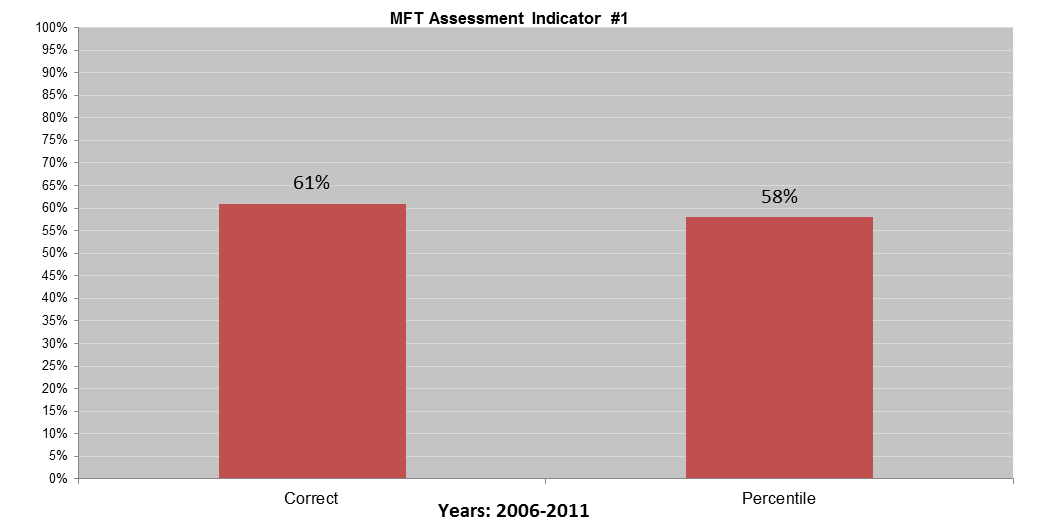
**Figure 4.12: CS490 Assessment Indicator Results (Quarterly)**

**Figure 4.13: Survey Results**

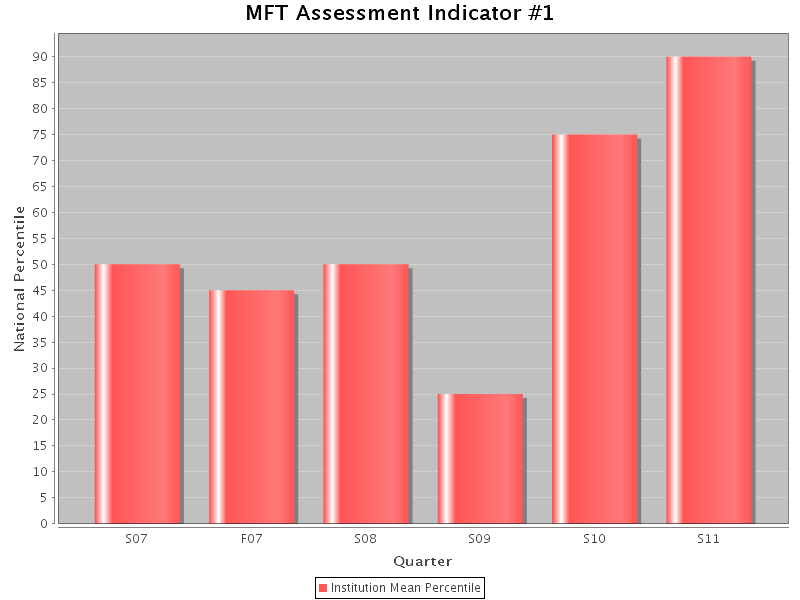
Analysis:

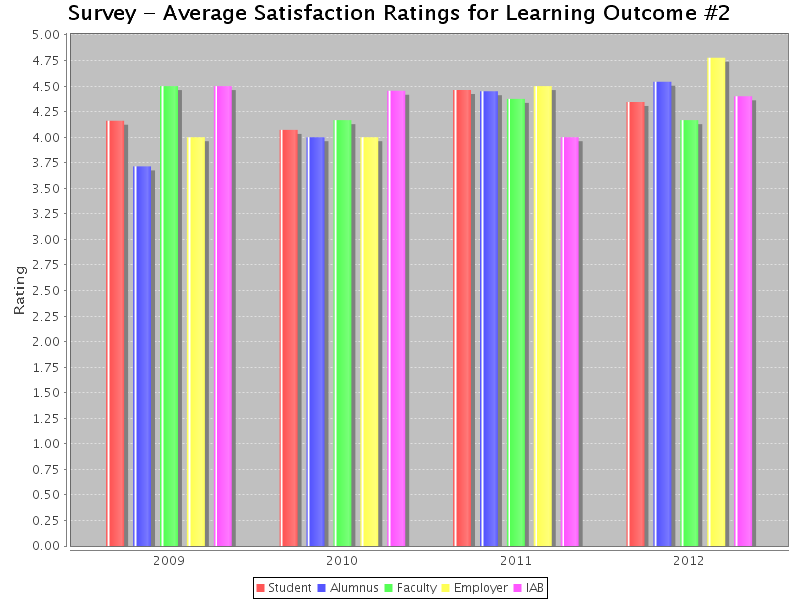
* From 2006-2011, our students placed in the 62nd percentile on MFT (based on the mean score of our students and the distribution of the mean scores of all the institutions. Our students’ mean score on the MFT is 38% (measured as the percentage of the questions that are answered correctly). (See Figure 4.10).
* It should be noted that since our class sizes are relatively small; there can large fluctuations from year to year, e.g., S09 vs. S11. (See Figure 4.11)
* The scores have been increasing over the last three years. This could also be attributed to increasing preparation in the CS490 course before taking the MFT. (See Figure 4.11)
* The results on the internal exam in CS490 indicate that the student average is around 3.25/5. (See Figure 4.12)
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory. (See Figure 4.13)
* All results exceed the target levels. Even though, the results are higher than the national averages, there is considerable room for improvement.

**Student Learning Outcome #2:** *Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*

****

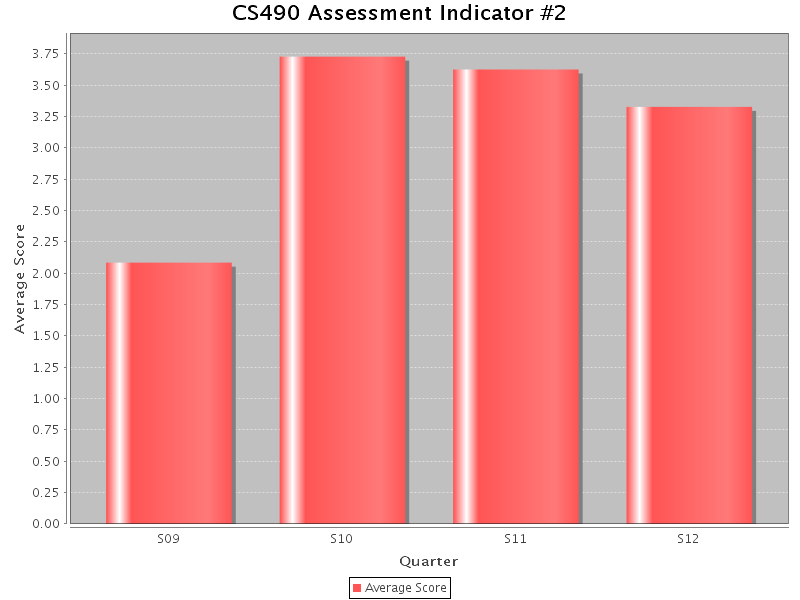
**Figure 4.14: MFT Assessment Indicator Results (2006-2011)**

**Figure 4.15: MFT Assessment Indicator Results (Quarterly)**



**Figure 4.16: CS490 Assessment Indicator Results (Quarterly)**

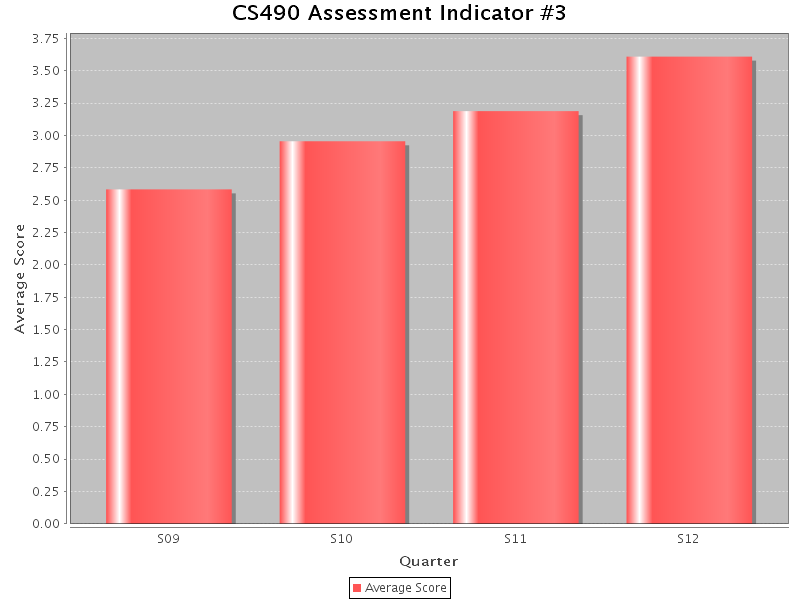
**Figure 4.17: Survey Results**



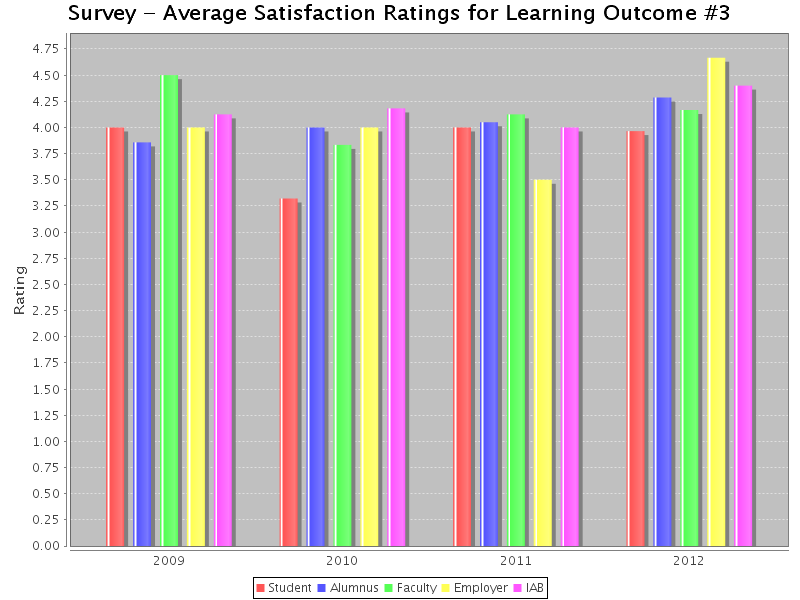
Analysis:

* From 2006-2011, our students placed in the 58th percentile on the MFT (based on the mean score of our students and the distribution of the mean scores of all the institutions). Our students’ mean score on the MFT is 61% (measured as the percentage of the questions that are answered correctly). (See Figure 4.14)
* It should be noted that since our class sizes are relatively small; there is a big fluctuation in the results as indicated in S09 and S11. (See Figure 4.15)
* The scores have been increasing over the last three years. This could also be attributed to increasing preparation in the CS490 course before taking the MFT. (See Figure 4.16)
* The results on the internal exam in CS490 indicate that the student average is around 3.25/5. (See Figure 4.16)
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory. (See Figure 4.17)
* All results exceed the target levels. There are no actions necessary to correct any deficiencies.

**Student Learning Outcome #3:** *Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*



**Figure 4.18:** **CS490 Assessment Indicator Results (Quarterly)**

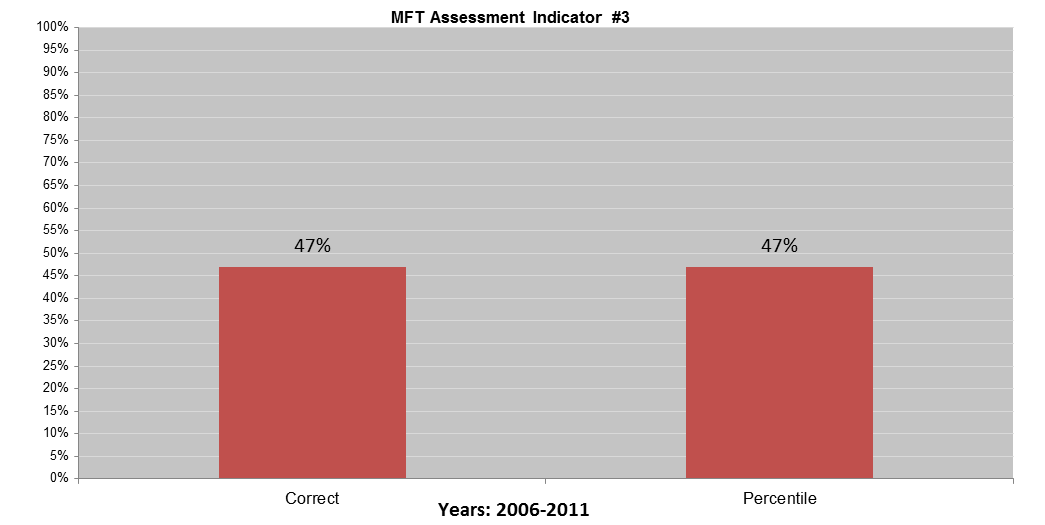


**Figure 4.19: Survey Results**

Analysis:

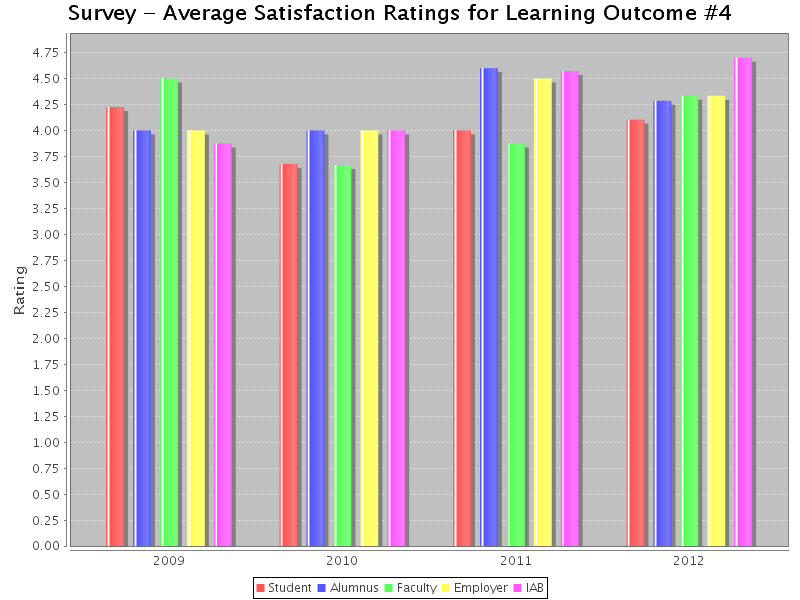
* From 2006-2011, our students placed in the 62nd percentile on MFT (based on the mean score of our students and the distribution of the mean scores of all the institutions). Our students’ mean score on MFT is at 38% (measured as the percentage of the questions that are answered correctly) (Note that MFT exam gives a combined result for SLO #1 And SLO#3. See Figure 4.10).
* The scores have been increasing over the last three years. This could also be attributed to increasing preparation in the CS490 course before taking the MFT (See Figure 4.11).
* The results on the internal exam in CS490 indicate that the student average is around 3.0/5 (See Figure 4.18).
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.19).
* All results exceed the target levels. Even though, the results are higher than the national averages and the results are better than last year, there is room for improvement.

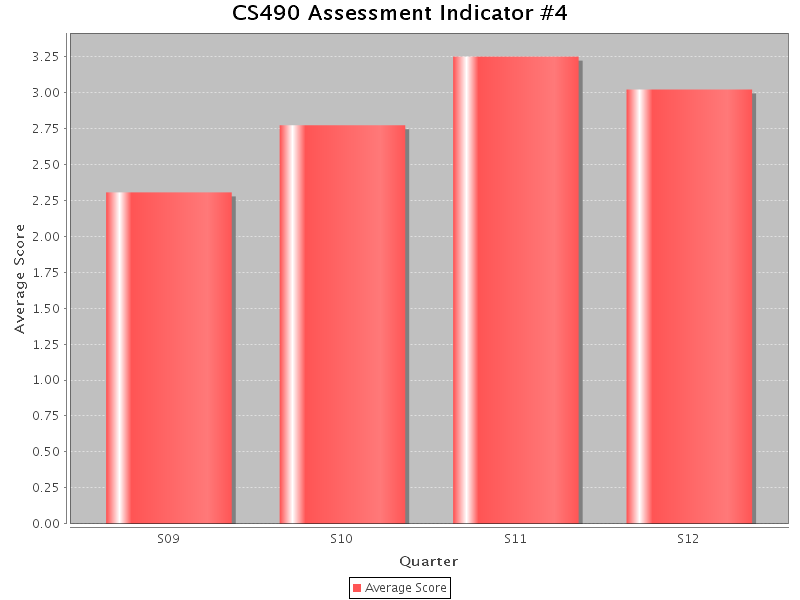
**Student Learning Outcome #4:** *Students will have a fundamental understanding of computer systems.*

****

**Figure 4.20: MFT Assessment Indicator Results (2006-2011)**

**Figure 4.21: MFT Assessment Indicator Results (Quarterly)**

**Figure 4.22: CS490 Assessment Indicator Results (Quarterly)**

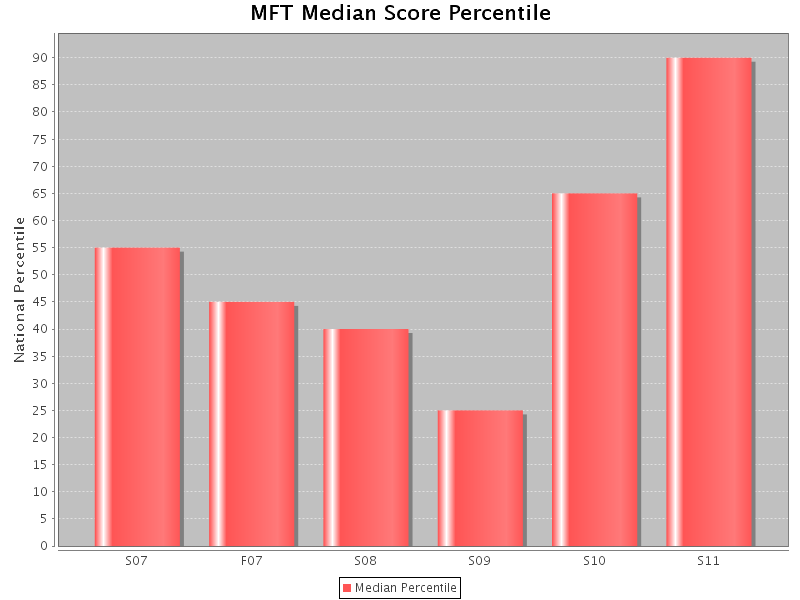


**Figure 4.23: Survey Results**

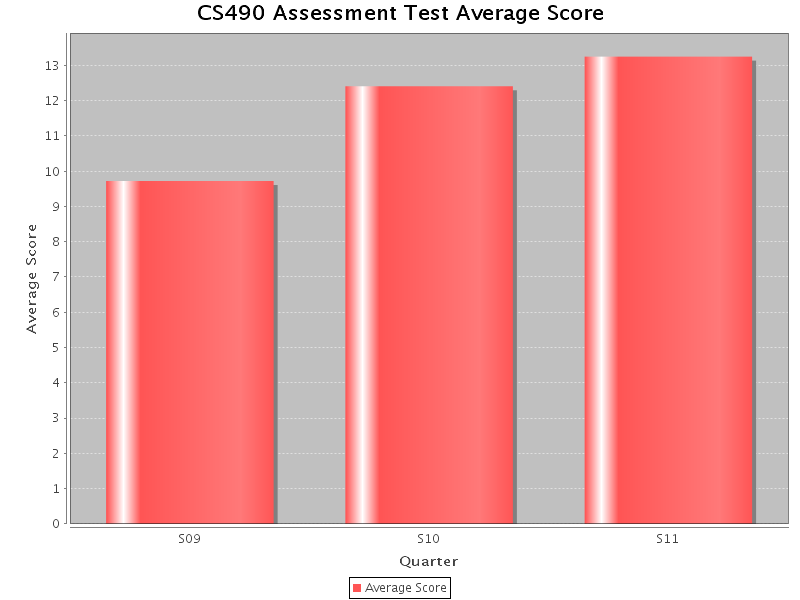
Analysis:

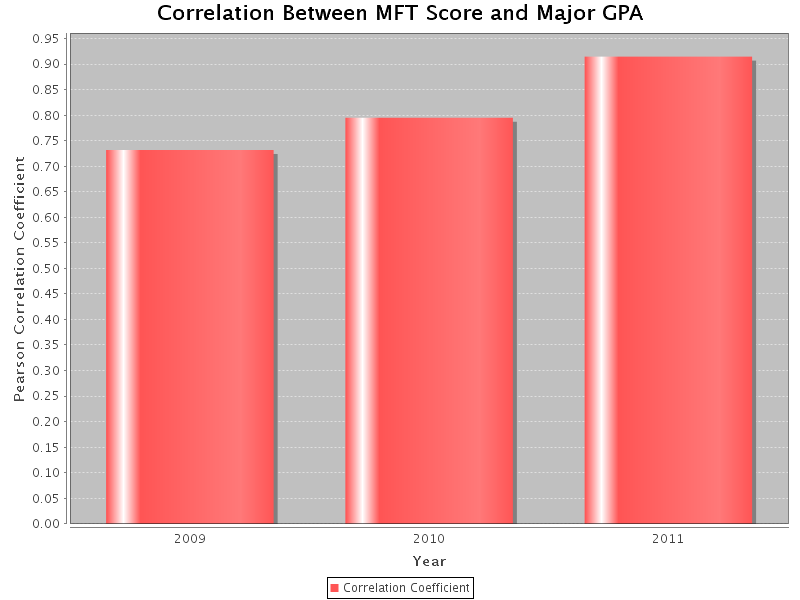
* From 2006-2011, our students placed in the 47th percentile on MFT (based on the mean score of our students and the distribution of the mean scores of all the institutions). Our students’ mean score on the MFT is at 47% (measured as the percentage of the questions answered correctly) (See Figure 4.20).
* It should be noted that since our class sizes are relatively small; there is a big fluctuation in the results as indicated in S09 and S11 (See Figure 4.21).
* The scores have been increasing over the last three years. This could also be attributed to increasing preparation in the CS490 course before taking the MFT (See Figure 4.21).
* The results on the internal exam in CS490 indicate that the student average is at 3.0/5 (See Figure 4.22).
* In 2006, ABET accreditation evaluators recommended that the coverage of topics pertaining to this SLO be introduced early in addition to an upper division required course. As a curricular action, CS245 was modified to expand the coverage of “Computer Organization” (2007) and CS490 had increased the coverage of these topics (2008). This contributed to the fact that the MFT results improved over the past few years.
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.23).
* All results exceed the target levels. Even though, the results are higher than the national averages and the results are better than last year, there is room for improvement.

**Student Learning Outcomes #1 to #4 (Summary)**

**Figure 4.24: MFT Median Results**

**Figure 4.25: CS490 Results**

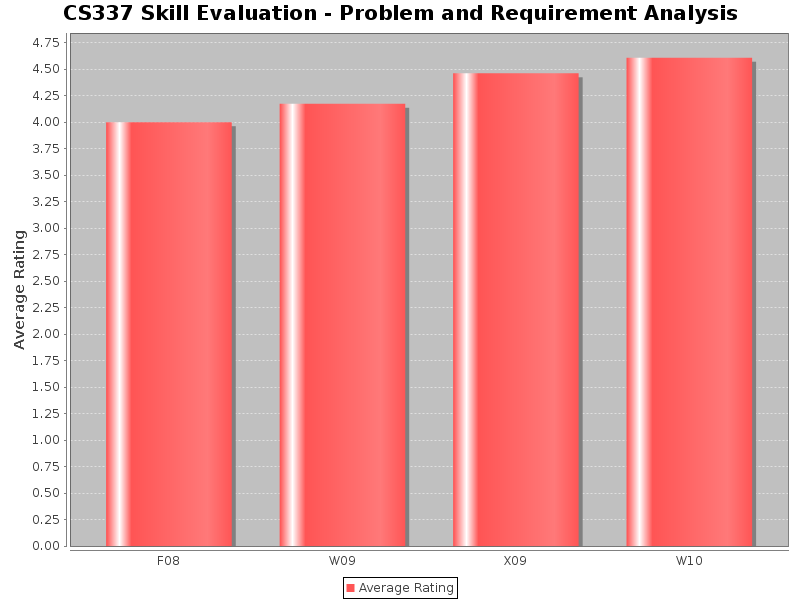
****

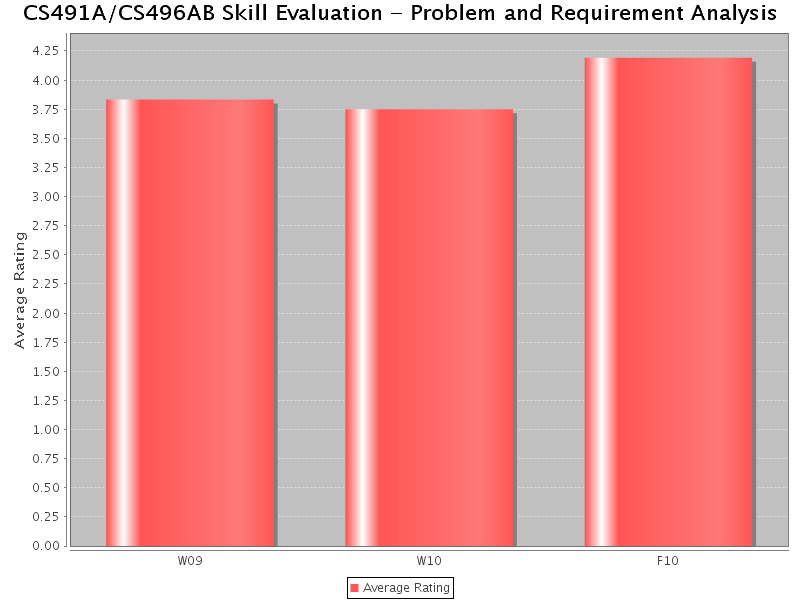
**Figure 4.26: MFT-GPA Correlation Results**

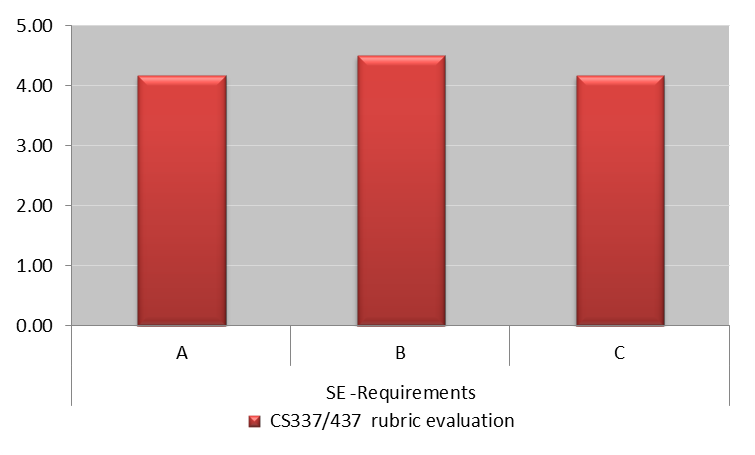
Analysis:

* Over the last five years, over 9,100 students from over 230 institutions have taken the MFT in Computer Science. During the same period, 130 graduating Computer Science seniors have taken the MFT at California State University, Los Angeles. The median student placed at the 57th percentile of the entire student population (See Figure 4.24).
* The results on the internal exam in CS490 indicate that the student average is 12.5/20 (See Figure 4.25).
* Correlation between MFT score and major GPA is around 0.8 (as measured by Pearson Correlation Coefficient) indicating a strong positive relationship between the two (See Figure 4.26).
* The overall results are satisfactory and above the target levels.

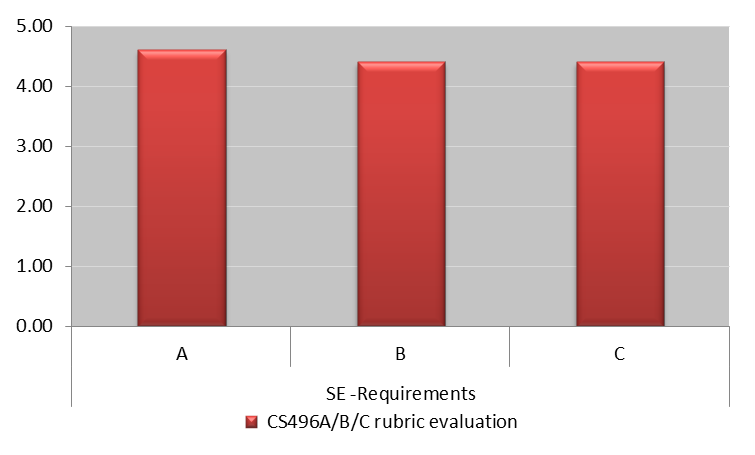
**Student Learning Outcome #5:** *Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*

**Figure 4****.27: CS337 Skill Evaluation**

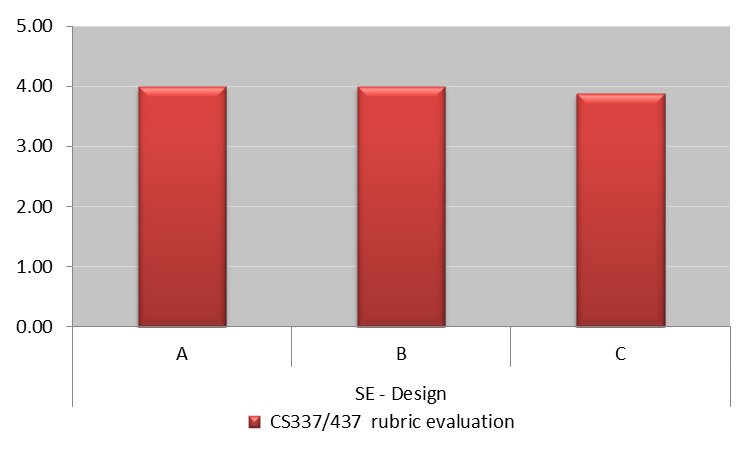
**Figure 4****.28: CS491/CS496 Skill Evaluation**

****

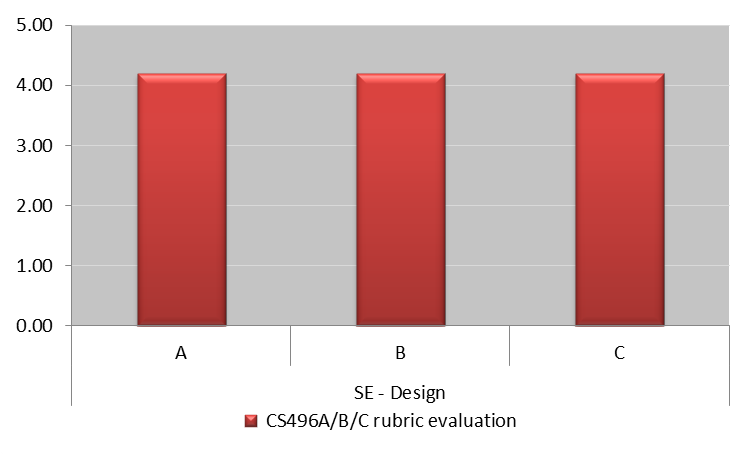
**Figure 4.29: CS337 Rubric Evaluation (SE-Requirements)**



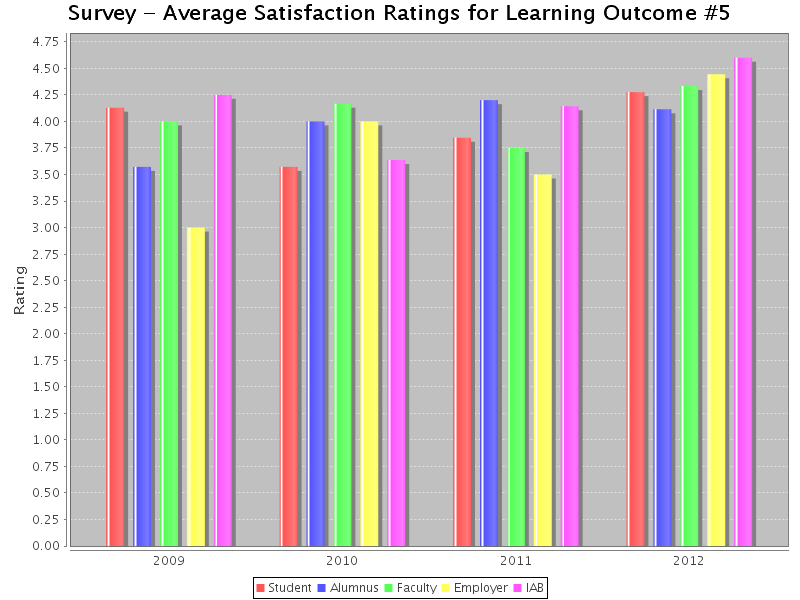
**Figure 4.30: CS496A Rubric Evaluation (SE-Requirements)**

****

**Figure 4.31: CS437 Rubric Evaluation (SE-Design)**



**Figure 4.32: CS496B Rubric Evaluation (SE-Design)**

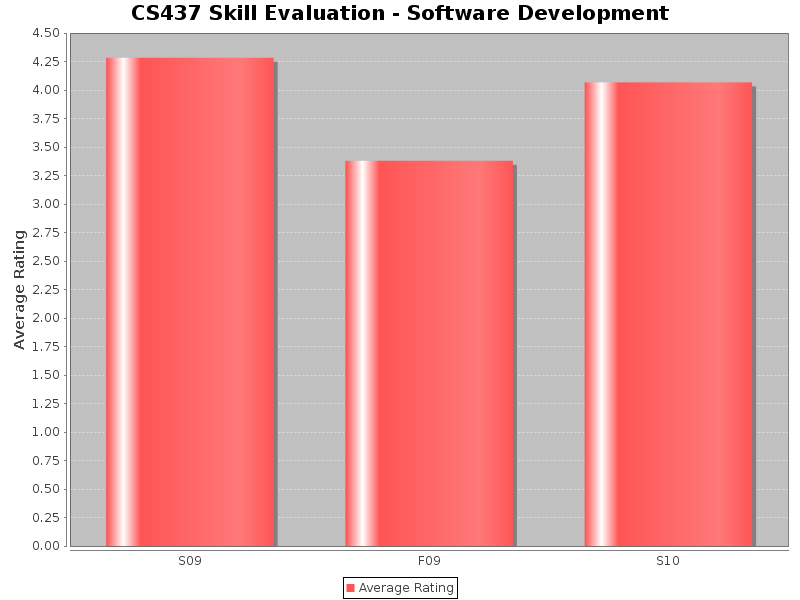


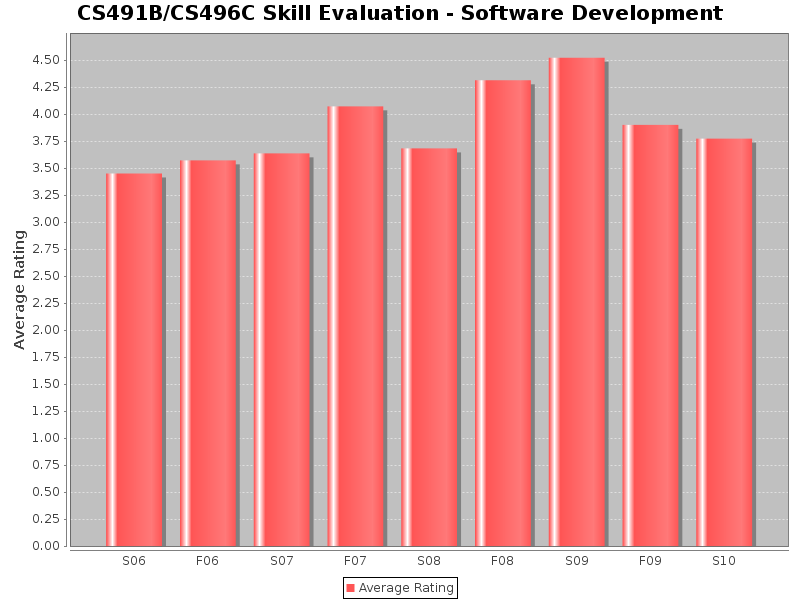
**Figure 4.33: Survey Results**

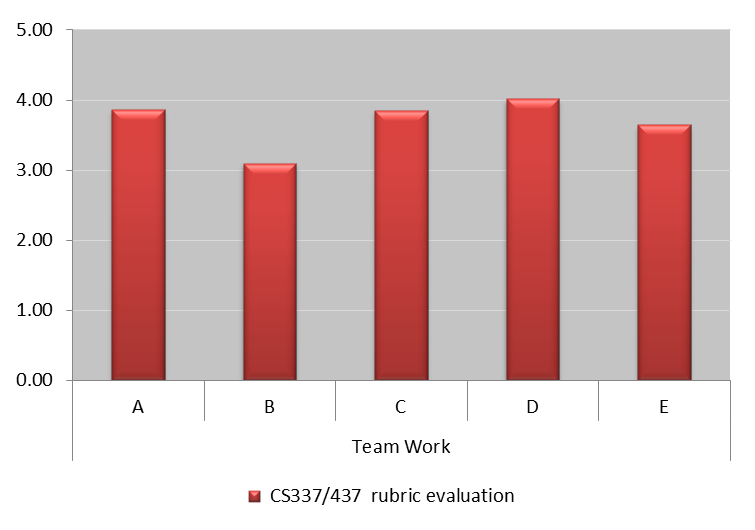
Analysis:

* Faculty skill evaluation of Problem and Requirement Analysis in CS337 and CS491A is satisfactory. Summative evaluations were employed till 2010 (See Figures 4.27 – 4.28).
* Skill evaluation techniques were redesigned in 2011-2012 after the faculty in the Assessment Committee attended an ABET sponsored *Faculty Workshop on Sustainable Assessment Processes*. Performance evaluation rubrics were developed to measure the various performance indicators pertaining to a particular skill evaluation. The results are satisfactory (see Figures 4.29 – 4.23, where A, B, C, D, E refer to the performance indicators defined in the rubrics).
* Rubric evaluation mechanism remains cumbersome and new tools are being developed on CSNS. This will also provide better feedback to the students. More meaningful inferences can also be drawn when the rubric evaluations are employed over a period of time.
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.33).
* Industry sponsorship of senior design projects in CS496 have received considerable positive feedback from students and the Industry Advisory Board.

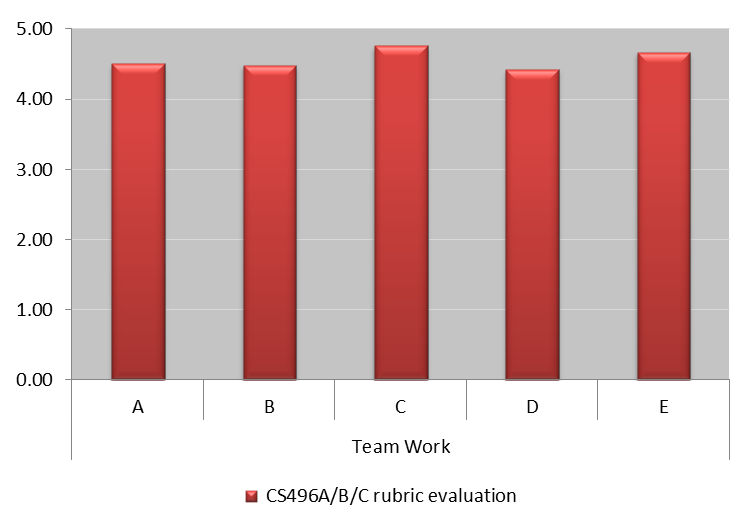
**Student Learning Outcome #6:** *Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*

**Figure 4****.34: CS437 Skill Evaluation**

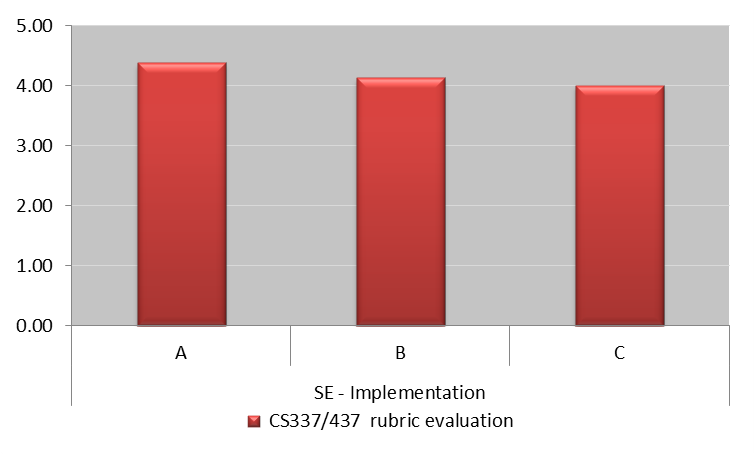
**Figure 4****.35: CS491B Skill Evaluation**

****

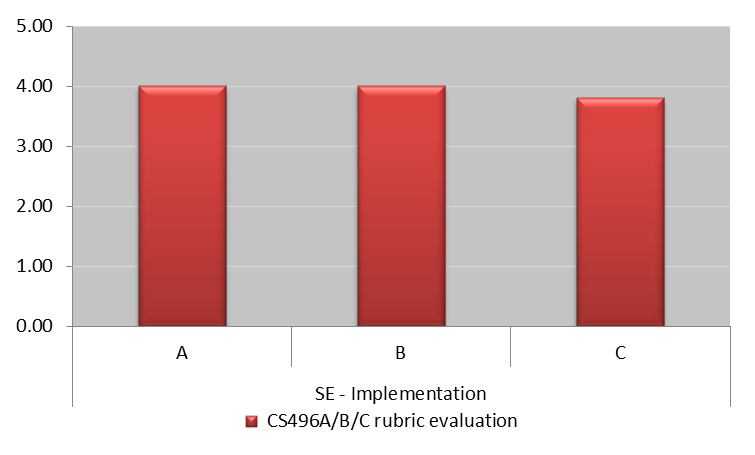
**Figure 4.36: CS437 Rubric Evaluation (SE-Team Work)**

****

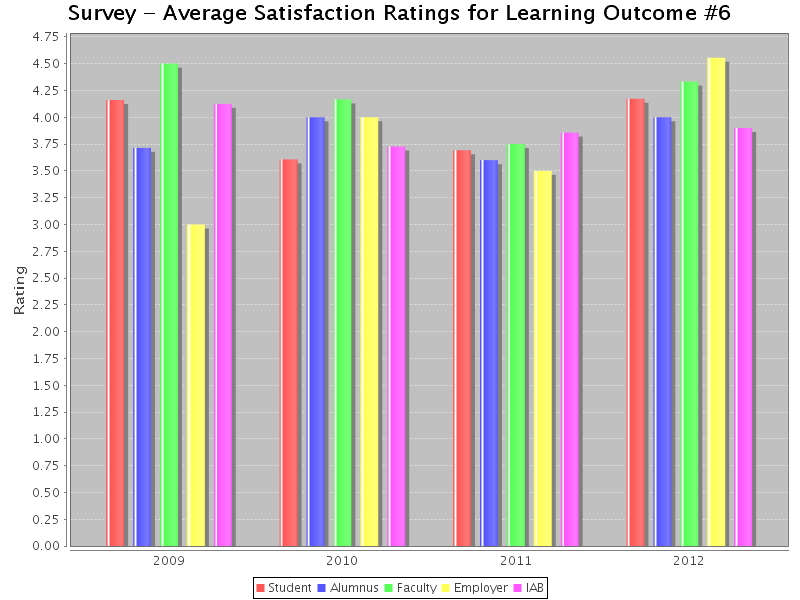
**Figure 4.37: CS496C Rubric Evaluation (SE- Team Work)**

****

**Figure 4.38: CS437 Rubric Evaluation (SE- Implementation)**

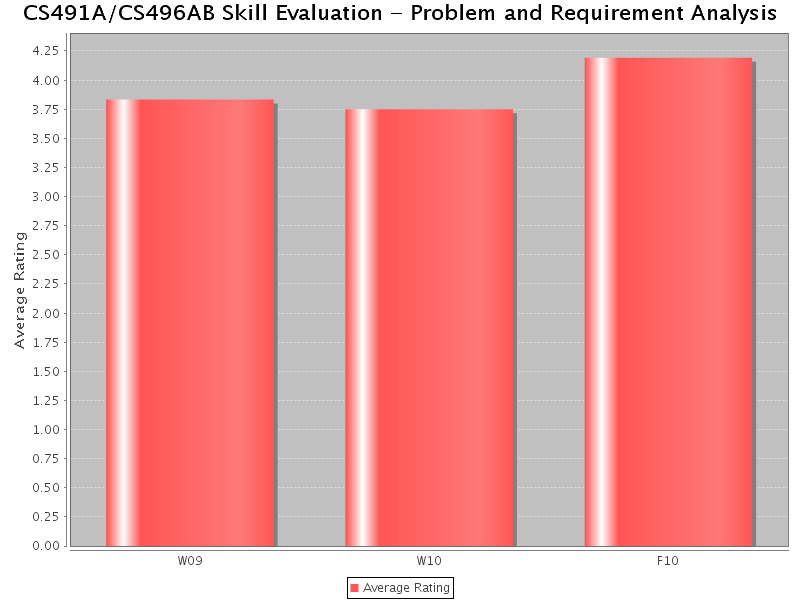
****

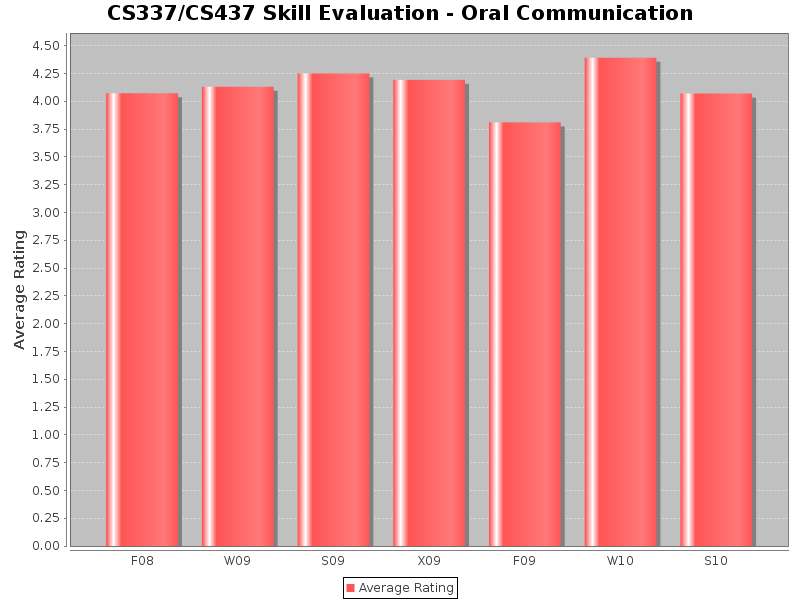
**Figure 4.39: CS496C Rubric Evaluation (SE- Implementation)**



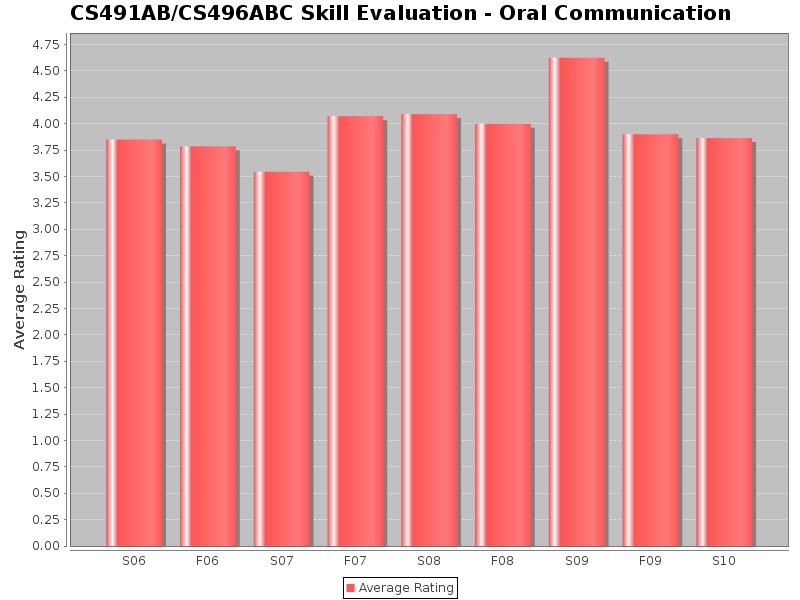
**Figure 4.40: Survey Results**

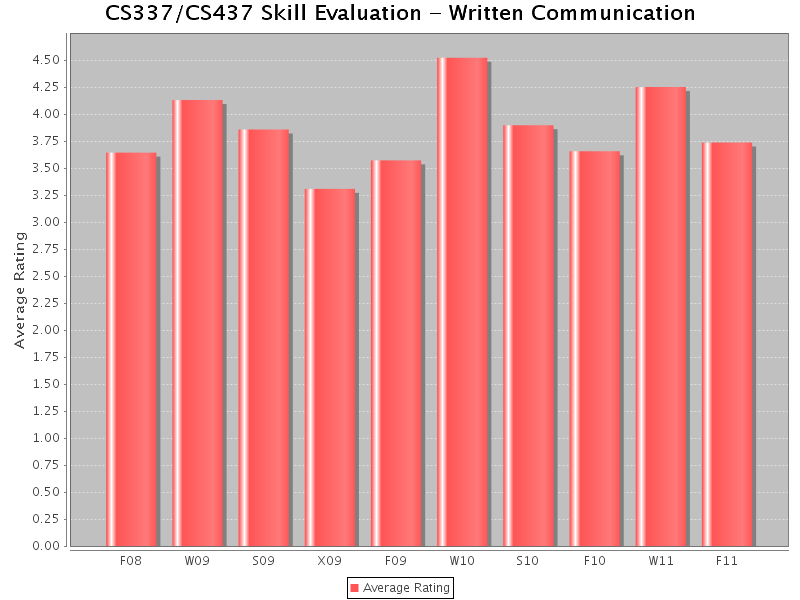
Analysis:

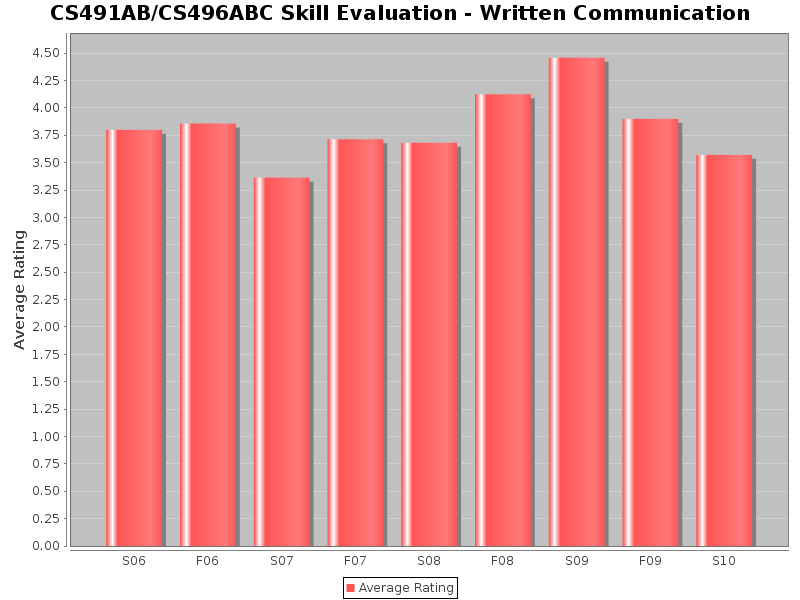
* Faculty skill evaluation of Software Development in CS437 and CS491B/CS496C is satisfactory. Summative evaluations were employed till 2010 (See Figures 4.34 – 4.35).
* Skill evaluation techniques were redesigned in 2011-2012 after the faculty in the Assessment Committee attended an ABET sponsored *Faculty Workshop on Sustainable Assessment Processes*. Performance evaluation rubrics were developed to measure the various performance indicators pertaining to a particular skill evaluation. The results are satisfactory (see Figures 4.36 – 4.39, where A, B, C, D, E refer to the performance indicators defined in the rubrics).
* Rubric evaluation mechanism remains cumbersome and new tools are being developed on CSNS. This will also provide a better feedback to the students. More meaningful inferences can also be drawn when the rubric evaluations are employed over a period of time.
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.40).
* Industry sponsorship of senior design projects in CS496 has received considerable positive feedback from students and the Industry Advisory Board.

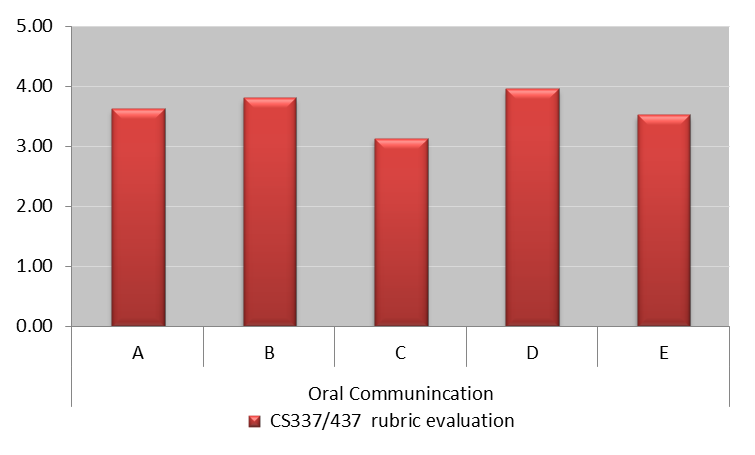
**Student Learning Outcome #7:** *Students will be able to communicate effectively orally and in writing.*

**Figure 4.41: CSCS337/437 Skill Evaluation**

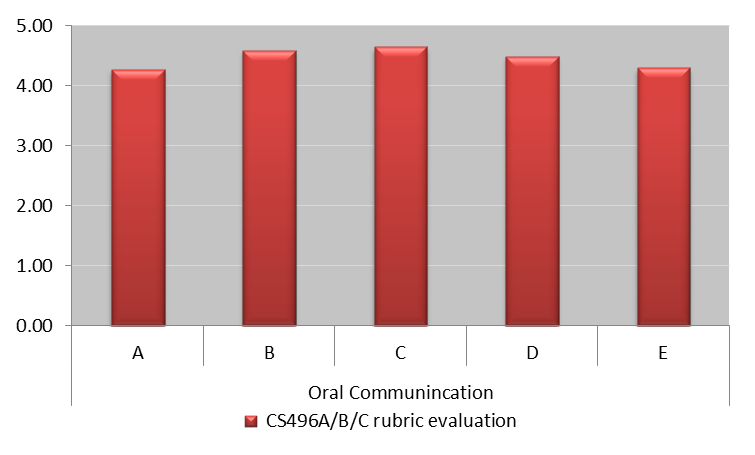
**Figure 4****.42: CS491/CS496 Skill Evaluation**

**Figure 4****.43: CS337/CS437 Skill Evaluation**

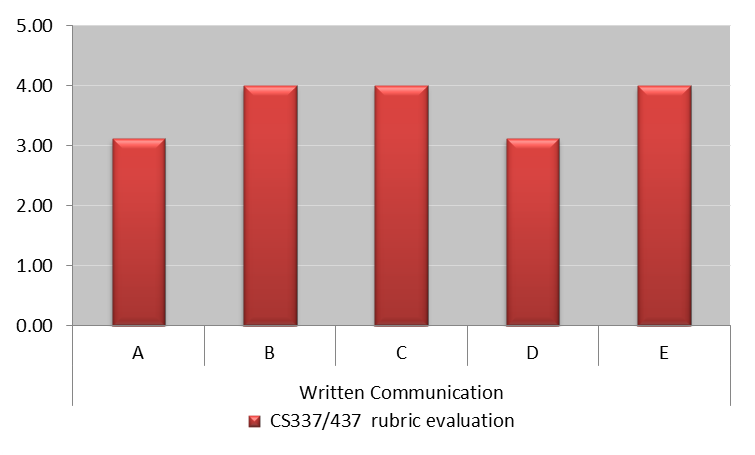
**Figure 4****.44: CS491/CS496 Skill Evaluation**

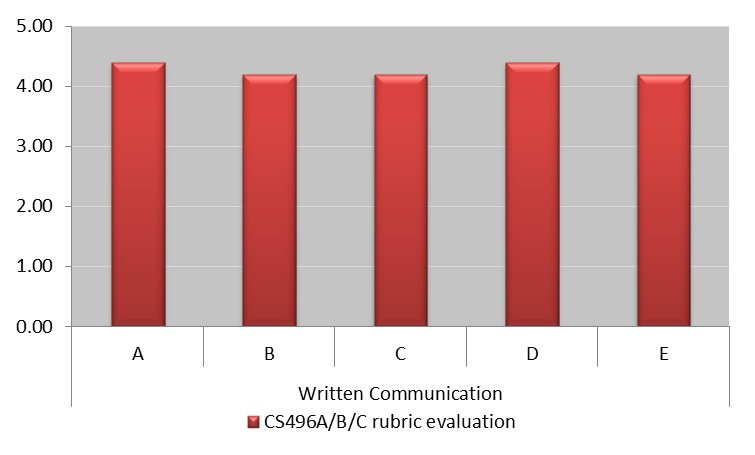


**Figure 4.45: CS437 Rubric Evaluation (Oral Communication)**

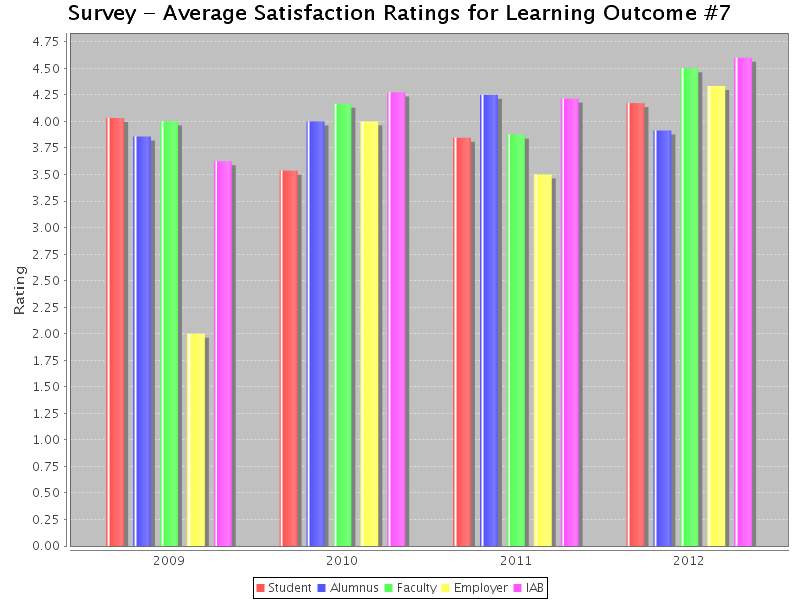
****

**Figure 4.46: CS496C Rubric Evaluation (Oral Communication)**

**Figure 4.47: CS437 Rubric Evaluation (Written Communication)**

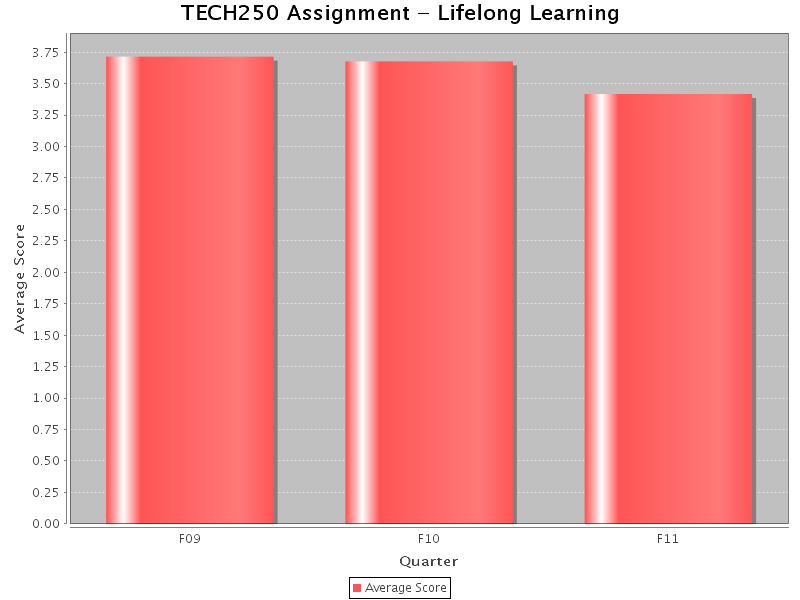
****

**Figure 4.48: CS496C Rubric Evaluation (Written Implementation)**

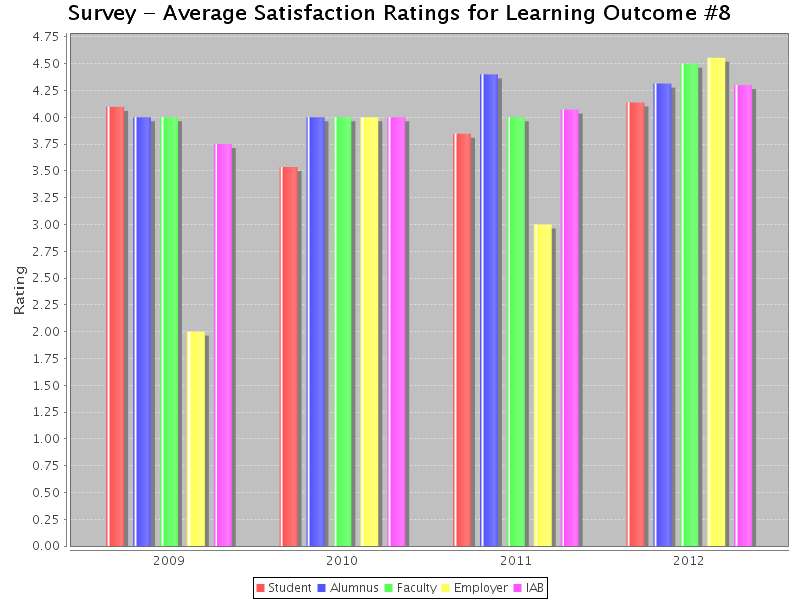
**Figure 4.49: Survey Results**

Analysis:

* Faculty evaluations of the oral and written communication skills are satisfactory (See Figures 4.41 - 4.44).
* Skill evaluation techniques were redesigned in 2011-2012 after the faculty in the Assessment Committee attended an ABET sponsored *Faculty Workshop on Sustainable Assessment Processes*. Performance evaluation rubrics were developed to measure the various performance indicators pertaining to a particular skill evaluation. The results are satisfactory (See Figure 4.45 – 4.48, where A, B, C, D, E refer to the performance indicators defined in the rubrics).
* Rubric evaluation mechanism remains cumbersome and new tools are being developed on CSNS. This will also provide a better feedback to the students. More meaningful inferences can also be drawn when the rubric evaluations are employed over a period of time.
* All students are also required to pass a Writing Proficiency Exam (WPE) which is a prerequisite for CS437 and CS496A. The course syllabi for UNIV400 and UNIV401 provide more detailed rubric evaluations of the WPE. (See Appendix A)
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.49).

**Student Learning Outcome #8:** *Students will have the knowledge, skills, and attitudes for lifelong self-development.*

**Figure 4.50: TECH250 Evaluation**

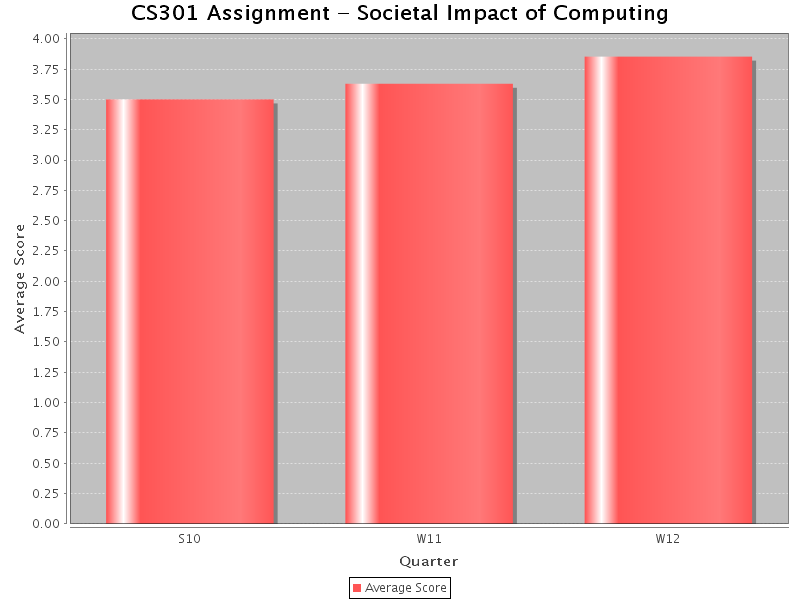


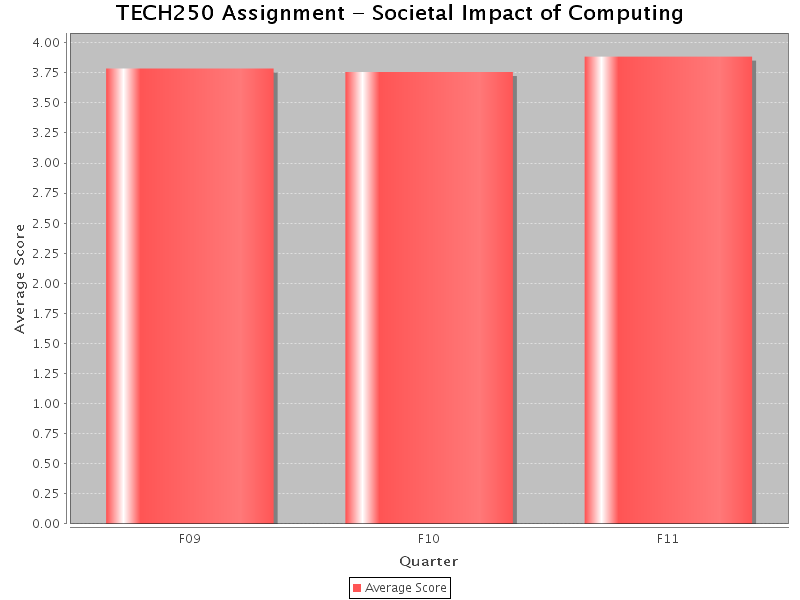
**Figure 4.51: Survey Results**

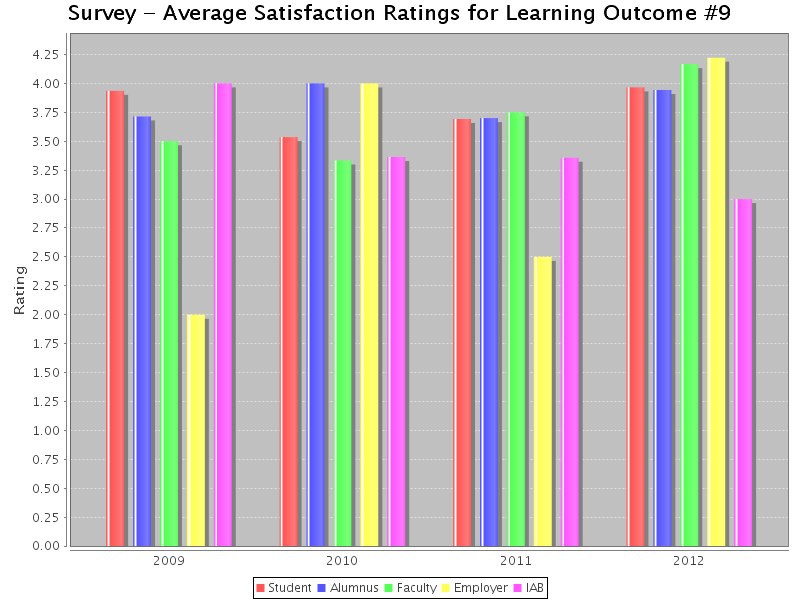
Analysis:

* Faculty evaluation in TECH250 is satisfactory. All students are required to take this class as a part of the “Life Long Understanding” requirement in General Education. Starting in Fall 2010, all Computer Science majors are grouped into the same section to carry out activities relevant to the computer science discipline. (See Figure 4.50)
* CS496ABC provides a very important culminating experience for the students, not only from the point of view of showcasing their acquired skills but also for the opportunities it provides for new learning experiences. Quite often the projects in CS496 are sponsored by external sources and require students to learn new technologies not covered earlier in any classrooms. The tools and techniques learnt during the curriculum give students the means to learn the new technologies required for their projects. As a result students come to recognize the importance of transferring and adapting knowledge received in their studies at CSULA to other environments. Completion of CS496ABC indicates that students have accepted responsibility for learning and that they value it as a lifelong process.
* True to the spirit of lifelong learning, around 20% of the students in the graduating classes from 2009-2011 indicated that they are inclined to consider advanced degrees.
* The local ACM chapter branch (<http://acm.calstatela.edu>) conducts a variety of activities that recognize the principle life-long learning. The organization’s constitution commits it: “To promote an increased knowledge of the science, design, development, construction, language, and applications of modern computing machinery”
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory. (See Figure 4.51)

**Student Learning Outcome #9:** *Students will have the ability to analyze the local and global impact of computing on individuals and society.*

**Figure 4.52: CS301 Evaluation**

**Figure 4.53: TECH250 Evaluation**

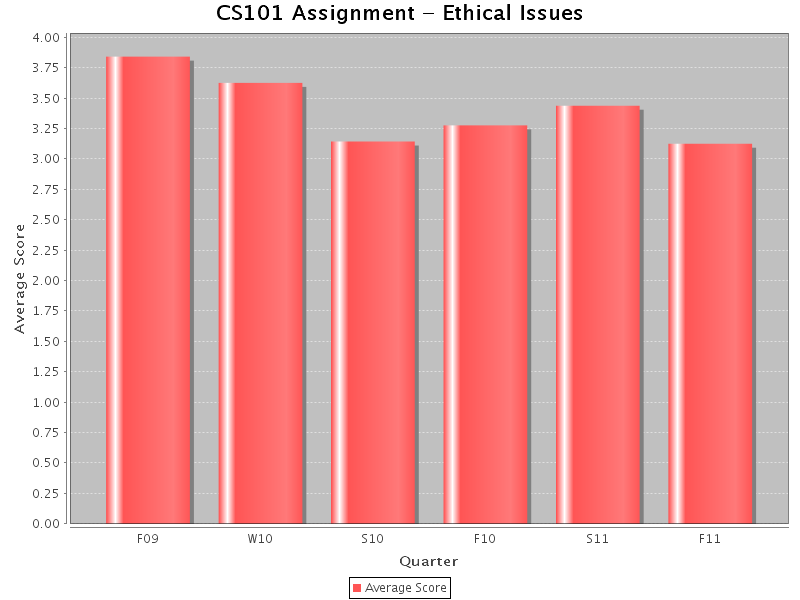


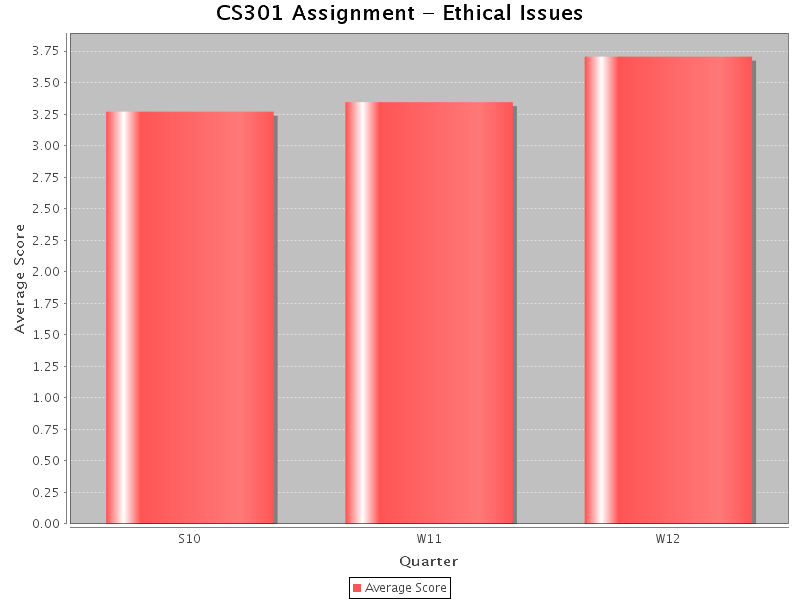
**Figure 4.54: Survey Results**

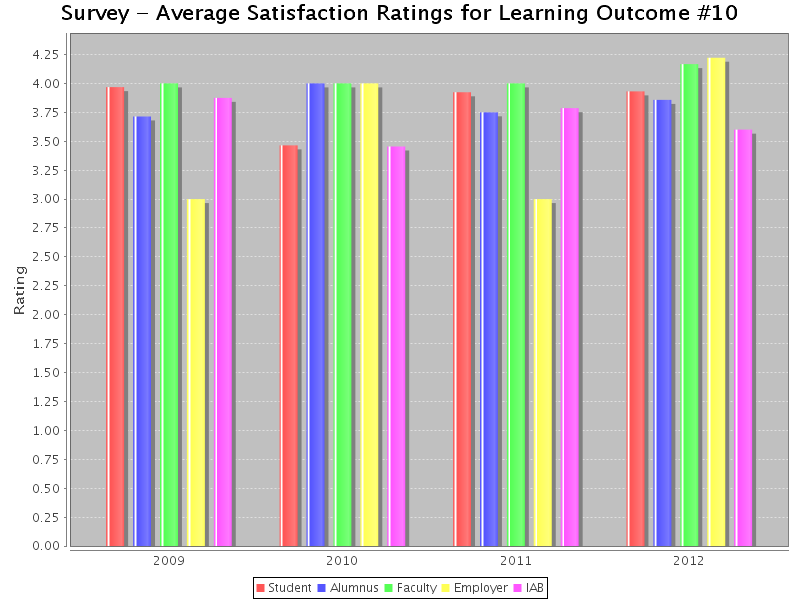
Analysis:

* Faculty evaluation is satisfactory in TECH250 (See Figure 4.52).
* CS301 focuses on two main areas: Ethical issues and Societal impact of computing. The material is presented primarily though selected videos, which then serve as the bases for in-class discussions followed by weekly quizzes. The evaluation by faculty has been satisfactory (See Figure 4.52).
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.54).

**Student Learning Outcome #10:** *Students will have a fundamental understanding of social, professional, ethical, legal, and security issues in computing.*

**Figure 4****.55: CS101 Evaluation**

**Figure 4.56: CS301 Evaluation**

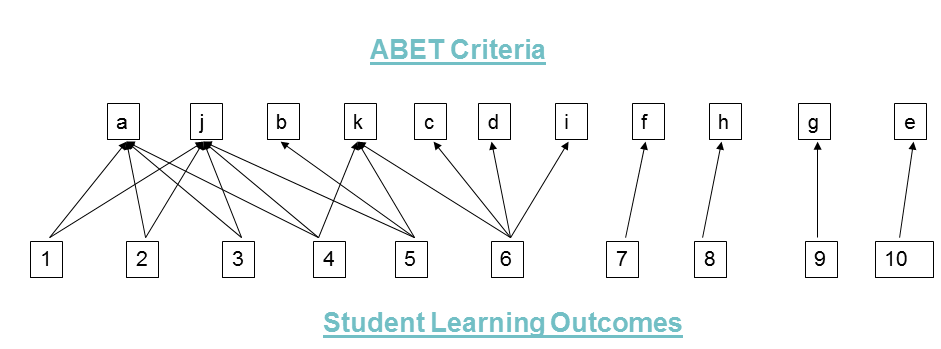
**Figure 4.57: Survey Results**

Analysis:

* Faculty evaluation is satisfactory in both CS101 (See Figures 4.55).
* CS301 focuses on two main areas: Ethical issues and Societal impact of computing. The material is presented primarily though selected videos, which then serve as the bases for in-class discussions followed by weekly quizzes. The evaluation by faculty has been satisfactory (See Figure 4.56).
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.57).

**B-4 Relationship between Student Learning Outcomes and ABET criteria**

Each Program Educational Objective is related through the Student Learning Outcomes to one or more ABET General Criteria. A mapping between the ABET CAC criteria and the Student Learning Outcomes is described in Figure 4.58 and Table 4.10.



**Figure 4.58: ABET criteria – Student Learning Outcomes Map**

| **ABET criteria** | **Student Learning Outcomes** |
| --- | --- |
| (a) An ability to apply knowledge of computing and mathematics appropriate to the discipline, | * Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems. * Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more. * Students will have a strong foundation in the design, analysis, and application of many types of algorithms. * Students will have a fundamental understanding of computer systems. |
| (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution, | * Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions. |
| (c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs, | * Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively. |
| (d) An ability to function effectively on teams to accomplish a common goal, | * Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively. |
| (e) An understanding of professional, ethical, legal, security and social issues and responsibilities, | * Students will have a fundamental understanding of social, professional, ethical, legal, and security issues in computing. |
| (f) An ability to communicate effectively with a range of audiences, | * Students will be able to communicate effectively orally and in writing. |
| (g) An ability to analyze the local and global impact of computing on individuals, organizations, and society, | * Students will have the ability to analyze the local and global impact of computing on individuals and society. |
| (h) Recognition of the need for and an ability to engage in continuing professional development | * Students will have the knowledge, skills, and attitudes for lifelong self-development. |
| (i) An ability to use current techniques, skills, and tools necessary for computing practice. | * Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively. |
| (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices. | * Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems. * Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more. * Students will have a strong foundation in the design, analysis, and application of many types of algorithms. * Students will have a fundamental understanding of computer systems. |
| (k) An ability to apply design and development principles in the construction of software systems of varying complexity. | * Students will have a fundamental understanding of computer systems. * Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions. * Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively. |

**Table 4.10: ABET criteria – Student Learning Outcomes Map**

Each ABET criterion is related to one or more Student Learning Outcomes. By demonstrating the achievement of Student Learning Outcomes, the undergraduate B.S degree program in Computer Science demonstrates its compliance with the ABET program outcomes criteria.

1. **Continuous Improvement**

The evaluation process has resulted in many improvements of the program. Some significant examples are indicated below in Table 4.11.

| # | Continuous Improvement Examples |
| --- | --- |
| **1** | Improvement: CS491AB (two quarter sequence of 6 units) has been modified to CS496ABC (three quarter sequence of 6 units)  Rationale: (i) Student and Faculty feedback indicated that two quarters does not give enough time to complete the demanding projects, especially the projects that are sponsored by the industry. (ii) The three quarter sequence also allows for interdisciplinary projects with the Engineering disciplines. (ii) Industry Advisory Board feedback indicated that a longer timeframe would be beneficial for senior design projects.  Results: This modification was suggested in Fall 2009 and implemented in Fall 2010. This modification has received considerable positive feedback from students and the Industry Advisory Board. |
| **2** | Improvement: Add CS337, a new course, to the core requirements.  Rationale: The Industry Advisory Board (2006 and 2007) has urged that the topic of “Software Requirements/Design” be made distinct from “Software Implementation”. Furthermore, to give real life software engineering experiences, the student group activity should be structured so that the group that develops software requirements be distinct from the group that does the software implementation. This modification addresses ABET concerns #1 and #4.  Results: In 2007, CS437 was modified and expanded into a two quarter CS337-CS437 sequence. CS337 was made a pre-requisite to CS496A. It allows the CS496A students to formulate the Problem and do Requirement Analysis much more effectively than they did before. |
| **3** | Improvement: Modify the CS245 course content from “ Using Operating Systems and Networks: to “Introduction to Computer Organization, Operating Systems and Networks”  Rationale: (i) Student performance on MFT declined from 2007 to unacceptable levels in 2009 as indicated by assessment data for Student Learning Outcome #4. (ii) An internal assessment exam was instituted in CS490, which indicated weakness in the “Computer Systems” area—leading to this modification. (iii) ABET concern #2 indicated that the coverage of “Computer Organization” in EE444 is late in the curriculum. This modification addresses that concern.  Results: CS245 was restructured in 2008. (i) Evaluation of Student Learning Outcome #4 has improved considerably. Starting in 2010, the results are evident in both the MFT exam and the CS490 internal assessment exam. |
| **4** | Improvement: Modify CS301 delivery mechanism  Rationale: (i) Existing format of presentations and debates was not allowing enough time for a complete dissemination of the material. A student was not getting more than two opportunities to make a presentation on one of the sub-topics. (ii) Restructuring the course to provide material using documents and videos now provides students with comprehensive information on “Ethical issues” and “Societal impact of computing”. (iii) This also allows direct evaluation of SLO #9 and SLO #10 using a number of multiple-choice quizzes.  Result: CS301 was restructured in Spring 2009. (i) There has been an improvement in student performance for both Student Learning Outcomes concerning SLO #9 & SLO#10. (ii) This also addresses ABET concern #1. |
| **5** | Improvement: Cohort Computer Science majors in TECH250  Rationale: (i) TECH 250 is a General Education course and is taken by students from across all disciplines in the university. Since TECH250 provides important data for SLO #8 and SLO #9, it would be useful to target a section only for Computer Science majors. Topical coverage could be made more relevant for Computer Science majors.  Result: This was implemented starting Fall 2009. Relevant TECH faculty has been trained in using CSNS. |
| **6** | Improvement: Coordinate MATH248 (Discrete Mathematics) and CS312 as a sequence.  Rationale: (i) There is considerable room for improvement concerning Student Learning Outcomes (#1, #3) in both the MFT and the CS490 internal exam. (ii) Faculty feedback also indicates that CS312 is an important course and needs to be coordinated with MATH248.  Result: (i) Dr. Akis, a joint faculty from Computer Science and Mathematics, has restructured the syllabus for MATH248 to better suit the Computer Science program. (ii) An alternative is to convert MATH248 to CS248 thereby offering the CS Department better control over the topical coverage. |
| **7** | Improvement: Survey responses data collection on CSNS  Rationale: Survey responses from various constituencies provide valuable feedback. Alumni information at the Department and the institutional level were very spotty. As a result, the survey responses have been inconsistent.  Result: Alumni data has been imported into CSNS and an alumni group has been created in 2010. |
| **8** | Improvement: Learning outcomes graphs on CSNS  Rationale: Each of the Student Learning Outcomes derives data from various measures. It has not been an easy task to organize the data and visually represent the graphs for each outcome.  Result: New tools were implemented on CSNS to extract the assessment data and represent them as graphs in 2010-2011. |
| **9** | Improvement: Rubrics implementation for Skill Evaluations  Rationale: (i) Skill Evaluations is a direct measure that allows faculty to directly observe a student's demonstration of certain skills using certain performance indicators. Prior to 2011, we employed a summative assessment of skills that were determined by the faculty. This practice was stopped after the faculty in the Assessment Committee attended an ABET sponsored *Faculty Workshop on Sustainable Assessment Processes*. More meaningful inferences can also be drawn when these rubric evaluations are employed.  Results: The rubrics were developed and employed Spring 2011. |
| **10** | Improvement: Rubrics implementation in CSNS.  Rationale: The results of rubric evaluations in 2011-2012 are satisfactory even though the evaluation mechanism was cumbersome. Currently, the rubrics data collection and analysis must be done manually, which puts a heavy burden on the faculty and staff. New tools need to be developed for rubric evaluations on CSNS. By automating or semi-automating this process, the faculty can devote more time and resources to provide timely feedback to the students.  Results: New tools need to be designed on CSNS in the 2012-2013 assessment cycle. |

**Table 4.11: Examples of Continuous Improvement**

1. **Additional Information**

**D-1 Assessment Reports**

Results of the annual internal assessment process are compiled into an Assessment Report on a two year cycle as shown below:

* Assessment Report, 2010-2012 (Spring 2012)
* Assessment Report, 2008-2010 (Spring 2010)
* Assessment Report, 2006-2008 (Spring 2008)
* Assessment Report, 2005-2006 (Spring 2006)
* Assessment Report, 2004-2005 (Spring 2005)
* Assessment Report, 2002-2004 (Spring 2004)
* Assessment Report, 2001-2002 (Spring 2002)

Assessment Report 2010-2012 contains more recent data that was collected in 2012 after submitting the Self Study Report. The three recent Assessment Reports covering the six years since the previous ABET visit are included as a separate document and will be provided to the ABET evaluators.

All reports on CSNS are also accessible online at <http://csns.calstatela.edu/wiki/content/assessment/documents/> for ABET evaluators online by logging in as user “abet” with a password “Status”.

**D-2 Constituency Member Comments**

Constituency surveys are conducted every year. A few pertinent comments received over the last five years are given below:

Industry Advisor Board/Employer Comments:

* *The most important skills in a work environment: "communication", “experience with large projects and working with others”.*
* *Developing problem solving skill is a key in the workplace.*
* *Offer new courses in “Mobile Applications”, “Embedded Systems”, “Information Assurance”.*
* *Offer courses and projects related to current market demands.  Like Android and IOS programming.  That helps the students chances of finding jobs and staying up to speed with the times.*
* *I believe that requiring an internship program will give real world experience within the field and would help in obtaining a job soon after graduation.*
* *It seems that the area of game design and implementation may be a very promising one to add to the entire program*
* *It will also be beneficial for students to have some knowledge of IT*
* *Web technology is growing and growing...are we doing enough to prepare our students both hardware and software wise to handle this most important challenge?*
* *Couple students with industry during studies as exemplified by the senior design projects.*
* *The senior project sequence is a large contributor to many of the elements described by the Objectives/Outcome....tying many diverse elements together.  The senior project in large software projects using systems engineering principles is a particularly good method of integrating these concepts.*
* *My response is based solely on the student design presentations given at the IAB meeting:  I was impressed with the design projects and the student presentations.  The students seemed conversant in not only the language of CS, but also in the language of the specific design projects with which they participated.  All students seemed to be engaged in the presentations.  The process of the senior design projects seemed to emulate real-world situations, and should serve the students well*
* *The students need to learn more about resource limitation.  How to develop software given system constraints like memory, disk space, network bandwidth, etc.*
* *Life long learning is very critical. It is not as important to ENTER the workforce as an established professional but it is Important that they have the skills to learn that quickly based on what they learned at the university.*
* *The field of computer science is always rapidly changing. It's more important to teach students how to learn and enforce good habits (communication, design, collaboration, etc...)*
* *I was very impressed by the selection of student projects this year. I think the students benefit tremendously from getting to work with real-world organizations for their senior projects. I am pleased with the program improvements over the past five years.*
* *Students that are not afraid to take on new challenges and can communicate their ideas and progress will have an advantage.*
* *The ability to communicate is critical.  This includes the ability to give presentations and understanding presentation structures.*
* *While being competent and having the appropriate skill set is important and required, the singular most important skill is good communication. Respect and a positive attitude also tend to be found in those who communicate well. Also being able to learn is probably more important than already knowing all the appropriate technologies, particularly for those who are just entering the industry from school.*
* *The ability to communicate technical concepts and ideas is a major skill that I look for.  I need people that are not only technically competent but can also communicate their ideas to others.  They need to be able to write reports, specifications, and documents using complete sentences and ideas.  I believe this skill is as important as their ability to design and generate code.*
* *In addition to programming, an understanding of hardware and the interaction of instrumentation on networks is critical.*
* *My top list of important skills would be as follows:*
  + *A clear understanding of software engineering problems and approaches concerning large programming efforts.*
  + *Excellent ("good"/"very good" is not good enough) programming skills in at least TWO classical programming languages (Java, C++)*
  + *Ability to use or get up to speed in a short period of time with new languages: C#, Pearl, Piton, etc...*
  + *Ability to understand pseudo code and translate it into a classical programming language.*
  + *Web design/programming ability.*
  + *Knowledge and techniques regarding computer security, databases, Operating System, Compilers, Interpreters.*
  + *Understanding of mobile devices' software*
  + *Knowledge/awareness of new upcoming technologies*

Alumni Comments:

* *My stay at CSULA was a great learning experience in terms of education and coursework especially in the field of computer science. I liked that one of the last courses taken was a recap of Computer Science topics in preparation for the MFT. It served a good review of all the concepts learned over the past years and kept them fresh in mind.*
* *The advisor was a real advocate for the students and was very committed to each student having an excellent outcome.  I appreciated the small class size and that we had access to our professors.   This was a far cry from my masters program at USC with classes of 150 students where teaching assistants were the main resource in case we had questions.  I had several exceptional professors. I feel that the CS program at CALSTATE LA gave me the foundation to succeed at my current job at the Aerospace Corp.*
* *I graduated from the undergraduate Computer Science (CS) program at Cal State University, Los Angeles (CSULA) in June 2006. In May 2007, I graduated from Cornell University with a Masters of Engineering degree in CS, and expect to graduate from Cornell with a PhD in CS in May 2013. While CSULA did not offer large academic research opportunities like those found at Cornell, the fundamental undergraduate education I received at CSULA was on par with equivalent offerings at Cornell, with notably smaller class sizes to boot.*
* *Cal State LA has given me the skills needed to develop applications for the Android platform.   The knowledge base needed for this type of work is very broad.  It would not be possible for me to self-learn Android development without, the extensive knowledge gleaned from computer science department.  The fact that I am able to quickly learn Android development, without a formal class, is a testament of Cal State’s ability to prepare students to adapt to emerging technologies.  Thank you for opening new possibilities for me!*
* *My experience at CSULA has been nothing but positive.  I believe the curriculum designed by the CS department has provided me with all the tools necessary to become a successful and thriving professional.  The staff was open to any and all questions and treated me with respect and kindness.  What I most appreciated from the curriculum was how diverse it was in respect to the many different programming languages we were exposed to.  My first job after graduation I had an opportunity to build an application, but had to write code in a language I had no experience with.  This was not an obstacle as I was comfortable having to learn quickly in order to produce a product.  In addition, the curriculum outside of the CS department, namely mathematics and physics, sharpened my ability to think both critically and perform analysis of complex problems. I am thankful for my education and for the CS department.*
* *First of all, I would like to say that I greatly appreciated all your effort and kindness for teaching us. After I graduated, I pursed the advanced degree. Currently, I am on PhD program from UCLA, but the fundamental knowledge and skills that I acquired during my school years from California State University, Los Angeles provided me with the strong background to continue my education. Again, thank you very much!*
* *In the late '80s, it was a commuter school with no sense of unity.  I have had ties with many students coming out since and have seen great improvement in all areas.  I feel that the students graduating today from CSULA are superior to me when I graduated.*
* *I have witnessed firsthand the quality and breadth of the graduates from the BS program.  They are excellent engineers right out of the gate.*
* *I am currently a systems developer for a small consulting firm. All my colleagues in my department/firm are also CSULA alumni. However, I am the only Computer Science major (the few, the proud, the computer scientists) ;).*
* *I've been working for Disney Interactive Labs since August 2011. We just launched* [*Disney Video*](https://email.calstatela.edu/exchweb/bin/redir.asp?URL=http://video.disney.com/) *in April 2012. Jay Donnell and Mark Luntzel work there as well. The CSULA CS grads are taking over!*

# CRITERION 5. CURRICULUM

### Program Curriculum

The Computer Science curriculum provides students with basic knowledge, training, discipline, and skills, as defined by the Computer Science Program Student Learning Outcomes. Through its lower division (CS100 & CS200 level) required courses, the curriculum provides students with the basic mathematical and science framework. Through its upper division (CS300 & CS400 level) required courses, the curriculum builds upon the fundamental principles of computer science for more advanced study. Through its upper division technical electives, students gain additional breadth and/or depth in computer science by an appropriate selection of courses. Through its Capstone courses students demonstrate their abilities to apply the knowledge and skills they acquired. The curriculum is thus consistent with the defined Program Educational Objectives and Student Learning Outcomes.

Since the university operates on an 11-week quarter system (including a final examination week), all courses are delivered using a formula where one quarter unit means one hour of lecture or three hours of laboratory per week.

**A-1 Course requirements**

Table 5-1 describes the suggested plan of study (by year and quarter term) for students in the computer science program. Depending on the summer schedules, students can elect to spread the load over four quarters rather than the three quarter schedule indicated in Table 5.1. For a variety of reasons many students formulate their own schedules with varying loads in any given quarter. An advisor is available to help students plan or adjust their schedules. Students select courses for the General Education (GE) blocks (C,D, and E) from a variety of courses. Similarly, students select upper division (CS300/CS400) courses as electives. A total of 66 units is required in the Computer Science core (lower division + upper division) requirements with an additional 24 units of electives in the upper division requirements.

Table 5.1 also includes additional information concerning scheduling and average section enrollments for all courses in the program over the two years immediately preceding the visit. Also note that the Credit Hours are indicated by lecture + laboratory units. For example:

* The Credit Hours for CS201 is described as 4+1 implying 4 lecture units (4 hours per week) + 1 laboratory unit (3 hours per week).
* The Credit Hours for CS490 is described as 0+2 implying 2 laboratory units (6 hours per week).

Similar information concerning elective courses is indicated in Table 5.2.

| **Course**  **(Dept., Number)** | **Required (R) or**  **Elective (E)** | **Math & Basic**  **Sciences**  **(Credit Hours)** | **Computer Science**  **Core**  **(Credit Hours\*)** | **General Ed**  **(Credit Hours)** | **Electives**  **(Credit Hours)** | **Last Two Quarters the Course was Offered:** | **Average Enrollment**  **(Last Two Quarters)** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Year #1 (Fall Qtr)** | | | | | | |  |
| CS101 | R |  |  | 2 |  | Sp’12, F’11 | 30 |
| MATH206 | R | 4 |  |  |  | Sp’12, W’12 | 30 |
| ENGL101 | R |  |  | 4 |  | Sp’12, W’12 | 30 |
| GE E + | R |  |  | 4 |  |  |  |
| **Year #1 (Winter Qtr)** | | | | | | |  |
| CS120 | R |  | 2+1 |  |  | Sp’12, W’12 | 30 |
| MATH207 | R | 4 |  |  |  | Sp’12, W’12 | 30 |
| ENGL102 | R |  |  | 4 |  | Sp’12, W’12 | 30 |
| HIS202A/B | R |  |  | 4 |  | Sp’12, W’12 | 30 |
| **Year #1 (Spring Qtr)** | | | | | | | |
| CS122 | R |  | 2+1 |  |  | Sp’12, W’12 | 30 |
| MATH208 | R | 4 |  |  |  | Sp’12, W’12 | 30 |
| COMM150 | R |  |  | 4 |  | Sp’12, W’12 | 30 |
| POLS150 | R |  |  | 4 |  | Sp’12, W’12 | 30 |
| **Year #2 (Fall Qtr)** | | | | | | | |
| CS201 | R |  | 4+1 |  |  | Sp’12, W’12 | 30 |
| MATH248 | R | 4 |  |  |  | Sp’12, W’12 | 30 |
| GE C1+ | R |  |  | 4 |  |  |  |
| PHYS101/  PHYS 211 | R | 4 (or 5) |  |  |  | F’11, F’10  Sp’12,W’12 | 30  30 |
| **Year #2 (Winter Qtr)** | | | | | | | |
| CS202 | R |  | 4+1 |  |  | Sp’12, W’12 | 30 |
| MATH255 | R | 4 |  |  |  | Sp’12, W’12 | 30 |
| GE C2+ | R |  |  | 4 |  |  |  |
| PHYS102/  PHYS 212 | R | 4 (or 5) |  |  |  | W’12, W’10  Sp’12,W’12 | 30  30 |
| **Year #2 (Spring Qtr)** | | | | | | | |
| CS203 | R |  | 4+1 |  |  | Sp’12, W’12 | 30 |
| MATH270 | R | 4 |  |  |  | W’12, W’11 | 30 |
| CS245 | R |  | 2+1 |  |  | Sp’12,F’11 | 30 |
| PHYS103/  PHYS 213 | R | 4 (or 5) |  |  |  | Sp’12, Sp’11  Sp’12,W’12 | 30  30 |
| **Year #3 (Fall Qtr)** | | | | | | | |
| CS312 | R |  | 4 |  |  | Sp’12, F’11 | 31 |
| CS332F | R |  | 1+1 |  |  | F’11, F’10 | 35 |
| CS337 | R |  | 2+1 |  |  | Sp’12, F’11 | 22 |
| GE C3 | R |  |  | 4 |  |  |  |
| **Year #3 (Winter Qtr)** | | | | | | | |
| CS Elective #1 | E |  |  |  | 4 | (Table 5.2) |  |
| CS301 | R |  | 0+1 |  |  | W’12,W’11 | 32 |
| CS332C | R |  | 1+1 |  |  | W’12, W’11 | 34 |
| CS320 | R |  | 2+1 |  |  | Sp’12, F’10 | 30 |
| CS437 | R |  | 4+1 |  |  | W’12, W’11 | 26 |
| **Year #3 (Spring Qtr)** | | | | | | | |
| GE D+ | R |  |  | 4 |  |  |  |
| CS332L | R |  | 1+1 |  |  | Sp’12,Sp’11 | 35 |
| CS386 | R |  | 4 |  |  | Sp’12, F’10 | 27 |
| CS Elective #2 | E |  |  |  | 4 | (Table 5.2) |  |
| CS Elective #3 | E |  |  |  | 4 | (Table 5.2) |  |
| **Year #4 (Fall Qtr)** | | | | | | | |
| CS496A | R |  | 0+2 |  |  | F’11, F’10 | 23 |
| EE444 | R |  | 4 |  |  | Sp’12,Sp’09 | 26 |
| CS Elective #4 | E |  |  |  | 4 | (Table 5.2) |  |
| GE UD Theme #1\*\* + | R | 4 |  |  |  |  |  |
| **Year #4 (Winter Qtr)** | | | | | | | |
| CS496B | R |  | 0+2 |  |  | W’12, W’11 | 22 |
| CS440 | R |  | 4 |  |  | W’12, W’11 | 28 |
| CS Elective #5 | E |  |  |  | 4 | (Table 5.2) |  |
| GE UD Theme #2+ | R |  |  | 4 |  |  |  |
| **Year #4 (Spring Qtr)** | | | | | | | |
| CS496C | R |  | 0+2 |  |  | Sp’12,Sp’11 | 18 |
| CS490 | R |  | 2 |  |  | Sp’12,Sp’11 | 19 |
| CS Elective #6 | E |  |  |  | 4 | (Table 5.2) |  |
| GE UD Theme #3+ | R |  |  | 4 |  |  |  |
| **TOTAL UNITS** |  | **40 - 43** | **66** | **50** | **24** |  |  |
| **% OF TOTAL (approx.)** |  | **22% 3.3%1.6%Sort offer new programsred by various departments with average class size of 25.ck to heir PhD** | **37%** | **28%** | **13%** |  |  |

\*: Credit Hours are described as Lecture + Laboratory units.

+: General Education courses offered almost every quarter

\*\*: General Education Class offered in the area of Natural Sciences and Mathematics

**Table 5.1: Curriculum (Required courses)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Course**  **(Dept., Number)** | **Required (R)**  **or Elective (E)** | **Computer Science**  **Electives**  **(Credit Hours)** | **Last Two Quarters the Course was Offered:** | **Average Enrollment**  **(Last Two Quarters)** |
| CS340 | E | 4 | W’11, W’07 | 16 |
| CS345 | E | 4 | W’12, Sp’10 | 23 |
| CS370 | E | 4 | F’11, F’10 | 15 |
| CS422 | E | 4 | Sp’12, F’11 | 19 |
| CS447 | E | 4 | F’11, F’10 | 24 |
| CS450 | E | 4+1 | W’12, F’10 | 15 |
| CS454\* | E | 4 | W’12, F’11 | 19 |
| CS460 | E | 4 | Sp’12, Sp’11 | 16 |
| CS461 | E | 4 | F’11, Sp’11 | 10 |
| CS470 | E | 4 | W’12, W’11 | 20 |
| CS480 | E | 4 | F’11, W’11 | 24 |
| CS486 | E | 4 | W’12, W’11 | 15 |
| CS488 | E | 4 | Sp’12, Sp’11 | 14 |

\*CS454 has been offered on different topics – *3D Computer Game Programming, User Interfaces and Interactions, Mobile Computing.* All of them can be used to fulfill the electives.

**Table 5.2: Curriculum (Electives)**

**A-2 Course prerequisite(s) requirements**

Major requirements, with prerequisites, are described in Table 5.3. A grade of "C" or better is required in all prerequisite coursework. Prerequisites are strictly enforced on the GET system at the time of registration.

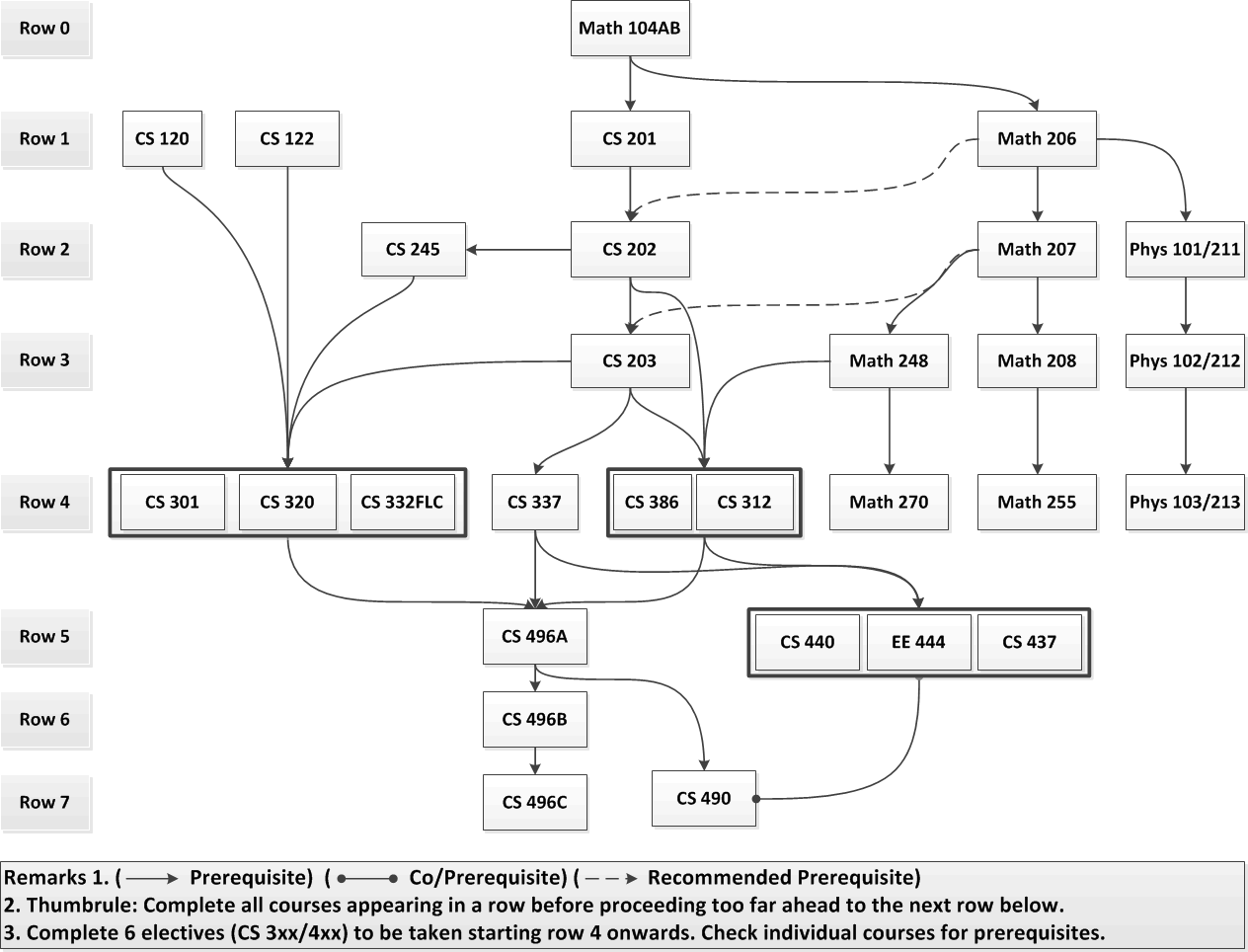
| **Course** | **Description** | **Prerequisite(s)** |
| --- | --- | --- |
| CS 120 | Introduction to Web Site Development (2 lecture units + 1 laboratory unit) | Computer Literacy |
| CS 122 | Using Relational Databases and SQL (2 lecture units + 1 laboratory unit) | Computer Literacy |
| CS 201 | Introduction to Programming (4 lecture units + 1 laboratory unit) | MATH 103/104B or  consent of the Instructor |
| CS 202 | Introduction to Object Oriented Programming (4 lecture units + 1 laboratory unit) | CS 201  Recommended: MATH 206 |
| CS 203 | Programming with Data Structures (4 lecture units + 1 laboratory unit) | CS 202  Recommended: MATH 207, 248 |
| CS 245 | Introduction to Computer Organization, Operating Systems and Networks (2 lecture units + 1 laboratory unit) | CS 202 |
| MATH 206 | Calculus I (4 lecture units) | MATH 102/104A and103/104B  both with min. C grade |
| MATH 207 | Calculus II (4 lecture units) | MATH 206 with minimum C grade |
| MATH 208 | Calculus III (4 lecture units) | MATH 207 with minimum C grade |
| MATH 248 | Discrete Mathematics (4 lecture units) | MATH 207 |
| MATH 255 | Introduction to Matrix Theory (4 lecture units) | MATH 208 |
| MATH 270 | Introduction to Probability and Statistics(4 lecture units) | MATH 208 |
| PHYS 101 or PHYS 211 | General Physics I (4 lecture units) or Mechanics (5 lecture units) | MATH 102/MATH 104A &  MATH 103/MATH 104B |
| PHYS 102 or PHYS 212 | General Physics II (4 lecture units) or Waves (5 lecture units) | PHYS 101 or PHYS 211 |
| PHYS 103 or PHYS 213 | General Physics III (4 lecture units) or Electricity (5 lecture units) | PHYS 102 or PHYS 212 |
| CS 301 | Computer Ethics in the Information Age (1 laboratory unit) | CS 203 |
| CS 312 | Data Structures and Algorithms (4 lecture units) | CS 203, MATH 208, MATH 248 |
| CS 320 | Web and Internet Programming (2 lecture units + 1 laboratory unit) | CS 120, CS 122, CS 203 |
| CS 332F | Functional Programming (1 lecture unit + 1 laboratory unit) | CS 203 |
| CS 332L | Logic Programming (1 lecture unit + 1 laboratory unit) | CS 203 |
| CS 332C | C++ Object Oriented Programming (1 lecture unit + 1 laboratory unit) | CS 203 |
| CS 337 | Software Design (2 lecture units + 1 laboratory unit) | CS 203 |
| CS 386 | Introduction to Automata Theory (4 lecture units) | CS 202, MATH 248 |
| CS 437 | Software Engineering (4 lecture units + 1 laboratory unit) | CS 312, 337, and WPE |
| CS 440 | Introduction to Operating Systems (4 lecture units) | CS 245 & 312 |
| EE 444 | Computer Architecture (4 lecture units) | CS 245 |
| CS 490 | Computer Science Recapitulation (2 lecture units) | CS 386, 437, 440, EE 444, MATH270 |
| CS 496A | Software Design Laboratory (2 laboratory units) | CS 312, 320, 337, 386, and senior standing |
| CS 496B | Software Design Laboratory (2 laboratory units) | CS 496A |
| CS 496C | Software Design Laboratory (2 laboratory units) | CS 496B |

**Table 5.3: Curriculum (with prerequisites)**

**A-3 Prerequisite flowchart**

The prerequisite flow chart for the B.S. in Computer Science curriculum is depicted in Figure 5.1. The courses are structured with the lower division requirements shown in Row1-Row3. The advanced required CS courses are shown in Row4-Row7. Elective courses are not shown in Figure 5.1 and are described in the course syllabi included in Appendix A. However, based on the general nature of the prerequisites for these upper division elective courses, they can be scheduled between Row4 and Row7.

As a general rule, students are advised to finish the courses in a particular row before progressing to the next row. Table 5.1 contains an ideal road map that adheres strictly to the prerequisites and scheduling patterns of the Department. This ideal roadmap provides a pathway for students to complete their undergraduate requirements in four years.



**Figure 5.1: Flow Chart of required courses**

**A-4 Capstone courses**

Students are required to complete a senior design project sequence (CS496ABC) and to take a recapitulation course (CS490) during their senior year. We use these two senior capstone experiences as a primary instrument in program assessment.

CS496ABC

This is a three-quarter senior design laboratory project sequence in which each student participates in a group project under the supervision of a faculty advisor. The goals of the course are

* To improve students’ ability to undertake complex software projects.
* To require students to learn new technologies and concepts on their own.
* To improve students’ oral communication skills through presentations and interaction with project stake holders.
* To improve students’ written communication skills through the writing of project documents.

These course goals contribute to the success of Student Learning Outcomes (SLO):

* SLO5. Students will have the training to analyze problems and to identify and define the computing requirements appropriate to their solutions.
* SL06. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.
* SLO7. Students will be able to communicate effectively orally and in writing.
* SLO8. Students will gain the knowledge, skills, and attitudes for lifelong self-development.

At the end of CS496C, the teams show off their projects at Senior Design Day—an event organized at the College level across all disciplines. Each project team prepares a poster and presents its project (<http://csns.calstatela.edu/projects.html>). The event is attended by the members of the Department’s Industry Advisory Board and the liaisons of the industry sponsored projects. This gives students a sense of what it means to work in a professional environment where customers expect the delivery of development products. At the same time the projects are crafted to meet the educational goals of the capstone design course, which runs the full senior year.

CS490

This is a recapitulation course covering the primary concepts of Computer Science. Students enhance their problem solving and presentation skills as they prepare solutions to several conceptual questions covering the core course topics in the undergraduate curriculum:

* Theory (Math248, CS312, CS386, CS337)
* Programming (CS201-203, CS332)
* Algorithms (CS203, Math248, CS312, CS332)
* Systems (CS120, CS122, CS245, CS440, EE444)

These course goals contribute to the success of Student Learning Outcomes (SLO):

* SLO#1: Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.
* SLO#2: Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.
* SLO #3: Students will have a strong foundation in the design, analysis, and application of many types of algorithms.
* SLO #4: Students will have a fundamental understanding of computer systems.

At the end of CS490, students take a standardized Major Field Test ([MFT](http://www.ets.org/mft/)) conducted by the Educational Testing Service ([ETS](http://www.ets.org/)) and the results are archived on CSNS at <http://csns.calstatela.edu/assessment/mft/viewMFTScores.html>.

CS490 provides the capstone experience to recapitulate the major concepts, analyze and solve problems, understand relationships and interpret material from the field of Computer Science.

The MFT is designed to measure the critical knowledge and understanding obtained by students in Computer Science. The exam produces a normed score for each student. In addition, the results provide three departmental summaries of assessment indicators which tie closely to the Student Learning Outcomes.

**A-5 Course coverage of required areas**

The required courses for the B.S in Computer Science degree program (See Table 5.1) includes:

66 units of Computer Science core courses

24 units of Computer Science electives

24 units of Mathematics courses

12 to 15 units of Science courses

One of the three Upper Division General Education Theme Courses (GE UD Theme) is a 4-unit Natural Science course.

The program meets all the specific requirements as indicated by the ABET category content for the Computer Science as described below:

**Computer science**

*Coverage of the fundamentals of algorithms, data structures, software design, concepts of programming languages and computer organization and architecture*

CS 101 Introduction to Higher Education for Computer Science Majors (2)

CS 201 Introduction to Programming  (5)

CS 202 Introduction to Object Oriented Programming  (5)

CS 203 Programming with Data Structures (5)

CS 245 Introduction to Computer Organization, Operating Systems and Networks (3)

MATH 248 Discrete Mathematics (4)

CS 312 Data Structures and Algorithms  (4)

CS 337 Software Design (3)

CS 386 Introduction to Automata Theory  (4)

*An exposure to a variety of programming languages and systems*.

CS 120 Introduction to Web Site Development (3)

CS 122 Using Relational Databases and SQL (3)

CS 201 Introduction to Programming  (5)

CS 202 Introduction to Object Oriented Programming  (5)

CS 203 Programming with Data Structures (5)

CS 245 Introduction to Computer Organization, Operating Systems and Networks (3)

CS332C Object-Oriented Programming with C++ (2)

CS332F Functional Programming (2)

CS332L Logic Programming (2)

*Proficiency in at least one higher-level language and familiarity with at least three more*

CS 201 Introduction to Programming  (5)

CS 202 Introduction to Object Oriented Programming  (5)

CS 203 Programming with Data Structures (5)

CS 312 Data Structures and Algorithms  (4)

CS 320 Web and Internet Programming (3)

CS332C Object-Oriented Programming with C++ (2)

CS332F Functional Programming (2)

CS332L Logic Programming (2)

*Advanced course work that builds on the fundamental course work to provide depth*.

CS 320 Web and Internet Programming (3)

CS 437 Software Engineering  (5)

CS 440 Introduction to Operating Systems (4)

EE 444 Computer Architecture (4)

CS 496A  Software Design Laboratory (2)

CS 496B  Software Design Laboratory (2)

CS 496C  Software Design Laboratory (2)

CS 490 Computer Science Recapitulation (2)

CS300/400 upper division computer science electives – six courses (24)

***Science and mathematics***:

*Mathematics*

MATH 206-208 Calculus I–III  (4,4,4)

MATH 248 Discrete Mathematics (4)

MATH 255 Introduction to Matrix Theory (4)

MATH 270 Introduction to Probability and Statistics(4)

*Science*

PHYS  101 – 103 General Physics (4,4,4)

[OR] PHYS 211-213 General Physics (5,5,5)

Natural Science (GE Upper Division Theme) (4)

**A-6 Relationship between courses and Student Learning Outcomes**

Each of the core courses required in our B.S. in Computer Science is related to the program Student Learning Outcomes. Student Learning Outcome could be Introduced (I), Reinforced (R) or Emphasized (E) in the course. i.e., it is indicated whether work on the knowledge, ability, or attitude specified in the Student Learning Outcome is begun in the course, given support in the course, or stressed in the course. Each listed Student Learning Outcome is addressed by the courses that are listed in the adjacent columns as indicated in Table 5.4.

CS electives are coursework beyond what is required in the Computer Science core that allows students with specific technical interests to gain a deeper understanding of the technologies associated with those interests. Each of these elective courses (listed in Table 5.2) reinforces certain Student Learning Outcomes as included in the syllabi of the courses (see Appendix A).

The relationships between courses, Student Learning Outcomes (1..10) and ABET program outcomes criteria (a..k) are established by Table 4.10 and Table 5.4.

| **SLO #** | **I**  **Introduce** | **R**  **Reinforce** | **E**  **Emphasize** |
| --- | --- | --- | --- |
| **1** | MATH206-MATH208 | MATH248,MATH255, CS312, CS386,  CS Electives | CS490 |
| **2** | CS201-203 | CS320,CS332,  CS Electives | CS490 |
| **3** | CS201-203 | CS312  CS Electives | CS490 |
| **4** | CS120,CS122  CS245 | CS440,EE444  CS Electives | CS490 |
| **5** | CS201-203 | CS320, CS337  CS Electives | CS496A |
| **6** | CS201-203  CS320 | CS337,CS496AB  CS Electives | CS437,CS496C |
| **7** | CS201-203  CS320 | CS337,CS496AB  CS Electives | CS437,CS496C |
| **8** | TECH250 | CS496AB  CS Electives | CS496C |
| **9** | TECH250 | CS301 |  |
| **10** | CS101 | CS301 |  |

**Table 5.4 Student Learning Outcomes: Courses Map**

### Course Syllabi and Course Journals

Course syllabi of all courses used to satisfy the mathematics, science, and

Computer Science requirements are included in Appendix A.

Course Syllabi are also available online on at <http://csns.calstatela.edu/courses.html?level=undergraduate>

All information pertaining to courses is compiled as course journals and will be provided to the ABET evaluators.

Course Journals are compiled with the following information:

ABET Syllabus

Instructor Course Syllabus

Handouts

Assignments and Exams

Student Samples

All Course Journals are available online on CSNS at <http://csns.calstatela.edu/assessment/viewCourseJournals.html?level=undergraduate>.

# CRITERION 6. FACULTY

1. **Faculty Qualifications**

Eight of the nine full-time faculty members hold PhD’s in Computer Science; and one has a PhD in Mathematics; As Table 6.1 indicates the entire full-time faculty is tenured and has many years of teaching experience.

The Department has seven part-time faculty who routinely teach Computer Science classes. The part-time faculty are working professionals in the software industry.

All faculty are thus highly qualified and dedicated to maintaining outstanding undergraduate and graduate programs.

1. **Faculty Workload**

As described in Table 6.2, all full-time faculty have a three quarter teaching load with twelve weighted teaching units (WTU) per quarter. This typically involves teaching three separate courses every quarter. Internal grants (Creative Leave or Sabbatical Leave), external grants (funding from the NSF or other agencies) and course release time (offered for specific duties such as Assessment Coordinator) are the only means to reduce the teaching load. Some faculty prefer to spread the thirty six annual WTU load over four quarters.

In addition faculty members are involved in general student advising and other committee assignments to satisfy the required 3 WTU of service activities per quarter.

Faculty are also responsible for advising graduate students working on a project/thesis. Students who register for advisement generate almost ½ WTU per student (5 units per student over two quarters), which is intended to reflect faculty effort. The University has a formula to give an extra credit to faculty that amounts to ½ WTU per quarter for two quarters for a total of 1 unit per student.

The report from the University’s 2010 review of our program included the following.

*College Recommendation #4: Provide adequate resources to faculty to ensure sufficient time and workload reduction for advising and supervising MS students, especially those in the thesis option.*

Unfortunately these recommendations are not implemented as yet and faculty are taking on graduate thesis/project advisement as an uncompensated overload.

The heavy teaching load limits the amount and types of scholarly activity faculty can perform. Regardless, faculty show excellent scholarship records, with faculty members frequently attending academic conferences and workshops. This is evident in the faculty vita detailed in Appendix B.

1. **Faculty Size**

*Adequacy of the size of the faculty:*

The Department has nine full-time faculty and seven regular part-time faculty, which is adequate to teach the courses we offer and to perform other tasks related to program assessment and continuous improvement. Part-time faculty typically teach four to eight units in any given quarter. Please see Appendix D (Personnel Data for Computer Science) that lists the intended faculty growth over the next six years.

*Interactions with students:*

The Department faculty maintain high levels of student/faculty interaction through teaching, research mentoring, student advising, and student counseling. Our students have ample opportunities to interact with the full-time faculty in the classroom or laboratory because most classes are small. Students also have ample opportunities to interact with the faculty outside the classroom through faculty office hours, or email or through CSNS forums.

Faculty student interaction is quite high during the year long senior design project sequence. In addition, some faculty employ both undergraduate and/or graduate students in their research projects. These projects could be external funded or generated through faculty interests in the form of directed studies.

Feedback from students and alumni is always very positive with respect to interactions between faculty and students.

*Student advising:*

Freshman advising is handled by the ECST Advisement, Outreach, and Recruitment Center. This office provides an array of services that includes: freshman and transfer orientations, academic advisement through professional staff and peer advisors, engaging freshman in the learning communities and academic excellence workshops, and assisting students with scholarship opportunities. This office also provides additional workshops for Mathematics and Physics courses.

Student advisement beyond the first year, including upper-division transfers, is handled by the academic advisor or the Department Chair. The Department used to have a Principal Undergraduate Advisor. Because of the current budget constraints, release time for that activity is no longer available. The Department Chair currently handles those responsibilities.

Academic advising is also part of our faculty members’ responsibilities. A full-time faculty member serves as our undergraduate advising coordinator. Students generally meet their principal undergraduate advisor (or Department chair). All faculty members can be approached for any advisement issues during their set office hours. The most common means of communications between faculty and students is via email. In addition, student advising is facilitated by a CSNS forum on Advisement which is moderated by the Department Chair.

*Oversight of the program:*

Full-time faculty have collective responsibility for and oversight over all programs. They are responsible for the consistency and quality of all our required and elective courses. Full-time faculty are responsible for coordinating the instruction of sections taught by part-time faculty. The Department also encourages student organizations that provide tutoring services to ensure that they understand what faculty expect from students.

The Assessment Committee, consisting of three faculty, are responsible for the evaluation of our programs. As described under Criterion 4, this committee ensures the continuous improvement of our programs.

1. **Professional Development**

The CSULA administration, College Dean, and the Department of Computer Science provide faculty with opportunities to develop their skills and learn new ones, in both technical areas and in the area of developing innovative teaching and learning strategies and appropriate use of technology in classrooms. To promote professional development and scholarly activities, faculty are encouraged to apply for externally-funded research grants, which they may use to “buy out” their time and reduce their teaching loads. The Office of Research and Development (ORAD) provides assistance to faculty with obtaining external funding in applying for extramural funding primarily, though not exclusively, through government agencies. ORAD provides services at all stages of the proposal submission and pre-award processes.

Faculty members may also apply to the University Mini-leaves, summer research, professional travel, and general research. A mini-leave generally provides 4 units of release time. Faculty members can also apply for a sabbatical leave—although these are severely limited. Several faculty have taken advantage of Mini-leaves and sabbatical leaves.

The university increased faculty travel support from $1,000 to $1,500 per year starting in 2012. This generally provides each faculty member funds to attend one conference per year.

To foster development in innovative teaching and learning strategies, faculty are encouraged to attend regional and national engineering educational seminars. The Center for Effective Teaching and Learning (CETL) was established to help achieve this goal. This Center advances the scholarship of teaching and learning by offering a variety of services supporting faculty development to develop full teaching potential. CETL offers support to create engaging learning environments to enhance student learning outcomes. Faculty development includes one-on-one consultations, workshops, brown bag discussions, webinars, and peer mentoring.

The report from the University’s 2010 review of our program included the following.

*The nine full-time Computer Science faculty members are current in their respective fields and actively involved in research. A significant amount of this research involves students. .*

***Commendation:*** *For engaging graduate students and guiding them in meaningful thesis projects and for obtaining high praise from external reviewers and Industry Advisory Board members, both of whom noted that some of the projects were at the level of doctoral research.*

***Commendation:*** *For engaging students in research including a dozen presentations at international conferences.*

Each year, during the National Engineers' Week, the College sponsors a major industry career day. During this event, representatives from industry, government and private practice are invited to set up information tables, and to meet with students throughout the day. Typically representatives from 20 to 30 companies are present at these events. This gives the faculty an opportunity to meet perspective employers and learn their current engineering needs. Often these interactions lead to an invitation for the employer to serve on the department’s Industrial Advisory Board.

Since CSULA is primarily a teaching institution and since full-time faculty are responsible for twelve weighted teaching units each quarter, faculty find it difficult to devote a significant amount of time to professional development. Despite this handicap, our faculty have been quite productive in their professional development work as indicated in their vita detailed in Appendix B.

1. **Authority and Responsibility of Faculty**

The faculty designs and delivers the curriculum. Its responsibility is to ensure that the curriculum is kept up to date with changing technologies and the professional needs of our graduates. An annual faculty retreat is conducted during the Spring quarter. Discussions cover a broad range of issues related to Curriculum Objectives, Learning Outcomes, and Courses.

The faculty controls the development and evaluation of our programs. Findings of the Department Assessment Committee lead to Course/Program modification proposals by the faculty. Typical proposals could be to:

* modify course content and/or its delivery
* modify curriculum (e.g., adding new courses, deleting ones that have become outdated),
* revise or modify modes of instruction (e.g., adding laboratory components).

Modification proposals are reviewed by the Department’s Instructional Affairs Committee (IAC) which must approve all proposed curricular changes. Once approved by the Department IAC proposed changes are reviewed by the College Instructional Affairs Committee. Upon approval, the College Instructional Affairs Committee forwards the documents to the College Associate Dean, who then forwards them to the University Educational Policy Committee. After approval at the University level, the changes are entered into the University catalog. Results of the annual internal assessment process are compiled into an Assessment Reports and posted on CSNS.

Each full-time faculty member serves as a coordinator for a number of courses. The course coordinator is a key in assessing and suggesting any improvement in the course. The responsibilities of the course coordinators are as follows:

* Provide Course Description and a sample Course Syllabus.
* Select a textbook.
* Check the syllabi that are provided by the instructors.
* Evaluate Course Journals.

The course coordinators are listed below in Table 6.3

| Faculty Name | Highest Degree Earned- Field and Year | Rank 1 | Type of Academic Appointment2  T, TT, NTT | FT or PT | Years of Experience | | | Professional Registration/ Certification | Level of Activity  H, M, or L | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Govt./Ind. Practice | Teaching | This Institution | Professional Organizations | Professional Development | Consulting/ summer work in industry |
| Russell J. Abbott | Ph.D. – 1973  Computer Science | P | T | FT | 30 | 38 | 25 | - | M | M (now)  H (prev) | L (now)  H (prev) |
| Vladimir Akis | Ph.D.- 1982  Mathematics | P | T | FT | 0 | 34 | 30 | - | L | M | L |
| Valentino Crespi | Ph.D.- 1997  Computer Science | ASC | T | FT |  |  |  | - | H | H | L |
| Huiping Guo | Ph.D.- 2003  Computer Science | ASC | T | FT | 4 | 6 | 6 | - | L | M | M |
| Jiang Guo | Ph.D.- 1996  Computer Science | P | T | FT | 3 | 11 | 10 | MSSE | M | M | L |
| Eun-Young “Elaine” Kang | Ph.D. - 2003  Computer Science | ASC | T | FT | 1 | 9 | 7.5 | - | M | M | L |
| Raj S Pamula | Ph.D. - 1987  Computer Science | P | T | FT | 1 | 23 | 23 | - | L | L | L |
| Behzad Parviz | Ph.D. - 1986  Computer Science | P | T | FT | 10 | 25 | 24 | - | L | L | L (now)  H (prev) |
| Chengyu Sun | Ph.D.- 2004  Computer Science | ASC | T | FT | 1 | 8 | 8 | - | L | M | L |
| Albert Cervantes | MS – 2006  Electrical Engineering | A | NTT | PT | 7 | 5 | 5 | - | L | M | H |
| Edmund Gean | MS – 1991  Computer Science | A | NTT | PT | 20 | 9 | 9 | MCPE  CCNA | M | H | H |
| Mahan Hajianpour | MS – 2007  Computer Science | A | NTT | PT | 7 | 5 | 5 | - | M | H | H |
| John Hurley | MS-2012 (Exp.)  Computer Science | A | NTT | PT | 20 | 5 | 1 | - | L | L | L |
| Jose M. Macias | Ph.D – 1998  Mathematics | A | NTT | PT | 31 | 30 | 30 | - | M | H | H |
| Bapa Rao | Ph.D – 1987  Computer Science | A | NTT | PT | 20 | 12 | 6.5 | - | L | M | H |
| John Tran | MS – 1998  Computer Science | A | NTT | PT | 14 | 14 | 5 | - | H | H | H |

**Table 6-1: Faculty Qualifications**

1. Code: P = Professor ASC = Associate Professor AST = Assistant Professor I = Instructor A = Adjunct O = Other
2. Code: TT = Tenure Track T = Tenured NTT = Non Tenure Track
3. The level of activity, high, medium or low, should reflect an average over the year prior to the visit plus the two previous years at the institution

| Faculty Member (name) | PT or FT | Classes Taught (Course No./Credit Hrs.)  (Fall2011, Winter 2012, Spring2012) | Program Activity Distribution | | | % of Time Devoted  to the Program |
| --- | --- | --- | --- | --- | --- | --- |
| Teaching | Research or Scholarship | Other  (Service/  Sabbatical  leave) |
| Russell J. Abbott | FT | CS301(1); CS332C(1+1); CS332F(1+1); CS454(4); CS461(4); CS496A(2); CS496C(2); CS560(4). | 80% | 10% | 10% | 100% |
| Vladimir Akis | FT | CS312(4) (F,S); CS386(4); CS512(4); MATHxxx (20) | 80% | 10% | 10% | 50% |
| Valentino Crespi | FT | CS201(4); CS312(4); CS386(4); CS486(4); CS594(4); | 55% | 35% | 10% | 100% |
| Huiping Guo | FT | CS122(3),CS120(4),CS201(4),CS422(4),CS470(4), CS480(4), CS522(4), CS570(4), CS580(4) | 80% | 10% | 10% | 100% |
| Jiang Guo | FT | CS202(4); CS245(4); CS440(4), CS496B(2);CS537(4), CS590(4) | 80% | 10% | 10% | 100% |
| Eun-Young “Elaine” Kang | FT | CS450(5); | 14% | 20% | 66%  (Sabbatical + Maternity) | 100% |
| Raj S Pamula | FT | CS101 (2) (F,W,S); CS370(4); CS454-4; CS490(2); | 40% | 10% | 50% | 100% |
| Behzad Parviz | FT | CS120(4),CS122(4),CS202(6),CS203(6),CS540(4), MATHxxx (12); | 80% | 10% | 10% | 100% |
| Chengyu Sun | FT | CS122(3),CS320(3),CS422(4),CS520(4),CS522(4) | 80% | 10% | 10% | 100% |
| Albert Cervantes | PT | CS242(4); CS345(4); CS320(3) | 20% | - | - | 20% |
| Edmund Gean | PT | CS447(4); CS581(4); CS342(4) | 20% | - | - | 20% |
| Mahan Hajianpour | PT | CS203(5); CS454(4); CS203(5) | 20% | - | - | 20% |
| John Hurley | PT | Fall: CS 120(3), CS 201( 5); CS 120(3), CS190 (2), CS 201 (labs, 2) | 50% | - | - | 50% |
| Jose M. Macias | PT | CS337(3); 1(5),Skill Evaluation/Survey as defined earleir measure. Note that theCS290(2); CS437 (5); CS332L (2); CS488 (4) | 20% to 30% | - | - | 20% |
| Bapa Rao | PT | CS460(4), CS575(4) | 20% | - | - | 20% |
| John Tran | PT | CS201(5); 1(5),Skill Evaluation/Survey as defined earleir measure. Note that theCS202(5) | 20% | - | - | 20% |

**Table 6-2: Faculty Workload Summary**

| **Courses** | **Coordinator** | **Committee** |
| --- | --- | --- |
| Math248, Math255, CS340, CS440 | V.Akis | B.Parviz, J.Guo |
| 301, 332F, 332L, 460, 454(UI, Mobile Dev.) | R.Abbott | E.Kang, H.Guo |
| 312, 386, 486, 488 | V.Crespi | V.Akis, B.Parviz |
| 245, 337, 437 | J.Guo | R.Pamula, R.Abbott |
| 447, 470, 480 | H.Guo | R.Pamula, C. Sun |
| 332C, 242, 342, 450, 451, 461, 454(Game Progr.) | E.Kang | V.Akis, V.Crespi |
| 101, 160, 190, 290, 370 | R.Pamula | J.Guo, B.Parviz |
| 201, 202, 203 | B.Parviz | E.Kang, J. Guo, R. Pamula |
| 120, 122, 320, 422 | C.Sun | H.Guo, B.Parviz |
| 496ABC | C.Sun | J.Guo, R.Abbott |
| 490 | R.Pamula | J.Guo, R.Abbott |

**Table 6.3: Course Coordinators**

# CRITERION 7. FACILITIES

**A.** **Offices, Classrooms and Laboratories**

Offices

Most full-time faculty members have private offices; two share an office. Even though it is desirable to have a private office for all full-time faculty, the current office space allocation is better than the university’s policy of having every office accommodating two full-time faculty. All our part-time faculty share one common office.

Full-time faculty and Department staff are assigned either a desktop or a laptop computer, as chosen by them, with access to a laser printer. These are refreshed on a three year cycle (approximately) as indicated by the university baseline plan ([http://www.calstatela.edu/its/baseline/)](http://www.calstatela.edu/its/baseline/). Part-time faculty and student assistants are also provided with computer setup. Campus-authenticated wireless connectivity is also available via numerous wireless access points in the building.

Classrooms

Most Computer Science courses are taught in Computer Classrooms, each of which is equipped with a computer projector. The instructor and students all have networked computers in the computer classroom.

In some instances, Computer Science courses are taught in Smart Classrooms which are equipped with an instructor networked computer station and a computer projector.

Some Computer Classrooms or Smart Classrooms are also equipped with Mediasite technology. Mediasite is a webcasting technology that completely automates the recording, distribution, management and analytics of high-quality video and multimedia presentations. Faculty present as usual with no need to learn new technology. By capturing the courses, faculty can build instant video libraries and provide a virtual classroom for students to watch at their convenience. This process should turn out to be significant for student retention.

Tables C.1 and Table C.2 (Appendix C) describes the Computer Classrooms and Mediasite rooms available for Computer Science program instruction.

Laboratories

Laboratories are described below under Computing Resources.

**B. Computing Resources**

Computing resources for the Computer Science program includes the following:

* Computer classrooms
* Mediasite classrooms
* Server platforms
* Miscellaneous Hardware
* Software

See Appendix C for a description and maintenance of these computing resources.

These computing resources are adequate. We are also requesting the campus ITS to provide Virtual Private Network (VPN) access for all the senior students. Currently, students are restricted to certain ports on certain servers. The time has come where students need to access a number of campus resources on an ongoing basis.

* Student need to setup application servers in order to work on their senior design projects and have secure access to them through VPN software without exposing these application servers to the outside world. Without the VPN access, the department continues to request for certain ports to be open on certain servers which causes security concerns.
* Expensive software installed on servers allows the students to work from home. This would give the commuting students the necessary time to complete their projects while not having to install the software on their desktops. Software is expensive and can be better utilized by providing the VPN access. This would increase the usage of our computing resources and would give students more flexibility for completing their projects in time.

The solution is for the campus to provide VPN access to all senior students in the College of ECST.

**C. Guidance**

All beginning CS1xx/CS2xx classes are integrated with a 1-unit laboratory component where students get proper guidance in the use of relevant hardware/software from the instructor. ITCs enhance the guidance by providing Instructional Technology support to the students.

Open Access labs are supervised by Lab Technicians (some of whom are computer science majors) who provide additional help to students. In addition, students have access to faculty by email, through forums, and during their office hours for any guidance concerning the use of hardware or software.

The Department hires programming tutors through an Instructional Resources Activity grant to provide additional guidance to students. This tutorial assistance is a part of the Department of Computer Science’s efforts to enhance its teaching mission from the point of view of assisting students currently taking undergraduate programming courses (CS190, CS201, CS202, CS203, CS242, and CS290). Computer Science, Engineering and Technology students enrolled in programming courses often require additional help in a computer laboratory that is often not available from any of our open access laboratories.

**D. Maintenance and Upgrading of Facilities**

Information Technology Services (ITS) is responsible for maintenance and upgrading of all computing facilities as indicated in their mission below:

*“ITS supports the University's ever-increasing use of new and various technologies including latest software applications. The Desktop Services (DS) is dedicated to providing a quality desktop downloadable image, called DSS (Desktop Services Software), containing the operating system, productivity, and security software to support all University machines. DS is also involved with wireless and mobile infrastructure and services. Baseline Services (BS) provides new desktop hardware and systems in a cycle called Baseline Refresh. Most University desktops and some notebooks fall into this category. It is also this group's responsibility to maintain these machines*

*The Academic Information Technology Consultants (ITCs) are full time staff member who play critical roles as liaison between University faculty and staff, Information Technology Services (ITS), and campus colleges in a wide range of areas related to the use of information technology and information security. ITCs enhance teaching, learning and administrative operations by providing high quality Instructional Technology support to the students, faculty, and staff. This includes workstation hardware and software support for faculty, staff, the computer laboratories, and technology enhanced classrooms (TECs)”*

All Department computer classrooms are covered by the baseline refresh cycle. All Department server platforms are upgraded approximately on a three year cycle by the College. Software will be renewed as per the agreements (renewing annual licenses or buying new licenses on upgrades.) In some cases, proprietary software could be substituted with equivalent free software.

The ITC staff for the College does an excellent job keeping track of hardware upgrades. In addition, the College ITC’s update the software image in all the labs to include both proprietary and open source software.

See Appendix C concerning maintenance and upgrading of overall hardware/software used by the program.

**E. Library Services**

To ensure that the Library achieves its mission, each department has a librarian liaison who serves as a point of contact between the library and the Department.  Over the years, the Library has striven to maintain its commitment to the College of Engineering, Computer Science, and Technology in a variety of ways, including support through the purchase of relevant materials – books, journals, electronic resources, media materials, etc. Prof. Valentino Crespi (Department Library Coordinator) interacts with Librarian Ken Ryan (Coordinator for the acquisition, implementation, and maintenance of the Library's digital resources and the liaison to the Department of Computer Science).

Faculty can request items from the Library to support their research and teaching. The Library collection consists of books, including a large collection of current e-books relating to computer science (Safari Books Online), and a variety of online databases that include journal articles and other information.

With over 40,000 online journal subscriptions in all disciplines, the Library carries very few print journals. Hence the majority of its Computer Science journals are found in its subscription databases. The two main databases to which the Library subscribes to support the Computer Science Department are IEEE/IET Electronic Library (IEL), which contains access to the full-text of over 11,000 publications, including a number of ACM/IEEE computer-related journals, and Engineering Village 2 which contains abstracts and references to over 5,000 journals, proceedings and technical reports, with links to selected online full-text materials. Other full-text interdisciplinary databases relevant to the program include Business Source Premier and ScienceDirect.

The library does not subscribe to all ACM or IEEE computer-related journals as the cost would be prohibitive. However, the library offers a no-cost option for faculty and students to obtain journal articles electronically via e-mail – the ILLiad InterLibrary Loan Service. Given the large number of academic libraries in the Los Angeles area, turnaround times are typically less than one week from the order date.

As a part of its educational mission, the Library also provides individual and group instruction as a part of its information literacy programs. A library professional gives a presentation to incoming freshmen and transfer students in CS101–a course taken by all students within the first two quarters of admission.

Given the demand for these services, the current staffing is adequate. (See Appendix D for more information)

1. **Overall Comments on Facilities**

The Instructional, Computing, Laboratory, and Library facilities are adequate. The classrooms, laboratories and associated equipment are adequate and promote faculty-student interaction. The computing infrastructures are in place to support the instructional and scholarly activities of students and faculty.

In addition, we are hopeful of securing VPN access to students (see Section B under Criterion 7) which will further aid in accomplishing the program’s educational objectives.

# CRITERION 8. INSTITUTIONAL SUPPORT

1. **Leadership**

The Chair of the Computer Science Department provides the leadership for the program. The Chair in conjunction with two other faculty (Dr. Abbott and Dr. Sun) form the Assessment Committee to oversee the implementation of the continuous improvement plan. The Chair in conjunction with the Industry Advisory Board Chair (Dr. Kang) organizes the annual Industry Advisory Board meeting. Each full-time faculty member provides leadership to ensure that the course objectives are met by serving as a coordinator for a number of courses. The Chair organizes an annual faculty retreat to discuss feedback from the Assessment Committee and the Industry Advisory Board. We use the retreat as an opportunity to discuss a broad range of issues regarding the Objectives, Learning Outcomes, and Courses of the undergraduate and the graduate curriculum.

The Dean and the Associate Dean provide leadership at the college level. The Department Chair gets release time and teaches one course per quarter. The Dean, Associate Dean, and the five department chairs meet weekly to provide leadership at the college level. This group establishes college priorities and addresses various issues that impact all College programs.

The Computer Science Department has enjoyed a good working relationship with the Dean’s office. The Dean has been very supportive of the Computer Science Department’s needs, both in the area of resources, and in recruiting new faculty.

### B. Program Budget and Financial Support

The budgetary process:

As a state supported institution, there are two primary sources of funds for the operation of the University—and thus the Department— [an allocation from the state based on enrollment](http://www.calstate.edu/budget/fybudget/2011-2012/documentation/2-marginal-cost-of-instruction-table.shtml) and [tuition and fee revenue](http://www.calstate.edu/budget/fybudget/2012-2013/documentation/14-mandatory-fees-table.shtml) collected from students. The University allocates funding received from these two sources to each of the colleges within the university based on enrollment targets. The College then prioritizes the distribution across the departments. There are potentially a number of sources of what are called “soft funds.” One is funding from external grants secured by the faculty. At the college level funds may also be available as gifts from individuals/companies. At the department level funds may be available for Instructional Related Activities from student fee revenue allocated by the university, Senior Design support grants from companies, and a portion of the revenues from departmental offerings through Extended Education. Please see Appendix D (Section 6 - Budget Allocations and Actual Expenditures) for budget information from the previous two years.

Teaching support:

The university provides financial support for all teaching related activities. This involves all forms of instructional (both classroom and laboratories) support. Classes typically have a maximum of 30 students. Lecture and laboratory instruction is carried out by the faculty. There is no provision for graders. In addition to salary and wages for full-time faculty members and staff, adequate budget is provided for additional part-time faculty when there is sufficient student demand. Funding for new faculty positions is allocated from the central administration.

Support for infrastructure, facilities, and equipment:

The Department’s laboratory computing equipment is covered by the University’s baseline refresh cycle. All Department server platforms are upgraded on a three year cycle by the College. All software is upgraded as and when needed. Most of the software is open source and as such has limited budgetary implications.

The College from time to time solicits equipment requests from its five departments. These are prioritized at the college level based on need and demand. Hardware and software equipment funding depends on the budget allocated to the college by the university,

The University realizes that the programs within the College of Engineering, Computer Science, and Technology are “high cost” programs (a lower student faculty ratio and a higher cost for each full-time equivalent student) and has provided the resources necessary to maintain program quality.

The University has shown a commitment to the Computer Science Department by creating four computer classrooms and four server platforms. These are the primary laboratory resources that are used in our program. Adequate funds are provided for software needed by faculty, students, and the computer classrooms.

The Major Field Test (MFT) (described in Section B-1 under Criterion 4) offered by ETS is priced at $25 per test. Typically around 20 to 25 students take this test every year. The cost has been borne by the University as MFT is used to measure the critical knowledge and understanding obtained by students in Computer Science.

Adequacy of resources to attain student outcomes:

The number of faculty members in the Department is adequate to teach the required courses and also to perform other tasks related to program assessment and continuous improvement. The faculty is currently comprised of nine full-time faculty members.

The support for hardware and software resources is adequate. Please see Appendix C that lists the current equipment and equipment needs over the next five years. Please see Appendix D (Section 6 - Budget Allocations and Actual Expenditures) that describes the actual equipment expenditures for the last 5 years.

### C. Staffing

There is one full-time permanent Administrative Support Coordinator for the Department. Her primary responsibility is to deal with the daily operations of the Department. The Department also employs a half-time student assistant to provide additional staff support. The Department office suite has a total of three workstations.

The staff does an excellent job supporting the administrative functions of the Department office and faculty. Our current Department administrator is very knowledgeable with respect to all policies and interacts effectively with other staff at the college and university levels.

The Department is in dire need of a dedicated technology staff position to manage its extensive hardware and software base. This need was recognized by an external evaluator and the University Program Review Committee, which made the following recommendation: *“In order to provide adequate technical support to curricular offerings, seek to have a permanent full-time Information Technology Consultant (ITC) assigned to the Department.”* It is our desire that this recommendation is met very soon. Funding of this position will facilitate and accelerate program offerings in mobile computing, cloud computing, network security and other new and evolving technologies.

The college currently has two Information Technology Consultants (ITC) that provide services to all the five departments. ITCs are essential members of the campus-wide team of technical personnel supporting the overall mission and operations of the University.

**D. Faculty Hiring and Retention**

There was no difficulty in hiring and retaining new faculty as we grew in size from four faculty in 2000-01 to nine faculty in 2005-06. No hiring was attempted in the last five years as the nine faculty were sufficient to cover all Department offerings.

We are currently making plans to offer a new program (Bachelor of Arts with various options) that could be ABET accreditable under the Information Technology Criteria. In addition, the Computer Science programs are experiencing some growth. To meet these demands, the Department will be requesting two new faculty hires (in the areas of Computer Graphics/Gaming and Computer Networks/Security) in the coming few years. The hiring requests will be forwarded to the College which in turn will forward the requests to the university. The university establishes the procedures for searching for new faculty based upon need and available resources.

Faculty currently receive competitive salaries and benefits, including the Public Employee Retirement System (PERS), health insurance, vision care, dental plans, death benefits, accrual of sick leave, life insurance, and long-term disability. Detailed information on faculty benefits can be found in Chapter VIII of the Faculty Handbook. The faculty are state employees whose terms and conditions of employment are under the Collective Bargaining Agreement between the CSU and the California Faculty Association, which is the union representing faculty employees.

There have been no difficulties in retaining our current faculty.

### E. Support of Faculty Professional Development

All faculty are required to teach 12 weighted teaching units per quarter. Internal grants (Creative Leave or Sabbatical Leave), external grants (funding from the NSF or other agencies), and release time are the only means to reduce the teaching load. In addition, all faculty members have been involved in service activities at all three levels department/college/university.

Faculty professional development is facilitated in a number of ways:

* + The university provided funding of up to $1000 per year through 2011 for conference travel. This has been increased to $1500 per year for 2011-12. These funds may be accumulated in a three year period. Faculty travel funds are augmented by the department/college from funds derived from extended education fees or sponsored senior design project fees.
  + Faculty compete for university sponsored mini grants or creative leave awards which, if awarded, provide one course release time. Competition across the University for these awards is intense.
  + Faculty compete for sabbatical awards which, if awarded, provide for one quarter release time. Competition across the University for these awards is intense.

Several faculty in the Department have been awarded mini-leaves and sabbatical leaves. Even with the heavy teaching/service responsibilities, all faculty members have kept up their scholarly activities. (See Appendix B)

# PROGRAM CRITERIA

The Computer Science program satisfies the applicable program criteria in the areas of curriculum and faculty as described below.

**Curriculum**

As described in Section B under Criterion 4, each of the Student Learning Outcomes, listed as “1..10”, is related to one or more ABET Outcomes Criteria, listed as “a..k”. By demonstrating the achievement of Student Learning Outcomes, the undergraduate B.S degree program in Computer Science demonstrates its compliance with the specific ABET program outcomes criteria.

The program meets all the specific requirements for Computer Science, Mathematics and Science in terms of hours and depth of study as described under Criterion 5 – Section A and summarized below:

**Computer science** (90 units)

*Coverage of the fundamentals of algorithms, data structures, software design, concepts of programming languages and computer organization and architecture* (35 units)

*An exposure to a variety of programming languages and systems*. (30 units)

*Proficiency in at least one higher-level language and familiarity with at least three more* (28 units)

*Advanced course work that builds on the fundamental course work to provide depth*. (48 units)

***Mathematics*** (24 units)

*Calculus , Discrete Mathematics, Matrix Theory, Probability and Statistics*

***Science*** (16 – 19 units)

General Physics, Natural Science

**Faculty**

As described under Criterion 6, all faculty are qualified and their responsibility includes the oversight of the degree programs. In addition, faculty maintain currency in their specialty areas.

**APPENDICES**

# Appendix A – Course Syllabi

This section contains Course Syllabi for all courses used in the undergraduate program and presented in the order indicated in Table A.1.

[Course Abbreviations: Computer Science (CS), Electrical Engineering (EE), Technology (TECH), Mathematics (MATH), Physics (PHYS), University (UNIV)].

| Course | Name |
| --- | --- |
| CS101 | Introduction to Higher Education for Computer Science Majors |
| CS120 | [Introduction to Web Site Development](http://csns.calstatela.edu/download.html?fileId=2066188) |
| CS122 | [Using Relational Databases and SQL](http://csns.calstatela.edu/download.html?fileId=2065290) |
| CS201 | [Introduction to Programming](http://csns.calstatela.edu/download.html?fileId=2066191) |
| CS202 | [Introduction to Object Oriented Programming](http://csns.calstatela.edu/download.html?fileId=2066192) |
| CS203 | [Programming with Data Structures](http://csns.calstatela.edu/download.html?fileId=2066193) |
| CS245 | [Introduction to Computer Organization, Operating Systems, and Networks](http://csns.calstatela.edu/download.html?fileId=2066195) |
| CS301 | [Computer Ethics in the Information Age](http://csns.calstatela.edu/download.html?fileId=2066197) |
| CS312 | [Data Structures and Algorithms](http://csns.calstatela.edu/download.html?fileId=2066198) |
| CS320 | [Web and Internet Programming](http://csns.calstatela.edu/download.html?fileId=2066199) |
| CS332C | [Advanced C++ Programming](http://csns.calstatela.edu/download.html?fileId=4314941) |
| CS332F | [Functional Programming](http://csns.calstatela.edu/download.html?fileId=2066200) |
| CS332L | [Logic Programming](http://csns.calstatela.edu/download.html?fileId=2066201) |
| CS337 | [Software Design](http://csns.calstatela.edu/download.html?fileId=4281529) |
| CS340 | [Assembly Language and Systems Programming](http://csns.calstatela.edu/download.html?fileId=2066202) |
| CS345 | [UNIX and Shell Programming](http://csns.calstatela.edu/download.html?fileId=2066204) |
| CS370 | [Parallel and Distributed Programming](http://csns.calstatela.edu/download.html?fileId=2066206) |
| CS386 | [Introduction to Automata Theory](http://csns.calstatela.edu/download.html?fileId=2066207) |
| CS422 | [Principles of Database Systems](http://csns.calstatela.edu/download.html?fileId=2066209) |
| CS437 | [Software Engineering](http://csns.calstatela.edu/download.html?fileId=2066210) |
| CS440 | [Introduction to Operating Systems](http://csns.calstatela.edu/download.html?fileId=2066211) |
| CS447 | [Computer Network Configuration and Management](http://csns.calstatela.edu/download.html?fileId=2066212) |
| CS450 | [Computer Graphics](http://csns.calstatela.edu/download.html?fileId=2066213) |
| CS451 | [Multimedia Software Systems](http://csns.calstatela.edu/download.html?fileId=2066214) |
| CS454 | Topics in Advanced Computer Science |
| CS460 | [Artificial Intelligence](http://csns.calstatela.edu/download.html?fileId=2066215) |
| CS461 | [Machine Learning](http://csns.calstatela.edu/download.html?fileId=2066216) |
| CS470 | [Computer Networking Protocols](http://csns.calstatela.edu/download.html?fileId=4278406) |
| CS480 | [Cryptography and Information Security](http://csns.calstatela.edu/download.html?fileId=4278437) |
| CS486 | [Computability and Intractability](http://csns.calstatela.edu/download.html?fileId=2066217) |
| CS488 | [Compilers](http://csns.calstatela.edu/download.html?fileId=2066218) |
| CS490 | [Computer Science Recapitulation](http://csns.calstatela.edu/download.html?fileId=2066219) |
| CS496A | [Software Design Laboratory](http://csns.calstatela.edu/download.html?fileId=4290810) |
| CS496B | [Software Design Laboratory](http://csns.calstatela.edu/download.html?fileId=4316852) |
| CS496C | [Software Design Laboratory](http://csns.calstatela.edu/download.html?fileId=4316854) |
| EE444 | [Computer Architecture](http://csns.calstatela.edu/download.html?fileId=2066221) |
| MATH206 | Calculus I |
| MATH207 | Calculus II |
| MATH208 | Calculus III |
| MATH248 | Discrete Mathematics |
| MATH255 | Introduction to Matrix Theory |
| MATH270 | Introduction to Probability and Statistics |
| PHYS I | PHYS 101/PHYS 211 |
| PHYS II | PHYS 102/PHYS 212 |
| PHYS III | PHYS 103/PHYS 213 |
| TECH250 | [Impact of Technology on Individuals and Society](http://csns.calstatela.edu/download.html?fileId=4313578) |
| UNIV400 | Writing Proficiency Examination |
| UNIV401 | Writing Proficiency |

**Table A.1: Courses Table**

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
| **Course Number** | CS101 | |
| **Course Name** | Introduction to Higher Education for Computer Science Majors | |
| **Credits** | 2 units | |
| **Contact Hours** | 2 hours/week | |
| **Coordinator** | Raj Pamula | |
| **Text book** | No textbook is required | |
|  | Other supplemental resources:   * [www.calstatela.edu](http://www.calstatela.edu/) * [Undergraduate Student Handbook](http://www.calstatela.edu/academic/ecst/cs/Student_Handbook.htm) * [www.calstatela.edu/cs](http://www.calstatela.edu/cs) | |
| **Course Information** | 1. Catalog Description: Exploration of skills and resources that will help students to obtain a baccalaureate degree in the Department of Computer Science. This course must be taken once during the first two quarters at Cal. State LA. 2. Prerequisites or co-requisites: None 3. Required/Elective: This course is required in the BS program. | |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #7. Students will be able to communicate effectively orally and in writing.  SLO #8. Students will have the knowledge, skills, and attitudes for lifelong self-development.  SLO #9. Students will have the ability to analyze the local and global impact of computing on individuals and society.  SLO #10. Students will have a fundamental understanding of social, professional, ethical, legal, and security issues in computing. | |
| Other outcomes of instruction:   1. Students will get an overview of Computing disciplines 2. Students will get an overview of hardware and software at an introductory level 3. Students will get an overview of the Computer Science requirements for the B.S. degree 4. Students will get an overview of the resources available to students | |
| **Brief list of topics to be covered** | |  |  | | --- | --- | | 1)      Introduction | | | * CSULA |  | | * University rules and regulations | | | * General Education/University requirements | | | 2)    What is Computer Science?   * Requirements for BS * Blended BS/MS |  | | 3)    Hardware/Software |  | | 4)    Languages and Compilers, Platforms | | | 5)     Campus Network | | | 6)    Software Engineering/CS Project |  | | 7)    ACM/CSNS |  | | 8)    Computer Ethics   * http://www.acm.org/about/se-code/ |  | |  |  | | |
| **Laboratory Projects** | Students will make a detailed presentation on the following topics:   * Quarterly planner to complete the requirements for the BS degree. * A chosen Senior Design Project * One of the eight principles described in the   “Software Engineering Code of Ethics and Professional Practice” | |
| **Academic**  **Integrity** | | | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. | |
| **ADA Statement** | | | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. | |

**ABET Course Syllabus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Title** | Introduction to Website Development | | |
| **Course Number** | CS 120 | **Coordinator** | Chengyu Sun |
| **Total Credit** | 3 | **Contact Hours** | 5 hours/week |

**Textbook**

* Anne Boehm. *Murach's HTML, XHTML, and CSS*, Mike Murach & Associates, 2010.
* Terry Felke-Morris. *Web Development and Design Foundations with XHTML*, Addison Wesley, 2008-2010.

**Course Information**

This course is required in the BS program.

a) Catalog Description: Development of client-side web pages using hypertext markup language ([d][x]html), Cascading Style Sheets (CSS). Javascript, and computer animation software. Lecture 2 hours, laboratory 3 hours. Graded ABC/NC.

b) Prerequisites or Co-requisites: Computer Literacy.

**Course Goals**

At the end of the course, students have a working knowledge of HTML, including the common tags and their attributes and how they are interpreted by web browsers. Students will be able to

* Create web sites with static HTML content.
* Use Cascading Style Sheets (CSS) to control the look and feel of a web site.
* Use JavaScript to add interactivity to web pages.
* Use Flash to create simple animation.
* Understand the syntax of XML and XHTML.

These course goals contribute to the success of **Student Learning Outcomes (SLO) 4**: *Students will have a fundamental understanding of computer systems.*

**Major Topics Covered in the Course:**

* Web servers and web sites
* HTML basics
* Hyperlinks and images
* Tables and frames
* CSS
* Forms
* Basic JavaScript
* Advanced JavaScript
* DHTML, XHTML, and XML
* Flash animation

**Laboratory Projects**

Each week the students complete a 1.5-hour lab project on a selected topic:

* Week 1: Design and setup a web site with 2 to 3 simple HTML pages, and one of the pages is an index page.
* Week 2: Add hyperlinks, images, and styling (fonts and colors) to the pages.
* Week 3: Use tables and frames to control page layout.
* Week 4: Use CSS to control the look and feel of the web site.
* Week 5: More exercises with CSS.
* Week 6: Create forms to take user input.
* Week 7: Use JavaScript to process user input.
* Week 8: More exercises on JavaScript.
* Week 9: Use Flash to create menus and navigation controls.
* Week 10: Create simple Flash animation.

**ABET Course Syllabus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Title** | Using Relational Databases and SQL | | |
| **Course Number** | CS 122 | **Coordinator** | Chengyu Sun |
| **Total Credit** | 3 | **Contact Hours** | 5 hours/week |

**Textbook**

* Gary Randolph and Jeffrey Griffin. *SQL Essentials*, Franklin Beedle and Associates, 2004.
* Alan Beaulieu. *Learning SQL*, O'Reilly Media, 2009.

**Course Information**

This course is required in the BS program.

a) Catalog Description: An introduction to relational databases and the SQL query language. Database modeling as collection of objects and their relationships; Entity relationship model. SQL as a query language. Grouping and other advanced queries. Lecture 2 hours, laboratory 3 hours. Graded ABC/ NC.

b) Prerequisites or Co-requisites: Computer Literacy.

**Course Goals**

At the end of the course, students are able to

* Set up and use at least one mainstream database management system.
* Use the SQL query language to express compound search conditions, combine and process data from multiple columns or tables, and format the results into user-friendly reports.
* Design and implement a database schema in 3rd Normal Form and improve an existing database schema by normalization.

These course goals contribute to the success of **Student Learning Outcomes (SLO) 4**: *Students will have a fundamental understanding of computer systems.*

**Major Topics Covered in the Course:**

* Introduction to relational database systems and SQL
* Selections
* Joins
* Aggregations
* Functions and set operations
* Subqueries
* Views and temporary tables
* Table creation and updates
* Introduction to database design
* Normalization
* SQL query performance issues

**Laboratory Projects**

Each week the students complete a 1.5-hour lab project on a selected topic:

* Week 1: Familiarize with the schema of given database, and retrieve information using simple selection queries.
* Week 2: Construct selection queries with more complex predicates using various operators.
* Week 3: Combine information from two tables using join queries.
* Week 4: Design more complex join queries such as self joins or joins that involve three or more tables.
* Week 5: Practice with aggregation queries, and the combination of aggregations and joins.
* Week 6: More on aggregation queries, as well as string and date functions.
* Week 7: Retrieve data using subqueries and set operators.
* Week 8: Create relations with given schema, and populate the relations using insert and update statements.
* Week 9: Given a database schema in 1st Normal Form, normalize it to 3rd Normal Form and implement it in a database management system.
* Week 10: Implement the same query using different techniques and study the performance implications of the different implementations.

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
| **Course number** | CS 201 | |
| **Course name** | Introduction to Programming | |
| **Credits** | 5 units | |
| **Contact hours** | 7 hours/week | |
| **Coordinator** | Behzad Parviz | |
| **Text book** | Introduction to JAVA Programming, by: Daniel Liang, 8th Edition  JAVA How to program, by Deitel and Deitel, 8th Edition | |
| **Course Information** | 1. Algorithm development for structured programming and computer programming; designing, coding, debugging, and documenting programs. Laboratory activities on problem analysis and software development 2. Prerequisites: Math 103: College Algebra and Trigonometry. 3. This course is required in the BS program | |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*  *SLO #7. Students will be able to communicate effectively orally and in writing.*  Other outcomes of instruction: At the end of the course students are able to:   |  | | --- | | * Divide a problem into its logical set of components | | * Have a good understanding of the basic programming concepts | | * Create simple classes with a few methods | | * Have a good understanding of how a good program design reduces coding and debugging time * Design and code mid-level problems | |  | | |
| **Brief list of topics to be covered** | |  | | --- | |  Introduction to Computers, Programs, and Java | | Elementary Programming | |       Control Statements | | Algorithms | | Pseudo code | | if…else Selection Statement | | while Repetition Statement | | Formulating Algorithms | | Compound Assignment Operators | | Primitive Types | | for Repetition Statement | | do…while Repetition Statement | | switch Multiple-Selection Statement | | break and *continue* Statements | | Logical Operators | |       Methods | | Program Modules in Java | | static Methods, *static* Fields | | Declaring and using Methods with Multiple Parameters | | Argument Promotion and Casting | | Java API Packages | | Scope f Declarations | | Method Overloading | |       Arrays | | Declaring and Creating Arrays | | Examples Using Arrays | | Passing Arrays to Methods | | Multidimensional Arrays | | Variable-Length Argument Lists | | |
| **Academic** Cheating will not be tolerated. Cheating on any assignment or exam will be **Integrity** taken seriously. All parties involved will receive a grade of F for the course and   are reported to the proper authorities.  **ADA Statement** Reasonable accommodation will be provided to any student who is registered   with the Office of Students with Disabilities and requests needed   accommodation. | |
|  | |

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
| **Course number** | CS 202 | |
| **Course name** | Introduction to Object Oriented Programming | |
| **Credits** | 5 units | |
| **Contact hours** | 7 hours/week | |
| **Coordinator** | Behzad Parviz | |
| **Text book** | Introduction to JAVA Programming, by: Daniel Liang, 8th Edition  JAVA How to program, by Deitel and Deitel, 8th Edition | |
| **Course Information** | 1. Algorithm development for Object Oriented Programming; designing, coding, and documenting programs. Laboratory activities on problem analysis and software development. 2. Prerequisites: CS201: Introduction to Programming 3. Recommended Prerequisite: Math 206: Calculus I 4. This course is required in the BS program | |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #2: Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3: Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*  *SLO #7. Students will be able to communicate effectively orally and in writing.*  Other outcomes of instruction: At the end of the course students are able to:   |  | | --- | | * Divide a problem into its logical set of components | | * Have a good understanding of the object oriented programming concepts | | * Create multiple classes to represent objects in the program definition. | | * Have a good understanding of inheritance and polymorphism. * Design and code high-level GUI programs. | |  | | |
| **Brief list of topics to be covered** | |  | | --- | |      Multidimensional Arrays (review)   * Objects and Classes * Defining Classes for Objects * Constructing Objects Using Constructors * Using classes from Java Library * Visibility Modifiers * Passing Objects to Methods * Array of Objects * Strings and Text I/O | |      Thinking in Objects | |       Inheritance and Polymorphism | |      Exception Handling | |      Graphics | |      Event-Driven Programming | |      Creating Graphical User Interfaces | | |
| **Laboratory Projects** | Each week students will complete a 3-hour lab projects on selected topics, except the exam weeks. Out of class projects: There are 3 or 4 large individual projects students are developing during the quarter. For these projects students are required to produce a design document (pseudocode, UML, etc.) and a user's manual. | |
| **Academic** Cheating will not be tolerated. Cheating on any assignment or exam will be **Integrity** taken seriously. All parties involved will receive a grade of F for the course and   are reported to the proper authorities.  **ADA Statement** Reasonable accommodation will be provided to any student who is registered   with the Office of Students with Disabilities and requests needed   accommodation. | |
|  | |

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
| **Course number** | CS 203 | |
| **Course name** | Programming with Data Structures | |
| **Credits** | 5 units | |
| **Contact hours** | 7 hours/week | |
| **Coordinator** | Behzad Parviz | |
| **Text book** | Introduction to JAVA Programming, by: Daniel Liang, 8th Edition  JAVA How to program, by Deitel and Deitel, 8th Edition | |
| **Course Information** | 1. Advanced programming techniques; elementary data structures such as dynamic arrays, linked lists, stacks, queues, and trees, sorting and searching algorithms. Laboratory activities on problem analysis and software development. 2. Prerequisites: CS202 3. Recommended Prerequisite: Math 207, Math 248 4. This course is required in the BS program | |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #2: Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3: Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*  *SLO #7. Students will be able to communicate effectively orally and in writing.*   |  |  |  | | --- | --- | --- | | Other outcomes of instruction: At the end of the course students are able to:   |  | | --- | | * Use recursion as a tool to solve some specific problems. | | * Know the standard Abstract Data Types, and their implementations. * Study and use different available JAVA Data Structures. * Know the standard searching and sorting algorithms and their efficiency. * Understand the complexity analysis for some simple software. | | | |
| **Brief list of topics to be covered** | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  | | --- | | * Applets (brief) | | * Recursion | | * Generics | | * Java Collections Framework | | * Algorithm Efficiency (brief) | | * Sorting | | | |  | | --- | | * Lists, Stacks, Queues, and Priority Queues | | |  | | --- | | * Binary Search Trees | | | * Graphs and Applications (Optional) | | | |  | | --- | | * Multithreading | | | |
| **Laboratory Projects** | Each week students will complete a 3-hour lab projects on selected topics, except the exam weeks. Out of class projects: There are 3 or 4 large individual projects students are developing during the quarter. For these projects students are required to produce a design document (pseudocode, UML, etc.) and a user's manual. | |
| **Academic** Cheating will not be tolerated. Cheating on any assignment or exam will be **Integrity** taken seriously. All parties involved will receive a grade of F for the course and   are reported to the proper authorities.  **ADA Statement** Reasonable accommodation will be provided to any student who is registered   with the Office of Students with Disabilities and requests needed   accommodation. | |
|  | |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 245 |
| **Course name** | Introduction to Computer Organization, Operating Systems and Networks |
| **Credits** | 3 units |
| **Contact hours** | 5 hours/week |
| **Coordinator** | Jiang Guo |
| **Text book** | Mark Sobell. A Practice Guide to the Unix System, Third Edition, Addison-Wesley  [Andrew S. Tanenbaum](http://www.amazon.com/Andrew-S.-Tanenbaum/e/B000AQ1UBW/ref=ntt_athr_dp_pel_1). Structured Computer Organization, 5th Edition, Prentice Hall |
| **Course Information** | 1. Catalog Description: Essential information about computer organization, operating systems and computer networks for programmers. Topics include: computer organization, data representation, the Windows/UNIX/LINUX operating system; and computer networks. Lecture 2 hours, laboratory 3 hours. Graded ABC/NC 2. Prerequisites or co-requisites: CS 202 3. Required/Elective: This course is required in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #4. Students will have a fundamental understanding of computer systems.  Other outcomes of instruction:  At the end of the course, students are able to  · Familiar with computer organization  ·     Familiar with Data Representation and Instructions of Computer  · Familiar with CPU and Memory and Input/Output  ·     Setup and use Windows Server  ·     Manage a Linux machine  ·     Use basic UNIX command  ·     Manipulate UNIX processes  ·     Edit UNIX files, such as vi or pico.  ·     Use UNIX shell  ·     Use UNIX networking  ·     Setup Windows Domain  ·     Manage Windows Domain user account, web server. |
| **Brief list of topics to be covered** | 1. Introduction to Computer Organization and Assembly Language 2. Data Representation, Von Neumann Model and Instructions of Computer 3. Binary Arithmetic and Data Type 4. Logic Operations and Boolean Algebra 5. Hardware Basic: Logic Gates 6. CPU and Memory and Input/Output 7. UNIX Operating System, UNIX Shell, UNIX File Structure 8. UNIX Networking 9. Unix Programming Tools 10. Windows Server and Domain, Building Windows Server TCP/IP Infrastructure 11. Using Active Directory 12. Web Server Configuration |
| **Laboratory Projects** | |  | | --- | | 1. Install Windows | | 1. Install Apache | | 1. Practice the UNIX command | | 1. Use UNIX networking | | 1. Use UNIX programming tools | | 1. Install Linux | | 1. Setup the Domain and TCP/IP Infrastructure | | 1. Managing User Account | | 1. Setup the Web Based Programming Environment | |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 301 |
| **Course name** | Computer Ethics in the Information Age |
| **Credits** | 1 unit (Lab) |
| **Contact hours** | 3 hours/week |
| **Coordinator** | Russ Abbott |
| **Text book** | No text. See [course wiki page](http://cs.calstatela.edu/wiki/index.php/Courses/CS_301). |
| **Course Information** | *Catalog Description:*  Responsibilities of computer scientists as influenced by growth in computer use and networks. Professional and Ethical Responsibilities; Intellectual Property; Piracy, Hacking, Viruses, Liability, Privacy, Crime, and Civil liberties. Graded ABC/ NC.  *Prerequisites or co-requisites: CS 203*  *Required/Elective: This course is required in the BS program.* |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #9. Students will have the ability to analyze the local and global impact of computing on individuals and society.*  *SLO #10. Students will have a fundamental understanding of social, professional, ethical, legal, and security issues in computing.*  *Other outcomes of instruction:* (from syllabus)  **Oral and Written Communications:**  Students will present ethical issues in oral form. They will also write analyses of ethical issues.  **Social and Ethical Issues:**  Computer Ethics is the subject matter of the course.  **Theoretical Content:**  Students are required to learn abstract concepts in computer-related ethics.  **Problem Analysis:**  Students are required to analyze situations that are likely to occur in the workplace for their ethical content.  **Solution Design:**  Students are required to determine how to make specific decisions in situations that involve ethical conflicts |
| **Brief list of topics to be covered** | From course syllabus   * What is Computer Ethics * Professional Ethics * Software Theft * Property Rights in Computer Software * Computers and Privacy * The Invasion of Privacy * Crime, Abuse, and Hacker Ethics * Computer Crime * Hacking and Viruses * Responsibility and Liability * The Social Implications of Computers: Autonomy and Access * ACM Code of Ethics and Professional Conduct * IEEE Code of Ethics |
| **Laboratory Projects** | *In class projects:* student presentations  *Out of class projects:* none |

|  |  |
| --- | --- |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 312 |
| **Course name** | Data Structures and Algorithms |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Valentino Crespi |
| **Text book** | Richard Johnsonbaugh, Marcus Schaefer. *Algorithms.* Prentice Hall, 2003.  Other Texts:   * Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Introduction to Algorithms (3rd edition).* MIT Press, 2009. * Sanjoy Dasgupta, Christos Papadimidriou, Umesh Vazirani. *Algorithms*. Mc Graw Hill, 2006. * Michael T. Goodrich, Roberto Tamassia. *Data Structures and Algorithms in Java (5th edition).* John Wiley & Sons, Inc, 2010. * Jon Kleinberg, Eva Tardos. *Algorithmic Design*. Addison-Wesley, 2005. * Robert L. Kruse.*Data structures and program design (3rd edition).* Prentice Hall, 1994*.* * Mark A. Weiss**.** *Data Structures and Algorithm Analysis in Java (2nd edition).* Addison-Wesley, 2006. * [Alfred V. Aho](http://www.amazon.com/exec/obidos/search-handle-url/index=books&field-author-exact=Alfred%20V.%20Aho&rank=-relevance%2C%2Bavailability%2C-daterank/102-5510037-5765706), [John E. Hopcroft](http://www.amazon.com/exec/obidos/search-handle-url/index=books&field-author-exact=John%20E.%20Hopcroft&rank=-relevance%2C%2Bavailability%2C-daterank/102-5510037-5765706), [Jeffrey D. Ullman](http://www.amazon.com/exec/obidos/search-handle-url/index=books&field-author-exact=Jeffrey%20D.%20Ullman&rank=-relevance%2C%2Bavailability%2C-daterank/102-5510037-5765706). *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974. |
| **Course Information** | 1. Catalog Description: Abstract data types and their use in constructing algorithms for manipulating lists, trees, and graphs; analysis of algorithms for searching, sorting, and data structure manipulation. 2. Prerequisites: Math 208, Math 248, CS203. 3. This course is required in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #1*. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  SLO #3*. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  Other outcomes of instruction:   * Analyze the correctness and computational complexity of computer algorithms. * Design (specify and implement) efficient advanced Data Structures. * Know advanced design techniques and their nontrivial application to classic problems of searching, sorting, graph optimization and combinatorial optimization. |
| **Brief list of topics to be covered** | * Mathematical Foundations (Summation Formulas, Logarithms, Induction, Lower and Upper bounds, Asymptotic Notation, Recurrence Relations, Master Theorem, Loop Invariants). * Analysis of the Correctness and of the Computational Complexity of Computer Algorithms. * Advanced Data Structures (Binary Search Trees, Balanced Trees, Heaps, Indirect Heaps, Priority Queues, Dictionaries, Hash Tables, Union-Find). * Graph Algorithms and Searching and Sorting Algorithms. * Design Techniques (Divide and Conquer, Greedy and Dynamic Programming). |
| **Laboratory Projects** | This class does not possess a laboratory module. Class projects are at the discretion of the instructor. Projects range from weekly assignments to a couple of class projects over the course of the term. |

|  |  |
| --- | --- |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

**ABET Course Syllabus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Title** | Web and Internet Programming | | |
| **Course Number** | CS 320 | **Coordinator** | Chengyu Sun |
| **Total Credit** | 3 | **Contact Hours** | 5 hours/week |

**Textbook**

Marty Hall and Larry Brown. *Core Servlets and JavaServer Pages, Vol. 1: Core Technologies, 2nd Edition*, Prentice Hall, 2003.

**Course Information**

This course is required in the BS program.

a) Catalog Description: Server-side internet programming. Development of full-fledged Internet enterprise services and applications. Laboratory activities on application development. Lecture 2 hours, laboratory 3 hours.

b) Prerequisites or Co-requisites: CS120, CS 122, CS 203.

**Course Goals**

At the end of the course, students are able to

* Understand the paradigm shift from client-side programming to server-side programming.
* Grasp the basic elements of web programming such as HTTP request/response, common HTML tags, cookies and session tracking.
* Work with one mainstream web development technology such as J2EE, PHP, or ASP.NET.
* Design and develop a complete database-driven, multi-tiered, interactive web application, and deploy and test such an application with an application server.

These course goals contribute to the success of **Student Learning Outcomes (SLO) 2, 5, 6, and 7**:

* SLO2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.
* SLO5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.
* SL06. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.
* SLO7. Students will be able to communicate effectively orally and in writing.

**Major Topics Covered in the Course:**

* Application server setup and configuration
* HTTP requests and response
* Cookies and session tracking
* Servlet programming
* JSP directives and scripting elements
* JavaBeans
* JSP Expression Language (EL)
* JSP Standard Tag Library (JSTL)
* Databases and JDBC
* Custom tag libraries
* Authentication, Authorization, and SSL
* Introduction to MVC architecture and other web development technologies

**Laboratory Projects**

Each week the students will complete an in-class lab of 2 hours and a half on selected topics:

* Week 1: Familiarize with the server setup, application deployment, and the development environment.
* Week 2: Develop a web application using servlet and HTML .
* Week 3: Add to the web application session tracking capabilities and support for multiple users.
* Week 4 and 5: Convert web application from servlet implementation to JSP, and address issues regarding the interactions among multiple JSP pages, and the separation of presentation and processing.
* Week 6: Lab on JavaBeans, Expression Language (EL), and JSP Standard Tag Library (JSTL).
* Week 7-8: Add database support and additional features to the web application.
* Week 9: Lab on creating custom tag libraries.
* Week 10: Add input validation and error handling to the web application; polish, package, and deploy the web application.

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
| **Course number** | CS 332C |
| **Course name** | C++ Object-Oriented Programming |
| **Credits** | 2 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Eun-Young Elaine Kang |
| **Text book** | Savitch, Absolute C++, 4th ed., Addison Wesley, 2010 |
| **Course Information** | a) Catalog Description: Breadth and depth of C++ programming concepts; control statements, functions, pointers, classes, application development using the object-oriented paradigm. Advanced topics such as I/0, templates, exception handling, Standard Template Library (STL)  b) Prerequisites or co-requisites: CS203  c) Required/Elective: This course is required in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively*.  Other outcomes of instruction: At the end of the course,   1. Students are able to distinguish and explain the principles of Object Oriented Paradigm. 2. Students gain experiences in developing practical applications using C++. |
| **Brief list of topics to be covered** | 1. Introduction to C++ Programming 2. Control 3. Functions 4. Vector and String 5. Arrays 6. Pointers 7. Classes and Data Abstraction 8. Operator Overloading 9. Inheritance and Polymorphism 10. Templates 11. Stream Input/Output 12. Exception Handling 13. Standard Template Library (STL) |
| **Laboratory Projects** | 1. In class projects: Each week students complete a lab assignment on selected topics from the covered materials. Each lab is designed to practice and reinforce particular concepts and be simple enough to be completed within the lab hours.  2. Out of class projects: Students are required to complete and submit three to five programming projects during the course. Each project is designed to be completed in one or two weeks. Project topics deal with real-world problems that require students to apply fundamental and advanced C++ features to solve practical computing problems. One project example can be writing a product management system with various search capabilities. |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 332F |
| **Course name** | Functional Programming |
| **Credits** | 1+1 unit |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Russ Abbott |
| **Text book** | Lipovača, M. *Learn You a Haskell for Great Good!* Available [online](http://learnyouahaskell.com/). |
| **Course Information** | * + - 1. *Catalog Description:*  Programming in functional nonprocedural programming language such as Haskell. Programming in a language (a) in which functions are values, (b) without assignments, (c) with very strong typing, (d) with lazy evaluation, and with other features common to the functional programming paradigm.       2. *Prerequisites or co-requisites: CS 203*       3. *Required/Elective:* This course is required in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *Other outcomes of instruction:* Learn the concepts of functional programming and become familiar with a functional programming language. |
| **Brief list of topics to be covered** | From course syllabus   * Types * Data structures * Recursion * Polymorphism * Folding * Functions as values * Lazy evaluation |
| **Laboratory Projects** | *In class projects:* Lab assignments each week  *Out of class projects:* Homework each week |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 332L |
| **Course name** | Logic Programming |
| **Credits** | 1+1 unit |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Russ Abbott |
| **Text book** | None. Use [online tutorials](http://cs.calstatela.edu/wiki/index.php/Courses/CS_332L#Prolog_Tutorials_listed_alphabetically_by_author). |
| **Course Information** | 1. *Catalog Description:*   Programming in a non-procedural logic programming language such as Prolog. Programming in a language that supports unification and backtracking and in which the execution of a program is the search for values that satisfy a declarative specification. 2. *Prerequisites or co-requisites:* CS 203 3. *Required/Elective:* This course is required in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *Other outcomes of instruction:* Learn the concepts of logical programming and become familiar with a logical programming language. |
| **Brief list of topics to be covered** | From course syllabus   * Declarative semantics * Operational semantics   + The Prolog programming language   + Backtracking   + Unification   + Data structures   + Meta predicates * Expert systems (optional) |
| **Laboratory Projects** | *In class projects:* Lab assignments each week  *Out of class projects:* Homework each week |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 337 |
| **Course name** | Software Design |
| **Credits** | 3 units |
| **Contact hours** | 5 hours/week |
| **Coordinator** | Jiang Guo |
| **Text book** | Software Engineering , Ian Sommerville Addison Wesley; 9 edition (March 13, 2010) ISBN: 978-0137035151 |
| **Course Information** | 1. Catalog Description: Methodologies and tools for requirements analysis and design of large complex software system; Process models, project planning, tracking, documentation, communication, and quality assurance; group laboratory project; oral and written presentations. Lecture 2 hours, laboratory 3 hours. 2. Prerequisites or co-requisites: CS 203 3. Required/Elective: This course is required in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*  *SLO #7. Students will be able to communicate effectively orally and in writing.*  Other outcomes of instruction:  At the end of the course, students are able to   |  |  | | --- | --- | |  | | |  | * *Estimate the cost and effort for software projects* * *Make schedules for software projects* | |  | * *Elicit the software requirement* | |  | * *Create data model, flow-oriented model and behavior model* | |  | * *Convert the requirement models into software architectures* | |  | * *Implement component-level design* | |
| **Brief list of topics to be covered** | 1. Estimation for Software Project 2. Software Project Scheduling 3. Software Process 4. Requirement Engineering 5. Analysis Modeling 6. Design Engineering 7. Architecture Design Web Server Configuration |
| **Laboratory Projects** | The students will be divided into small groups, 4 students in each group, to complete a big project based on a selected topic from their survey or instructor.  1. Estimate the effort of a selected project  2. Build data model, flow-oriented model and behavior model for the selected project  3. Design software architecture based on the requirement  4. Create the component-level design  5. Implement system components  6. Integrate system components  7. System Demo and Presentation  “Project Design Document” will be evaluated using the rubric: <http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/Problem_and_Requirement_Analysis/> |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
|  |  | |
| **Course Number** | CS 340 | |
|  |  | |
| **Course name** | Assembly Language and Systems Programming | |
| **Credits** | 4 units | |
| **Contact hours** | 4 hours/week | |
| **Coordinator** | Vladimir Akis | |
| **Text book** | "Assembly Language for x86 Processors" by Kip R. Irvine, 6th Edition  Additional References:  “Assembly Language Programming for the Intel Family” by W. B. Giles.  “Assembly Language for the IBM PC Family” by W. B. Jones. | |
| **Course Information** | 1. Catalog Description: Assembly language; addressing techniques; subroutines; macros; system input/output; interrupts and traps; assemblers; linkers; loaders; microprocessors. 2. Prerequisites: CS245 3. Required/Elective: This course is an elective in the BS program. | |
|  |  | |
| **Course Goals:** | The Student Learning Outcomes that are addressed by the course are:   1. *Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.* 2. *Students will have a strong foundation in the design, analysis, and application of many types of algorithms.* 3. *Students will have a fundamental understanding of computer systems.*   Other outcomes of instruction:  Students will be able to:   * Write Assembly Language code in stand-alone programs and interfaced with high level languages. * Understand low level programming, the two pass assembly process, and the Fetching Cycle. * Use interrupts and understand their mechanism. * Understand basic computer architecture and computer hardware. * Use files at the Assembly Language level. | |
| **Topics:** | Introduction to Computer Organization and Assembly Language  Simple MASM programs, Assembly Directives  Loops  Procedures, numeric input/output  Macros, Midterm Exam  Recursion  System Interrupts  Files  Applications  Overview | |
| **Lab Projects:** | Five programming projects using MASM. | |
|  | 1 | 3 -4 Simple output only programs |
|  | 2 | Numeric input/output procedures |
|  | 3 | Program using Macros and Recursion |
|  | 4 | Programming with date files |
|  | 5 | Programming with System Interrupts. |
|  |  |  |
| **Academic Integrity:** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and be reported to the appropriate university authorities. | |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed accommodation. | |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 345 |
| **Course name** | Unix and Shell Programming |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Jiang Guo |
| **Text book** | Sobell, *A Practical Guide to Linux. Commands, Editors, and Shell*  *Programming*, Prentice Hall, 2008 ISBN-13: 978-0-13-147823-7  References:  Tansley, David,. *Linux and Unix Shell programming*, Addison Wesley, 2000.  Deitel, Deitel, Nieto, McPhie,. *Perl How To Program*, Prentice Hall, 2003. |
| **Course Information** | 1. Catalog Description: A theoretical and practical study of the UNIX operating system and shell programming. Topics: Shell commands and utilities, UNIX file system, UNIX shells, UNIX graphical user interfaces, and shell programming 2. Prerequisites or co-requisites: CS 203 3. Required/Elective: This course is an elective in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #4. Students will have a fundamental understanding of computer systems.*  Other outcomes of instruction:   1. Write shell scripts effectively. 2. Create scripts to automate common tasks in Unix system, and to   guard against malicious intents against the Unix operating system.   1. Create reports |
| **Brief list of topics to be covered** | * Unix commands: command options. * Advanced VI commands. * Unix commands: piping and redirections. * Unix power utilities: sed and awk. * Introduction to Unix shell programming. * Using shell scripts to automate common tasks. * Using shell scripts to generate report and online report. * Introduction to Perl. * Creating report and online report using Perl cgi and Mysql queries. |
| **Laboratory Projects** | |  | | --- | | Students will complete in-class laboratory assignments on selected topics as well as individual and group homework assignments. For these assignments students are to implement the required visual and functional features using the appropriate technologies.   * Apply Unix commands to solve common problems. * Apply Unix commands, piping and redirections to solve a common problem. * Apply sed and awk to solve common problems and generate report. * Creating shell scripts to automate common tasks. * Creating shell scripts to generate report and online report. * Generating perl cgi scripts to generate report and online report. | |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course Number** | CS370 |
| **Course Name** | Parallel and Distributed Programming |
| **Credits** | 4 units |
| **Contact Hours** | 4 hours/week |
| **Coordinator** | Raj Pamula |
| **Text book** | No textbook is required  Other supplemental resources:  <http://www.calstatela.edu/faculty/rpamula/cs370/cs370fall.htm> |
| **Course Information** | 1. Catalog Description: Parallel programming techniques; abstract models of hardware and operating systems to support parallel programs. 2. Prerequisites or co-requisites: CS203, CS245 3. Required/Elective: This course is an elective in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:   1. *Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.* 2. *Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.* 3. *Students will have a strong foundation in the design, analysis, and application of many types of algorithms.* 4. *Students will have a fundamental understanding of computer systems.* 5. *Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.* |
|  | Other outcomes of instruction: Students get a good understanding of |
|  | * Parallel hardware architectures |
|  | * Parallel programming models |
|  | * Parallel algorithms |
|  | * Writing parallel programs |
| **Brief list of topics:** | * Parallel hardware architectures * Parallel programming models * Shared memory model * Shared memory access * Distributed memory model * Operating Systems support * Clusters * Message Passing Interface * Parallel Algorithms * Efficiency and timing analysis |
|  |  |
| **Lab Projects:** | The students complete lab 5 to 6 projects on above topics |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
| **Course number** | CS 386 | |
| **Course name** | Introduction to Automata Theory | |
| **Credits** | 4 units | |
| **Contact hours** | 4 hours/week | |
| **Coordinator** | Valentino Crespi | |
| **Text book** | Peter Linz*. An Introduction to Formal Language and Automata*. Jones & Bartlett Pub (fourth edition), 2006.  Other texts:   * J. Hopcroft, R. Motwani, J. Ullman. *Introduction Automata Theory, Languages and Computation*. Addison-Wesley. * Arto Salomaa. *Computation and Automata.* Cambridge University Press. * Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. The MIT Press. * Michael Sipser. *Introduction to the Theory of Computation.* Thomson. * [Christos H. Papadimitriou](http://www.amazon.com/exec/obidos/search-handle-url/index=books&field-author-exact=Christos%20H.%20Papadimitriou&rank=-relevance%2C%2Bavailability%2C-daterank/102-5510037-5765706). *Computational Complexity*. Addison-Wesley. | |
| **Course Information** | 1. Course Description: formal approach to automata theory; finite state machines, regular expressions, regular languages. Develops mathematical foundation for Computer Science. 2. Prerequisites: Math 248 (Discrete Mathematics), CS202 (Introduction to Object Oriented Programming). 3. This course is required in the BS program. | |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #1. *Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  Other outcomes of instruction:   |  | | --- | | 1. Understand and manipulate formal descriptions of languages, automata and grammars with focus on Regular and Context Free Languages, Finite State Automata and Regular Expressions. 2. Apply rigorously formal mathematical methods to prove properties of languages, grammars and automata. | | |
| **Brief list of topics to be covered** | * Mathematical Preliminaries and Proof Techniques: sets, alphabets, strings, languages, proofs by contradiction and structural induction. * Grammars and Automata: language generation and language recognition. * DFSA and NDFSA. The Subset construction. * Regular Languages and Regular Expressions. * Minimization of DFSA. * Linear Grammars. * Pumping Lemma for Regular Languages. * Context Free Languages. * Parse Trees. Leftmost and rightmost derivations. Ambiguity. * PDA and DPDA. * Pumping Lemma for Context Free Languages. | |
| **Laboratory Projects** | This class does not possess a laboratory module. Class projects are at the discretion of the instructor. Projects range from weekly assignments to a couple of class projects over the course of the term. | |
| **Academic**  **Integrity** | | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. | |
| **ADA Statement** | | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. | |

**ABET Course Syllabus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Title** | Principles of Data Base Systems | | |
| **Course Number** | CS 422 | **Coordinator** | Chengyu Sun |
| **Total Credit** | 4 | **Contact Hours** | 4 hours/week |

**Textbook**

* Edward Sciore. *Database Design and Implementation*, Wiley, 2008.
* Ramez Elmasri and Shamkant Navathe. *Fundamentals of Database Systems*, Addison Wesley, 1994-2010.
* Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*, *3rd edition*, Mc.Graw Hill, 2002.

**Course Information**

This course is an elective in the BS program.

a) Catalog Description: Normal forms, database system architecture, query optimization, file structures, transaction management, data warehouses, object-oriented databases, databases for e-commerce.

b) Prerequisites or Co-requisites: CS122, CS 312.

**Course Goals**

At the end of the course, students are able to

* Understand Entity-Relationship (ER), relational, and object-oriented data models.
* Be proficient in query languages including relational algebra, relational calculus, and SQL.
* Design and implement complex databases schemas using ER diagrams, normalization, integrity constraints, and advanced database system features such as stored procedures and triggers.
* Understand the internals of a database management system including disk access, buffer management, failure recovery, concurrency control, query execution, and indexes.
* Improve database performance using hardware, software, and query tuning techniques.

These course goals contribute to the success of **Student Learning Outcomes (SLO) 1, 3, and 4**:

* *SLO1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*
* *SLO3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*
* *SL04. Students will have a fundamental understanding of computer systems.*

**Major Topics Covered in the Course:**

* Introduction to database system architecture
* Entity-Relationship model and ER diagram
* Relational data model
* Functional dependencies and normalization
* Relational algebra and relational calculus
* Structured Query Language (SQL)
* Transactions, stored procedures, and triggers
* Disk access and buffer management
* Failure recovery
* Concurrency control
* Indexes
* Query execution and query tuning

**Laboratory Projects**

Each week students either work on a homework assignment or complete a 2-hour in-class lab on selected topics:

* Week 1: Exercises on ER diagram.
* Week 2: Develop a database schema using ER diagram and translate it into relational model.
* Week 3: Normalize database schema to BCNF/3NF.
* Week 4: Construct and evaluate queries in SQL.
* Week 5: Design and implement stored procedures and triggers.
* Week 6: Exercises on disk access and buffer management.
* Week 7: Exercises on failure recovery algorithms.
* Week 8: Implementation of one or more locking protocols.
* Week 9: Exercises on indexes.
* Week10: Exercises on query execution and query tuning.

**ABET Course Syllabus**

|  |  |
| --- | --- |
| **Course number** | CS 437 |
| **Course name** | Software Engineering |
| **Credits** | 5 units |
| **Contact hours** | 7 hours/week |
| **Coordinator** | Prof. Jiang Guo |
| **Text books** | Software Engineering , Ian Sommerville Addison Wesley; 9th edition, 2011, ISBN: 978-0137035151  Recommended Textbook: "Software Engineering: A Practitioner's Approach", Roger Pressman, McGraw-Hill, 7th edition, 2009, ISBN: 0073375977. |
| **Course Information** | 1. Catalog Description: Methodologies and tools for the development, implementation, integration, testing, evaluation, and maintenance of software systems. Software quality assurance; ethical issues in software development. Group laboratory project and oral and written presentations. This course satisfies the upper division writing requirement. Lecture 4 hours, laboratory 3 hours 2. Prerequisites or co-requisites: CS 312 & CS 337 3. Required/Elective: This course is required in the BS program. |
| **Course Goals** | The Student Learning Outcomes (SLO) that are addressed by the course are:  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*  *SLO #7. Students will be able to communicate effectively orally and in writing.*  Other outcomes: At the end of the course, students are able to   * *Estimate the cost and effort for software projects* * *Make schedules for software projects.* * *Elicit software requirements* * *Create data model, flow-oriented model and behavior model* * *Convert the requirement models into software architectures* * *Implement component-level design* * *Implement system-level design* * *Produce a test plan for all requirements* |
| **Brief list of topics to be covered** | * Software Detailed Design * Project Management & Scheduling * Elements of Rapid Software Development * Verification and Validation * Software Quality Assurance * Software Configuration Management * Risk Analysis and Risk Mitigation * Economic Valuation of a Project and Metrics |
| **Laboratory Projects** | The students will be divided into groups, preferably 4 students in each group, to complete a big project based on a selected topic from their survey or instructor.   1. Estimate the effort of a selected project 2. Analyze a flow-oriented model for the selected project 3. Design software architecture based on requirements 4. Create the component-level and a system-level design 5. Implement system components 6. Integrate system components and elaborate the Test Plan 7. System Demo and Presentation |
| **Format of the Software Design Document & Test Plan Document (SDD/TPD)** | This course cs437 is strongly linked to the cs337 class and both classes form the "Software Engineering" main skeleton around which all the undergraduate software engineering is built. The students will start reviewing analyzing and completing the SRD (Software Requirement Document) created by students in the cs337 class, and then based on this fully edited document they will complete the Design, Implementation and Testing sections of the document. To correctly perform this function students and instructor will use the Computer Science Department official rubrics <http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/> |
| **Academic Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously.  All parties involved will receive a grade of F for the course and be reported to the Academic Senate. |

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
| **Course number** | CS 440 | |
| **Course name** | Introduction to Operating Systems | |
| **Credits** | 4 units | |
| **Contact hours** | 4 hours/week | |
| **Coordinator** | Vladimir Akis | |
| **Text book** | Silberschatz, Galvin and Gagne, Operating System Concepts, Addison-Wesley | |
| **Course Information** | 1. Catalog Description: Resource, memory and process management; concurrent processing; networking and distributed systems. Lecture 4 hours. 2. Prerequisites or co-requisites: CS 245, CS 312 3. Required/Elective: This course is required in the BS program. | |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #4. Students will have a fundamental understanding of computer systems.  Other outcomes of instruction:  At the end of the course, students are able to   * Familiar with the main concepts of modern Operating Systems * Familiar with interrelationships among users and hardware components * Familiar with process and threads management * Familiar with CPU scheduling, * Familiar with process synchronization and deadlocks handling * Familiar with memory management and storage management | |
| **Brief list of topics to be covered** | 1. Introduction to Operating System 2. Overview of the operating system components 3. Operating system Structures 4. Process Management 5. Processes 6. Threads 7. CPU Scheduling 8. process Synchronization 9. Deadlocks 10. Memory Management 11. Basic Memory Management 12. Virtual Memory Management 13. Storage Management 14. File System Interface 15. File System Implementation 16. Mass Storage Structure 17. I/O Systems | |
| **Laboratory Projects** | |  | | --- | | No lab component for this course | |  | | |
| **Academic**  **Integrity** | | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. | |
| **ADA Statement** | | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. | |

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
| **Course number** | CS 447 | |
| **Course name** | Computer Network Configuration and Management | |
| **Credits** | 4 units | |
| **Contact hours** | 4 hours/week | |
| **Coordinator** | Huiping Guo | |
| **Text book** | Lammle, Todd., CCNA: Cisco Certified Network Associate Study Guide, Deluxe Edition, Sybex Inc., 2001. | |
| **Course Information** | 1. Catalog Description: Overview of principles and concepts in computer networks and distributed systems;  network structures, topology, architecture, and related software. 2. Prerequisites or co-requisites: Math270, CS440 3. Required/Elective: This course is an elective in the BS program. | |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #4. Students will have a fundamental understanding of computer systems.*  *SLO #7. Students will be able to communicate effectively orally and in writing.*  *SLO #8. Students will have the knowledge, skills, and attitudes for lifelong self-development.*  .  Other outcomes of instruction:  At the end of the course, students have a good understanding of   1. Computer network concepts 2. Computer network devices 3. Computer network protocols | |
| **Brief list of topics to be covered** | 1. Networking concepts 2. Network devices such as  router, switch, bridge, hub, transceiver, NIC, cabling 3. Network topologies such as Ethernet, token ring, ATM,  xDSL, Frame Relay 4. Network protocols such as  TCP/IP, ARP,  ICMP, SNMP,  and DHCP 5. Routing protocols such as RIP, OSPF, EIGRP 6. Access Control Lists 7. Configuration of Cisco router  using Cisco IOS 12.2 8. Configuration of Cisco switches | |
| **Laboratory Projects** | Students complete 4-5 lab projects (two weeks each) and gain practical experience in a networking lab through the following lab projects:   1. Configure Cisco routers 2. Configure Cisco Ethernet switches 3. Capturing and decoding Ethernet packets. 4. Special project | |
|  |  | |
| **Academic**  **Integrity** | | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. | |
| **ADA Statement** | | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. | |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 450 |
| **Course name** | Foundations of Computer Graphics |
| **Credits** | 5 units |
| **Contact hours** | 7 hours/week (lecture 4 hours and laboratory 3 hours) |
| **Coordinator** | Eun-Young Elaine Kang |
| **Text book** | F. S. Hill, Computer Graphics using OpenGL, 3rd Ed., Prentice Hall, 2007 |
| **Course Information** | a) Catalog Description: Programming in object oriented graphics environment implementing primitive operations in two and three dimensions. Image modeling using affine transformations, polygonal meshes and other topics.  b) Prerequisites or co-requisites: CS 203, MATH 208, and MATH 255  c) Required/Elective: This course is an elective in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*  Other outcomes of instruction: At the end of the course, students are able to  1. Describe how graphics display devices work and what graphics primitives are.  2. Explain and work with coordinate spaces, coordinate conversion, and transformations of graphics objects.  3. Explain the graphics modeling process and create polygonal meshes models.  4. Describe the 3D graphics rendering pipeline.  5. Create virtual scenes, transform objects, and work with a camera.  6. Acquire skills in programming with OpenGL and 3D modeling with Maya. |
| **Brief list of topics to be covered** | This course covers topics in 2D rendering and 3D rendering/modeling: lines, polygons, windows/viewports, transformations, polygonal meshes, and rendering pipeline. It also introduces widely used graphic libraries/tools in academia and industry such as OpenGL and Maya.   |  |  | | --- | --- | | 1. Introduction to Computer Graphics  2. Drawing Graphics Primitives  3. Linear Algebra (Vector Tools)  4. Transformation of Objects  5. Clipping  6. Modeling  7. Three Dimensional Viewing  8. Others: Shading Models, Texture Mapping, Raster Display and Aliasing, Curve and Surfaces |  | |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed accommodation. |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 451 |
| **Course name** | Multimedia Software Systems |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Eun-Young Elaine Kang |
| **Text book** | 1. Ze-Nian Li, Mark S Drew, Fundamentals of Multimedia (Hardcover), Prentice Hall, October 22, 2003. ISBN: 0130618721.  2. Handouts and reading materials (technical papers) provided by the instructor |
| **Course Information** | a) Catalog Description: Introduction to Multimedia Information and Processing. Topics: Basic Signal Processing. Color Space, Formations of Image, Video, and Audio data. Current standards and the state-of-art techniques for multimedia systems.  b) Prerequisites or co-requisites: CS312 Data Structures and Algorithms or CS 342 Object Oriented Programming Using C++  c) Required/Elective: This course is an elective in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*  Other outcomes of instruction: At the end of the course, students will be able to  1. Describe the current multimedia data types (images, video, audio, graphics etc).  2. Identify with the requirements and the algorithms for multimedia systems.  3. Implement efficient design solutions and established standards for multimedia.  4. Gain programming experiences in multimedia processing.  5. Develop a multimedia software system related to video (audio) codec, multimedia database, or other multimedia software application on network. |
| **Brief list of topics to be covered** | 1. Introduction to Multimedia.  2. Digital Data Acquisition  3. Coding Theory  4. Image Compression Techniques and Standards  5. Color Theory  6. Video Compression Techniques and Standards  7. Audio Compression Techniques and Standards  8. 2D/3D Graphics  9. MPEG-4 |
| **Laboratory Projects** | Out of class projects: Throughout the course, students are required to complete three to five projects. Each project is designed to be completed in one or two weeks based on the recommended schedule and topics below. Projects are to be suitable for scaffolding.  1. Week 1-2: Be familiar with an image input/output method.  2. Week 3-4: Implement a lossless entropy coding algorithm.  3. Week 4-7: Implement a lossy image compression/decompression algorithm.  4. Week 7-10: Implement a multimedia software system related to video (audio) codec, multimedia database, or other multimedia software application on network. |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed accommodation. |

|  |  |  |  |
| --- | --- | --- | --- |
| **ABET Course Syllabus** | | | |
| **Course number** | | CS 454 | |
| **Course name** | | Topics in Advanced Computer Science  *Introduction to 3D Computer Game Programming* | |
| **Credits** | | 4 units | |
| **Contact hours** | | 4 hours/week | |
| **Coordinator** | | Eun-Young Elaine Kang | |
| **Text book** | | Steve Rabin, Introduction to Game Development, Thomson Charles River Media. | |
| **Course Information** | | a) Catalog Description: Current topics of special interest to students in computer science, as announced in Schedule of Classes. May be repeated to a maximum of 20 units of credit as topic changes.  b) Prerequisites or co-requisites: As needed for a specific topic.  c) Required/Elective: This course is an elective in the BS program. | |
| **Course Goals** | | The Student Learning Outcomes addressed by the course are:  *SLO #1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*  Other outcomes of instruction: At the end of the course, students are able to  1. Describe game genres and game design process.  2. Use a game programming language and work under a common game programming architecture.  3. Explain and describe topics, issues and solutions in game math, collision detection, and physics, 2D/3D graphics, and animation.  4. State a list of game engines and their features.  5. Prototype a 3D Game using a Game Engine.  6. Use modeling and animation tools and create custom models and animations. | |
| **Brief list of topics to be covered** | | Introduction to 3D computer game programming. Topics include 3D game genre and styles; 3D game engines and their components; Scripts; GUI; Models; Textures; Sound and Music; Network. Hands on Experience and Rapid Development.  1. Introduction to Computer Games (Genre, Styles, and Designs)  2. Introduction to 3D Computer Graphics and Mathematics for Game programming  3. Introduction to Common Game Programming Languages (Python or C#)  4. Introduction to Open-source 3D Game Engines  5. Graphics, 3D Modeling and Animation  6. Controls and Event in a Game Engine  7. Collision Detection and Handling  8. AI in Game  9. Game Physics | |
|  | | |
| **Laboratory Projects** | Out of class projects: During the course, students are required to complete a game development project that is implementing a 3D single player video game individually or in group. The project is designed to consist of four to six sub- tasks. Each task is designed to be completed in one or two weeks. | |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 460 |
| **Course name** | Artificial Intelligence |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Russ Abbott |
| **Text book** | Russell, S and P. Norvig, *Artificial Intelligence a Modern Approach.* |
| **Course Information** | 1. *Catalog Description:*  Knowledge representation; problem solving strategies and search algorithms; applications from such areas as theorem proving, expert systems, natural language processing, robotics, and pattern recognition. 2. *Prerequisites or co-requisites:* CS 312 3. *Required/Elective:* This course is an elective in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are: none listed, but the following make sense.  *SLO #1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *Other outcomes of instruction:*   * Students will have been introduced to the theory and technologies of artificial intelligence. |
| **Brief list of topics to be covered** | * Intelligent agents * Search * Knowledge and Reasoning * Planning * Reasoning with Uncertainty |
| **Laboratory Projects** | *In class projects:*  None  *Out of class projects:* Homework and development of a project that applies artificial intelligence techniques to a non-trivial problem. |
| **Academic**  **Integrity** | *Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities.* |
| **ADA Statement** | *Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation.* |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 461 |
| **Course name** | Machine Learning |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Russ Abbott |
| **Text book** | Witten, I. et. al. (2011) *Data Mining*, Morgan Kaufmann. |
| **Course Information** | 1. *Catalog Description:* Means that enable computers to perform tasks for which they were not explicitly programmed; learning paradigms include inductive generalization for examples, genetic algorithms, and connectionist systems such as neural nets. 2. *Prerequisites or co-requisites:* CS 312 3. *Required/Elective:* This course is an elective in the BS program. |
| **Course Goals** | *The Student Learning Outcomes that are addressed by the course are:* none on the list, but the following are reasonable.  *SLO #1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #2. Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*  *SLO #7. Students will be able to communicate effectively orally and in writing.*  Other outcomes of instruction:   * To introduce students to tools and techniques for modeling complex systems and for the automatic creation computer programs. Subsidiary goals will depend on the approach(es) the instructor chooses to take.   + To introduce students to the theories, tools, and technologies used to study complexity, including evolutionary computing and agent-based modeling.   + To introduce students to inductive generalization from examples and other traditional learning paradigms.   + To introduce students to the use of artificial neural nets for learning. |
| **Brief list of topics to be covered** | * Agent-based modeling * Modeling probability density functions and optimization in artificial neural networks, decision trees, Gaussian process regression (k-Nearest Neighbor and expectation-maximization algorithm), Bayesian networks, Markov Random Fields, and support vector machines. * Complex systems; the nature of emergence, evolutionary programming and optimization through evolutionary programming |
| **Laboratory Projects** | *In class projects:* none  *Out of class projects:* Weekly assignments and term project |
| **Academic**  **Integrity** | *Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities.* |
| **ADA Statement** | *Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation.* |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS470 |
| **Course name** | Computer Networking Protocols |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Huiping Guo |
| **Text book** | [Computer Networking : A Top-Down Approach Featuring the Internet](http://www.amazon.com/Computer-Networking-Top-Down-Approach-5th/dp/0136079679/ref=ntt_at_ep_dpi_1) (5th Edition) |
| **Course Information** | 1. Catalog Description: Study of computer network layered architecture and protocols. Topics to be covered include: network architecture, data link layer, addressing, LAN, network layer, transport layer and network applications. 2. Prerequisites or co-requisites: CS245, CS312 3. Required/Elective: This course is an elective in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.  SLO #3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.  SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.  SLO #8. Students will have the knowledge, skills, and attitudes for lifelong self-development.  Other outcomes of instruction:   1. Students will understand the basic concepts of computer network and layered protocol architecture. 2. Students will understand the Internet architecture and the TCP/IP protocol suite, and details of representative protocols at the application, transport network and network layer. 3. Students will have experience in application-layer network programming using TCP and UDP sockets. 4. Students will have experience in network simulation using OPNET. |
| **Brief list of topics to be covered** | 1. Introduction 2. Data link layer 3. Internetworking and ARP 4. OPNET 5. Network layer and IP 6. IP addressing and Subnetting 7. Routing algorithms 8. RIP 9. ICMP, OSPF, BGP 10. NAT, UDP, DHCP 11. TCP overview, segment structure 12. TCP  reliable data transmission 13. TCP  flow control and congestion control 14. Application protocols: FTP, HTTP, SMTP and DNS |
| **Laboratory Projects** | In class projects:   1. **Build your home network**: Introduction to OPNET Software 2. **Subnetting**: Learn the functions of network devices 3. **PING and RIP**: Explore functions of RIP through PING 4. **OSPF**: Explore functions of OSPF 5. **TCP flow control**: Study how the window size affects performance of TCP 6. **Why congestion**: Study the factors that cause network congestion   Out of class projects:   1. **What can you do with home network?**  * how to compare and analyze simulation results of multiple scenarios * How to Improve network performance  1. **TCP Congestion Control**  * Reinforce TCP congestion control algorithms taught in class * How to improve critical thinking |
|  |  |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS480 |
| **Course name** | Cryptography and Information Security |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Huiping Guo |
| **Text book** | Behrouz A. Forouzan, Cryptography and Network Security, McGraw Hill Higher Education,   ISBN:  0072870222 |
| **Course Information** | 1. Catalog Description: This course covers both principles and practice of cryptography and information security. Topics covered include basic concepts of cryptology, classical ciphers, modern symmetric ciphers (DES, IDEA, RC5), Advanced Encryption Standard (AES), public key cryptography, data integrity and digital signature schemes. 2. Prerequisites or co-requisites: CS245, CS312 3. Required/Elective: This course is an elective in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #1. Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.  SLO #3. Students will have a strong foundation in the design, analysis, and application of many types of algorithms.  SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.  SLO #8. Students will have the knowledge, skills, and attitudes for lifelong self-development.  SLO #9. Students will have the ability to analyze the local and global impact of computing on individuals and society.  Other outcomes of instruction:   1. Students will understand basic security concepts 2. Students will understand the traditional and modern security algorithms 3. Students will understand some important network security protocols |
| **Brief list of topics to be covered** | 1. Introduction 2. Mathematics of cryptography 3. Traditional ciphers 4. Modern symmetric ciphers 5. Modes of operation 6. Asymmetric key cryptography 7. Message authentication and hash functions 8. Digital signature 9. Key management 10. SSL |
| **Laboratory Projects** | RSA Encryption: implement the well-known RSA algorithm for public-key cryptography |
|  |  |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 486 |
| **Course name** | Computability and Intractability |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Valentino Crespi |
| **Text book** | Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, second edition, 2005.  Other texts:   * Arto Salomaa. *Computation and Automata.* Cambridge University Press. * J. Hopcroft, R. Motwani, J. Ullman. *Introduction Automata Theory, Languages and Computation*. Addison-Wesley. * Peter Linz. *An Introduction to Formal Languages and Automata*. Jones and Barlett Publishers. * Hartley Rogers, Jr. *Theory of Recursive Functions and Effective Computability*. The MIT Press. * Michael Sipser. *Introduction to the Theory of Computation.* Thomson. * Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman & Company (June 1979). * [Christos H. Papadimitriou](http://www.amazon.com/exec/obidos/search-handle-url/index=books&field-author-exact=Christos%20H.%20Papadimitriou&rank=-relevance%2C%2Bavailability%2C-daterank/102-5510037-5765706). *Computational Complexity*. Addison-Wesley. |
| **Course Information** | 1. Course Description: Theory of Computing; nondeterminisms, decidability and unsolvable problems; NP completeness and intractable computations. 2. Prerequisites: CS386 Introduction to Automata Theory. 3. This course is elective in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #1. *Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  Other outcomes of instruction:   |  | | --- | | * Understand the rigorous notion of algorithm and study the decidability of computational problems. In particular students will learn to distinguish between recursive, recursively enumerable and non recursively enumerable problems. * Understand the role of nondeterministic computations and characterize the class of problems NP. In particular students will learn the concept of NP-completeness and its importance in the P vs NP question. | |
| **Brief list of topics to be covered** | * Turing Machines. Decision problems and languages. The Halting Problem. Recursive, Recursively Enumerable and non Recursively Enumerable languages. Reductions. Rice's Theorem and the Recursion Theorem. * Time and Space Complexity of a Turing Machine. Nondeterministic Turing Machines. * The classes P and NP. Polynomial time many-one reductions. * Cook's Theorem. NP-completeness and the P vs NP question. |
| **Laboratory Projects** | This class does not possess a laboratory module. Class projects are at the discretion of the instructor. Projects range from weekly assignments to a couple of class projects over the course of the term. |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 488 |
| **Course name** | Compilers |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Valentino Crespi |
| **Text book** | Aho, Alfred V. and Sethi, Ravi and Ullman, Jeffrey D. Compilers - Principles, Techniques, and Tools, Addison Wesley,  References:  J. Louden, Kenneth C. Compiler Construction, Principles and Practice, PWS Publishing Company, 1997. |
| **Course Information** | 1. Course Description: Compiler construction;  lexical analysis, including regular languages and finite-state acceptors;  syntactic analysis, including parsing techniques and grammars;  code generation and optimization. 2. Prerequisites: CS312, CS332, CS386 3. This course is an elective in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #1. *Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  SLO #4. Students will have a fundamental understanding of computer systems.  Other outcomes of instruction:   |  | | --- | | Students have a good understanding of:   1. The organization of a compiler 2. The concepts of scanning, parsing, and translation 3. Compiler writing tools 4. Modular development of a significant programming system | |
| **Brief list of topics to be covered** | * Introduction to compiler design. * Symbol tables. * Lexical analysis. * Syntactic analysis – top down and bottom up parsing schemes. * The use of compiler-writing tools: automated parsers and lexical analyzers. * Error recovery. * Semantic analysis. * Translation of source to an intermediate language. * Translation of intermediate language to object code. * Optimization of object code. |
| **Laboratory Projects** | The students will work on various phases of compiler design for a simple programming language using object-oriented techniques and modular design. |
| **Academic  Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will betaken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
|  |  |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | CS 490 |
| **Course name** | Computer Science Recapitulation |
| **Credits** | 2 units |
| **Contact hours** | 2 hours/week |
| **Coordinator** | Raj Pamula |
| **Text book** | No textbook is required |
|  | Other supplemental materials: [GRE Computer Science Practice test books #1](http://csns.calstatela.edu/download.html?fileId=2731320)  [GRE Computer Science Practice test books #2](http://csns.calstatela.edu/download.html?fileId=2731323) |
| **Course Information** | 1. Catalog Description: A recapitulation of the primary concepts of Computer Science in preparation for the Major Field Achievement Test. 2. Prerequisites or co-requisites: MATH 270, CS 386, 437, 440, EE 444 3. Required/Elective: This course is required in the BS program. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO#1: Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO#2: Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*  *SLO #3: Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*  *SLO #4: Students will have a fundamental understanding of computer systems.* |
| Other outcomes of instruction:  Students enhance their problem solving and presentation skills as they prepare solutions to several conceptual questions and present them in class. |
| **Brief list of topics to be covered** | All core course topics in the undergraduate curriculum revolving around the four student learning outcomes   * Theory   + Concepts and techniques from all core courses. Emphasis on Math248, CS312, CS386, and CS337 * Programming languages   + CS201-203, CS332FLC * Algorithms   + CS203, Math248, CS312, CS332FLC * Systems   + Database, web, hardware, network and operating systems. Emphasis on CS120, CS122, CS245, CS440, EE444 |
| **Laboratory Projects** | Students take many quizzes and exams similar to the Major Field Test. |
| **Grading**  **Policy** | The students will be evaluated and graded based MFT and other exams. 50% of the grade will be based on the percentile score on the MFT (final exam), 30% of the grade is determined by exams/quizzes, and 20% of the grade for assignments/presentations. |
| **Academic**  **Integrity** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities. |
| **ADA Statement** | Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed  accommodation. |

**ABET Course Syllabus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Title** | Software Design Laboratory | | |
| **Course Number** | CS 496A | **Coordinator** | Chengyu Sun |
| **Total Credit** | 2 | **Contact Hours** | 6 hours/week |

**Textbook:** None.

**References:**

* Ian Sommerville. *Software Engineering (9th Edition)*, Addison Wesley, 2010.
* Steve McConnell. *Code Complete: A Practical Handbook of Software Construction*, Microsoft Press, 2004.
* Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley Professionals, 1994.

**Course Information**

This course is required in the BS program.

a) Catalog Description: Approaches to software design, including Design Patterns and other strategies for designing software systems. Students working individually or as a group will propose a substantial software project, resulting in a preliminary report and project presentation. Laboratory 6 hours. Graded ABC/NC.

b) Prerequisites or Co-requisites: CS312, CS320, CS337, CS386, and senior standing.

**Course Goals**

This is the first part of a three-quarter senior design project sequence in which each student must participate in a group project under the supervision of a faculty advisor. The goals of the course are

* To improve the ability of the students to undertake complex software projects by guiding them through the early stages of a project development cycle, which include problem and requirement analysis, and the research of tools, libraries, and technologies.
* To improve the ability of the students to perform independent learning of new technologies and concepts.
* To improve the oral communication skills of the students through oral presentations and interaction with project stake holders.
* To improve the written communication skills of the students through the writing of a Project Requirement Document.

These course goals contribute to the success of **Student Learning Outcomes (SLO) 5, 6, 7, and 8**:

* *SLO5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*
* *SLO6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*
* *SLO7. Students will be able to communicate effectively orally and in writing.*
* *SLO8. Students will have the knowledge, skills, and attitudes for lifelong self-development.*

**Laboratory Projects**

The students will be divided into groups with 3 to 4 students in each group. Each group works on a senior design project provided by faculty or industry sponsors. The project is to be completed in three quarters. At the end of CS496A each group should complete the problem and requirement analysis phase and submit a Project Requirement Document.

**Grading Policy**

The students will be evaluated and graded based on the following rubrics:

* *Software Engineering - Requirements*, evaluated by the project faculty advisor.
* *Teamwork*, evaluated by the project faculty advisor and the project group members.
* *Oral Communication*, evaluated by the project faculty advisor.
* *Written Communication*, evaluated by the project faculty advisor.

Details of the rubrics can be found online at <http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/>

**ABET Course Syllabus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Title** | Software Design Laboratory | | |
| **Course Number** | CS 496B | **Coordinator** | Chengyu Sun |
| **Total Credit** | 2 | **Contact Hours** | 6 hours/week |

**Textbook:** None.

**References:**

* Ian Sommerville. *Software Engineering (9th Edition)*, Addison Wesley, 2010.
* Steve McConnell. *Code Complete: A Practical Handbook of Software Construction*, Microsoft Press, 2004.
* Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley Professionals, 1994.

**Course Information**

This course is required in the BS program.

a) Catalog Description: Software design, including Design Patterns and other strategies for designing software systems. Students will make further progress on the project initiated in CS 496A resulting in an intermediate report and project presentation. Laboratory 6 hours. Graded ABC/NC.

b) Prerequisites or Co-requisites: CS496A.

**Course Goals**

This is the second part of a three-quarter senior design project sequence in which each student must participate in a group project under the supervision of a faculty advisor. The goals of the course are

* To improve the ability of the students to undertake complex software projects by guiding them through the design stage of a project development cycle, in which an overall system architecture is established and the requirements developed in CS496A are allocated to hardware or software components of the system.
* To improve the ability of the students to apply their knowledge of software engineering to the design of a complex system.
* To improve the oral communication skills of the students through oral presentations and interaction with project stake holders.
* To improve the written communication skills of the students through the writing of a Project Design Document.

These course goals contribute to the success of **Student Learning Outcomes (SLO) 6, 7, and 8**:

* *SLO6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*
* *SLO7. Students will be able to communicate effectively orally and in writing.*
* *SLO8. Students will have the knowledge, skills, and attitudes for lifelong self-development.*

**Laboratory Projects**

The students will be divided into groups with 3 to 4 students in each group. Each group works on a senior design project provided by faculty or industry sponsors. The project is to be completed in three quarters. At the end of CS496B each group should complete the system design phase and submit a Project Design Document.

**Grading Policy**

The students will be evaluated and graded based on the following rubrics:

* *Software Engineering - Design*, evaluated by the project faculty advisor.
* *Teamwork*, evaluated by the project faculty advisor and the project group members.
* *Oral Communication*, evaluated by the project faculty advisor.
* *Written Communication*, evaluated by the project faculty advisor.

Details of the rubrics can be found online at <http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/>

**ABET Course Syllabus**

|  |  |  |  |
| --- | --- | --- | --- |
| **Course Title** | Software Design Laboratory | | |
| **Course Number** | CS 496C | **Coordinator** | Chengyu Sun |
| **Total Credit** | 2 | **Contact Hours** | 6 hours/week |

**Textbook:** None.

**References:**

* Ian Sommerville. *Software Engineering (9th Edition)*, Addison Wesley, 2010.
* Steve McConnell. *Code Complete: A Practical Handbook of Software Construction*, Microsoft Press, 2004.
* Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison Wesley Professionals, 1994.

**Course Information**

This course is required in the BS program.

a) Catalog Description: Software design, including Design Patterns and other strategies for designing software systems. Students will complete the project initiated in CS 496AB resulting in a final formal report and project presentation. Laboratory 6 hours. Graded ABC/NC.

b) Prerequisites or Co-requisites: CS496B.

**Course Goals**

This is the last part of a three-quarter senior design project sequence in which each student must participate in a group project under the supervision of a faculty advisor. The goals of the course are

* To improve the ability of the students to undertake complex software projects by guiding them through the late stages of a project development cycle including implementation and testing.
* To improve the ability of the students to apply their programming skills in implementing a complex system.
* To improve the oral communication skills of the students through oral presentations and interaction with project stake holders.
* To improve the written communication skills of the students through the writing of a final Project Report.

These course goals contribute to the success of **Student Learning Outcomes (SLO) 6, 7, and 8**:

* *SLO6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*
* *SLO7. Students will be able to communicate effectively orally and in writing.*
* *SLO8. Students will have the knowledge, skills, and attitudes for lifelong self-development.*

**Laboratory Projects**

The students will be divided into groups with 3 to 4 students in each group. Each group works on a senior design project provided by faculty or industry sponsors. The project is to be completed in three quarters. At the end of CS496C each group should complete the implementation and testing of the system, and submit a final Project Report.

**Grading Policy**

The students will be evaluated and graded based on the following rubrics:

* *Software Engineering - Implementation*, evaluated by the project faculty advisor.
* *Teamwork*, evaluated by the project faculty advisor and the project group members.
* *Oral Communication*, evaluated by the project faculty advisor.
* *Written Communication*, evaluated by the project faculty advisor.

Details of the rubrics can be found online at <http://csns.calstatela.edu/wiki/content/assessment/undergrad/Skill_Evaluations/>

|  |  |  |
| --- | --- | --- |
| **ABET Course Syllabus** | | |
| **Course number** | EE 444 | |
| **Course name** | Computer Architecture | |
| **Credits** | 4 units | |
| **Contact hours** | 4 hours/week | |
| **Coordinator** | Charles Liu | |
| **Text book** | Computer Organization and Architecture, Designing for Performance, Eight Edition, by William Stallings, Prentice Hall | |
| **Course Information** | 1. Description: Computing Systems. Integrated study of computer hardware and firmware. Introduction to parallel architectures. 2. Prerequisites: CS 245 3. Required/Elective: This course is required in the CS BS program. | |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  SLO #4. *Students will have a fundamental understanding of computer systems.*  Other outcomes of instruction: At the end of the course, students are be familiar with:   * Computer organization and Architecture * Computer evolution * Processor, Memory, and I/O System * Parallel Architectures | |
| **Brief list of topics to be covered** | 1. Number systems    * Binary addition, subtraction, multiplication    * Hexadecimal systems 2. Boolean Algebra and Gates    * Logic gates and logic circuits    * Evaluation of logical expressions 3. Introduction to Computer Organization    * vonNeumann model 4. Processor Design    * Organization, instruction formats, ALU 5. Control Unit Design    * Clock, Instruction sequencing, Different types of design 6. Memory Organization    * Virtual memory, Cache memory 7. Input/Output systems 8. Introduction to parallel architectures | |
| **Academic Integrity:** | Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and be reported to the Academic Senate. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ABET Course Syllabus**    1. Department, Course Number, and Course Title: | | | | | |  | |
|  | | | | | | | |
| Mathematics | | | |  | | | |
| MATH 206 Calculus I: Differentiation (4) | | | | | | | |
|  | | | | | | | |
| 2. Designation: | Required | 🗹 | Elective | | 🞎 | | |
|  | Lower Division | 🗹 | Upper Division | | 🞎 | | |
|  | | | | | | | |
| 3. Course Description: | Functions, graphs, limits, continuity, derivatives, antidifferentiation, and applications. | | | | | | |
|  | | | | | | | |
| 4. Prerequisites: | MATH 104A AND 104B, each with a minimum *C* grade *or* satisfactory score on placement examination; students with a grade less than B- in either MATH 104A or MATH 104B must enroll concurrently in MATH 206P. | | | | | | |
|  | | | | | | | |
| 5. Text and Materials: | Calculus, Early Transcendentals, 7th ed., Stewart, Brooks-Cole, 2012. | | | | | | |
|  | | | | | | | |
| 6. Course Objectives: | This course is designed to teach the students the principles and techniques of differential calculus of functions of one real variable with a selection of typical applications | | | | | | |
|  | | | | | | | |
| Course Outcomes  Students who successfully complete this course will be able to: | | | | | | | |
| * Compute the limit of a function using numerical, graphical and algebraic approaches. * Determine if a function is continuous at a real number. * Apply the Intermediate Value Theorem and Mean Value Theorem. * Interpret the derivative as either the slope of a tangent line or instantaneous rate of change. * Find the derivative of a function as a limit. * Apply differentiation formulas to compute derivatives. * Compute derivatives of polynomial, trigonometric, exponential and logarithmic functions. * Apply differentiation to solve related rate problems and optimization problems, including applications to related fields. * Compute local and global extrema of a function. * Use implicit differentiation. * Graph functions using methods of calculus. * Compute basic antiderivatives. | | | | | | | |
|  | | | | | | | |
| 7. Topics Covered: | (in Order of Presentation) | | | | | | |
|  |  | | | | | | |
| * Review of functions (Ch. 1) * Idea of limit and continuity (Ch. 2) * Computation of limits (Ch. 2) * Derivative as a slope (Ch. 2) * Derivative as a rate of change (Ch. 2) * Derivative as a limit (Ch. 2) * Computation of derivatives (Ch. 3) * Implicit differentiation (Ch. 3) * Related rates (Ch. 3) * Linear approximations and differentials (Ch. 3) * Application to graphing (Ch. 4) * Application to optimization problems (Ch. 4) * Antiderivatives (Ch. 4) * Initial Value Problems (Ch. 4) | | | | | | | |
|  | | | | | | | |
| 8. Class Schedule: | Number of Sessions per week: | | 4 lectures | | | | |
|  | Duration of each session: | | Lectures | | | 50 minutes | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 9. Contribution of course to meeting the professional component: | | | | | |  | |
|  | | | | | |  | |
| This course is part of the one year (24 quarter units) of Basic Mathematics. | | | | | | | |
|  | Mathematics | | 4 units | | |  | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 10. Relationship of course to program objectives: | | | | | |  | |
|  | | | | | |  | |
| This course relates to the program objectives by contributing to the following measurable outcomes at the level indicated for all computer science graduates: | | | | | | | |
| *SLO #1: Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #7: Students will be able to communicate effectively orally and in writing.* | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| 11. Prepared by: | Melisa Hendrata | | | | | | 01/2012 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ABET Course Syllabus**    1. Department, Course Number, and Course Title: | | | | | |  | |
|  | | | | | | | |
| Mathematics | | | |  | | | |
| MATH 207 Calculus II: INTEGRATION (4) | | | | | | | |
|  | | | | | | | |
| 2. Designation: | Required | 🗹 | Elective | | 🞎 | | |
|  | Lower Division | 🗹 | Upper Division | | 🞎 | | |
|  | | | | | | | |
| 3. Course Description: | The definite integral, Fundamental Theorem of Calculus, transcendental functions, methods of integration, applications to physics and biology. | | | | | | |
|  | | | | | | | |
| 4. Prerequisites: | MATH 206 with a minimum *C* grade; students with a grade less than B- in MATH 206 must enroll concurrently in MATH 207P. | | | | | | |
|  | | | | | | | |
| 5. Text and Materials: | Essential Calculus: Early Transcendentals, Stewart, Thomson/Brooks/Cole, 2007. | | | | | | |
|  | | | | | | | |
| 6. Course Objectives: | This course is designed to teach the students the principles and techniques of integral calculus of functions of one real variable with a selection of typical applications | | | | | | |
|  | | | | | | | |
| Course Outcomes  Students who successfully complete this course will be able to: | | | | | | | |
| * Understand the definition of the integral of a function. * Compute definite, indefinite, and improper integrals using the various techniques of integration. * Understand the fundamental theorem of calculus. * Compute the area between two curves and the volume of a solid of revolution. * Compute the arc length of a curve. * Sketch curves described by parametric equations or polar coordinates and use calculus to study these curves. | | | | | | | |
|  | | | | | | | |
| 7. Topics Covered: | (in Order of Presentation) | | | | | | |
|  |  | | | | | | |
| * Review of antiderivatives (Ch. 4) * Integral as limit of Riemann sums (Ch. 5) * Fundamental theorem of calculus (Ch. 5) * Substitution rule (Ch. 5) * Integration by parts (Ch. 6) * Trigonometric integrals and trigonometric substitution (Ch. 6) * Partial fractions (Ch. 6) * Improper integrals (Ch. 6) * Areas between curves (Ch. 7) * Volumes of solids of revolution (Ch. 7) * Arc length of curves (Ch. 7) * Parametric curves (Ch. 9) * Polar coordinates (Ch. 9) * Areas and lengths in polar coordinates (Ch. 9) | | | | | | | |
|  | | | | | | | |
| 8. Class Schedule: | Number of Sessions per week: | | 4 lectures | | | | |
|  | Duration of each session: | | Lectures | | | 50 minutes | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 9. Contribution of course to meeting the professional component: | | | | | |  | |
|  | | | | | |  | |
| This course is part of the one year (24 quarter units) of Basic Mathematics. | | | | | | | |
|  | Mathematics | | 4 units | | |  | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 10. Relationship of course to program objectives: | | | | | |  | |
|  | | | | | |  | |
| This course relates to the program objectives by contributing to the following measurable outcomes at the level indicated for all computer science graduates:  *SLO #1: Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #7: Students will be able to communicate effectively orally and in writing.* | | | | | | | |
|  | | | | | | | |
| 11. Prepared by: | Melisa Hendrata | | | | | | 01/2012 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ABET Course Syllabus**    1. Department, Course Number, and Course Title: | | | | | |  | |
|  | | | | | | | |
| Mathematics | | | |  | | | |
| MATH 208 Calculus III: SEQUENCES, SERIES,  AND COORDINATE SYSTEMS (4) | | | | | | | |
|  | | | | | | | |
| 2. Designation: | Required | 🗹 | Elective | | 🞎 | | |
|  | Lower Division | 🗹 | Upper Division | | 🞎 | | |
|  | | | | | | | |
| 3. Course Description: | Limits of sequences and series, indeterminate forms, Taylor series, plane coordinate systems, and change of coordinates. | | | | | | |
|  | | | | | | | |
| 4. Prerequisites: | MATH 207 with a minimum *C* grade; students with a grade less than B- in MATH 207 must enroll concurrently in MATH 208P. | | | | | | |
|  | | | | | | | |
| 5. Text and Materials: | Essential Calculus: Early Transcendentals, Stewart, Thomson/Brooks/Cole, 2007. | | | | | | |
|  | | | | | | | |
| 6. Course Objectives: | This course is designed to teach the students the infinite series, convergence results of such series, different coordinate systems and their interrelationships. | | | | | | |
|  | | | | | | | |
| Course Outcomes  Students who successfully complete this course will be able to: | | | | | | | |
| 1. Compute the limit of a convergent sequence algebraically. 2. Determine whether or not a sequence converges. 3. Apply the limit rules for sequences to find the limit of a sequence. 4. Compute the sum of a geometric series or a telescoping series. 5. Use tests (integral, comparison, limit comparison, alternating series, ratio, and root) to determine whether a series converges. 6. Determine whether a series is absolutely convergent. 7. Compute the radius of convergence and interval of convergence of a power series. 8. Represent functions by their Maclaurin and Taylor series and determine the interval of convergence. 9. Multiply, divide, integrate, and differentiate power series. 10. Sketch cylinders and quadratic surfaces in the three dimensional coordinate system. | | | | | | | |
|  | | | | | | | |
| 7. Topics Covered: | (in Order of Presentation) | | | | | | |
|  |  | | | | | | |
| * Sequences (Ch. 8) * Convergence tests for series (Ch. 8) * Power series (Ch. 8) * Taylor and Maclaurin series (Ch. 8) * Three-dimensional coordinate systems (Ch. 10) * Vectors (Ch. 10) * The dot and cross products (Ch. 10) * Equations of lines and planes (Ch. 10) * Cylinders and quadric surfaces (Ch. 10) * Vector functions and space curves (Ch. 10) * Arc length and curvature (Ch. 10) * Motion in space (Ch. 10) | | | | | | | |
|  | | | | | | | |
| 8. Class Schedule: | Number of Sessions per week: | | 4 lectures | | | | |
|  | Duration of each session: | | Lectures | | | 50 minutes | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 9. Contribution of course to meeting the professional component: | | | | | |  | |
|  | | | | | |  | |
| This course is part of the one year (24 quarter units) of Basic Mathematics. | | | | | | | |
|  | Mathematics | | 4 units | | |  | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 10. Relationship of course to program objectives: | | | | | |  | |
|  | | | | | |  | |
| This course relates to the program objectives by contributing to the following measurable outcomes at the level indicated for all computer science graduates:  *SLO #1: Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #7: Students will be able to communicate effectively orally and in writing.* | | | | | | | |
|  | | | | | | | |
| 11. Prepared by: | Melisa Hendrata | | | | | | 01/2012 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ABET Course Syllabus**  1. Department, Course Number, and Course Title: | | | | | |  | |
|  | | | | | | | |
| Mathematics | | | |  | | | |
| MATH 248: DiSCRETE MATHEMATICS (4) | | | | | | | |
|  | | | | | | | |
| 2. Designation: | Required | 🗹 | Elective | | 🞎 | | |
|  | Lower Division | 🗹 | Upper Division | | 🞎 | | |
|  | | | | | | | |
| 3. Course Description: | Fundamentals of logic and set theory, counting techniques, relations, induction and recursion, graphs and trees. | | | | | | |
|  | | | | | | | |
| 4. Prerequisites: | MATH 207. | | | | | | |
|  | | | | | | | |
| 5. Text and Materials: | *Essentials of Discrete Mathematics,* 2nd ed. by David Hunter, Publisher: Jones & Bartlett Learning, 2012. ISBN: 978-1-4496-0442-4 | | | | | | |
|  | | | | | | | |
| 6. Course Objectives: | This course is designed to teach the students basic concepts of discrete mathematics, such as logic, sets, math induction, graph theory, and counting techniques. | | | | | | |
|  | | | | | | | |
| Course Outcomes  Student who successfully complete this course will: | | | | | | | |
| * be able to translate between English sentences and logical expressions, including use of quantifiers. * be able to apply logical thinking to mathematical reasoning (definitions, theorems, proofs, counter examples). * be able to determine an appropriate method for proving a given theorem. * be knowledgeable about foundations of set theory, including the formal definitions of functions and relations. * be able to identify equivalence relations and equivalence classes. * be familiar with modular arithmetic. * have fundamental knowledge of graph theory concepts. * be able to identify isomorphic graphs and find the Euler and Hamiltonian paths and circuits. * be able to distinguish between recursive and explicit definitions of function, and to obtain one from the other. * be able to think recursively. * be able to perform proof by induction. * be able to apply basic counting techniques (permutation, combination, pigeonhole principle). | | | | | | | |
|  | | | | | | | |
| 7. Topics Covered: | (in Order of Presentation) | | | | | | |
|  |  | | | | | | |
| * Formal, propositional and predicate logic (Ch. 1) * Methods of proof (Ch. 1) * Sets, functions and relations (Ch. 2) * Equivalence relations (Ch. 2) * Modular arithmetic (Ch. 2) * Basic graph theory (Ch. 2) * Recurrence relations (Ch. 3) * Proof by induction (Ch. 3) * Optional topic: recursive data structures (Ch. 3) * Counting techniques (permutations, combinations, pigeonhole principle) (Ch. 4) * Selected topics: algorithms (Ch. 5) or discrete-time population models (Ch. 6) | | | | | | | |
|  | | | | | | | |
| 8. Class Schedule: | Number of Sessions per week: | | 2 lectures | | | | |
|  | Duration of each session: | | Lectures | | | 1 hour and 40 minutes | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 9. Contribution of course to meeting the professional component: | | | | | |  | |
|  | | | | | |  | |
| This course is part of the one year (24 quarter units) of Basic Mathematics. | | | | | | | |
|  | Mathematics | | 4 units | | |  | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 10. Relationship of course to program objectives: | | | | | |  | |
|  | | | | | |  | |
| This course relates to the program objectives by contributing to the following measurable outcomes at the level indicated for all computer science graduates:  *SLO #1: Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #7: Students will be able to communicate effectively orally and in writing.* | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| 11. Prepared by: | Melisa Hendrata | | | | | | 01/2012 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ABET Course Syllabus**  1. Department, Course Number, and Course Title: | | | | | |  | |
|  | | | | | | | |
| Mathematics | | | |  | | | |
| MATH 255: INTRODUCTION TO MATRIX THEORY (4) | | | | | | | |
|  | | | | | | | |
| 2. Designation: | Required | 🗹 | Elective | | 🞎 | | |
|  | Lower Division | 🗹 | Upper Division | | 🞎 | | |
|  | | | | | | | |
| 3. Course Description: | Vector spaces, linear transformations, linear equations, matrices, determinants, eigenvectors and eigenvalues, canonical forms. | | | | | | |
|  | | | | | | | |
| 4. Prerequisites: | MATH 208. | | | | | | |
|  | | | | | | | |
| 5. Text and Materials: | *Elementary Linear Algebra A Matrix Approach,* 2nd ed. by Spence, Insel and Friedberg, Publisher: Pearson Prentice Hall, 2007. ISBN: 978-0-13-187141-0. | | | | | | |
|  | | | | | | | |
| 6. Course Objectives: | This course is designed to teach the students basic concepts of linear algebra. | | | | | | |
|  | | | | | | | |
| Course Outcomes  Student who successfully complete this course will: | | | | | | | |
| * be able to perform matrix operations * be able to solve systems of linear equations * be able to compute matrix determinants and inverses * be able to determine linear dependence/independence of vectors in Euclidean spaces * have fundamental knowledge of bases and change of bases of Euclidean spaces * have fundamental knowledge of linear transformations in Euclidean spaces * have fundamental knowledge of general vector spaces and subspaces * have fundamental knowledge of eigenvectors, eigenvalues and eigenspaces * have fundamental knowledge of canonical forms | | | | | | | |
|  | | | | | | | |
| 7. Topics Covered: | (in Order of Presentation) | | | | | | |
|  |  | | | | | | |
| * Matrices and vectors (Ch. 1) * Systems of linear equations (Ch. 1) * Gaussian elimination (Ch. 1) * Span and linear independence (Ch. 1) * Matrix multiplication and inverses (Ch. 2) * Determinant of a matrix (Ch. 3) * Linear transformations (Ch .2) * Subspaces (Ch. 4) * Basis and dimension (Ch. 4) * Coordinate systems (Ch. 4) * Eigenvalues and eigenvectors (Ch. 5) | | | | | | | |
|  | | | | | | | |
| 8. Class Schedule: | Number of Sessions per week: | | 2 lectures | | | | |
|  | Duration of each session: | | Lectures | | | 1 hour and 40 minutes | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 9. Contribution of course to meeting the professional component: | | | | | |  | |
|  | | | | | |  | |
| This course is part of the one year (24 quarter units) of Basic Mathematics. | | | | | | | |
|  | Mathematics | | 4 units | | |  | |
|  |  | |  | | |  | |
|  | | | | | | | |
| 10. Relationship of course to program objectives: | | | | | |  | |
|  | | | | | |  | |
| This course relates to the program objectives by contributing to the following measurable outcomes at the level indicated for all computer science graduates:  *SLO #1: Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #7: Students will be able to communicate effectively orally and in writing.* | | | | | | | |
|  | | | | | | | |
|  | | | | | | | |
| 11. Prepared by: | Melisa Hendrata | | | | | | 03/2012 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ABET Course Syllabus**  1. Department, Course Number, and Course Title: | | | | | |  | |
|  | | | | | | | |
| Mathematics | | | |  | | | |
| MATH 270: INTRODUCTION TO PROBABILITY AND STATISTICS (4) | | | | | | | |
|  | | | | | | | |
| 2. Designation: | Required | 🗹 | Elective | | 🞎 | | |
|  | Lower Division | 🗹 | Upper Division | | 🞎 | | |
|  | | | | | | | |
| 3. Course Description: | Descriptive statistics, sample mean and variance, basic rules of probability, conditional probability, independence, random variables, special discrete and continuous distributions, expectation, central limit theorem, application: Markov chains. | | | | | | |
|  | | | | | | | |
| 4. Prerequisites: | MATH 208, MATH 248. | | | | | | |
|  | | | | | | | |
| 5. Text and Materials: | *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers,* 2nd ed. by Yates and Goodman, Publisher: Wiley, 2004. ISBN-10: 0471272140, ISBN-13: 978-0471272144. | | | | | | |
|  | | | | | | | |
| 6. Course Objectives: | This course is designed to teach the students basic concepts of probability and statistics. | | | | | | |
|  | | | | | | | |
| Course Outcomes  Student who successfully complete this course will be able to: | | | | | | | |
| * Determine frequencies and relative frequencies * Draw a histogram of the data * Construct a stem-and-leaf display of the data * Calculate the values of the sample mean and median * Find the Standard Deviation, Lower fourth, Upper fourth, Fourth spread, Trimmed Mean * Construct a boxplot * Draw a Venn diagram * Compute a probability * Apply counting techniques to solve permutations and combinations problems * Draw a tree diagram to determine a probability * Find a conditional probability * Compute Expected Value E(X) and Variance V(X) of a discrete random variable * Calculate and graph the cumulative distribution function of a discrete/continuous random variable * Find a probability using binomial tables, cumulative distribution function B(x;n,p) * Find a probability using binomial probability mass function b(x;n,p) * Find a probability using Poisson distribution table, cumulative distribution function F(x;λ) * Find a probability using Poisson probability mass function p(x;λ) * Find a probability of an exponential distribution * Compute Expected Value E(X) and Variance V(X) of a continuous random variable * Obtain a cdf (cumulative distribution function), expected value, standard deviation, and variance of a continuous random variable using integration * Find the (100p)th percentile of a distribution * Compute a probability by standardizing to normal distribution * Calculate a joint probability in two variables * Determine the marginal pmf of a discrete random variable * Find Expected Value, Covariance, and Correlation of a discrete joint distribution * Construct the joint probability table | | | | | | | |
|  | | | | | | | |
| 7. Topics Covered: | (in Order of Presentation) | | | | | | |
| * Basic probability axioms (Ch. 1) * Conditional probability and independent events (Ch. 1) * Counting techniques (Ch. 1) * Discrete random variables: expected value, variance, and cumulative distribution function (Ch. 2) * Continuous random variables: expected value, variance, standard deviation, and cumulative distribution function (Ch. 3) * Jointly distributed random variables: joint density, joint distribution functions, and marginal densities (Ch. 4) * Stochastic processes (Ch. 10) * Discrete time Markov chains (Ch. 12) | | | | | | | |
|  | | | | | | | |
| 8. Class Schedule: | Number of Sessions per week: 2 lectures | | | | | | |
|  | Duration of each session: Lectures 1 hour and 40 minutes | | | | | | |
|  |  | |  | | |  | |
| 9. Contribution of course to meeting the professional component: | | | | | |  | |
| This course is part of the one year (24 quarter units) of Basic Mathematics. | | | | | | | |
|  | Mathematics | | 4 units | | |  | |
|  |  | |  | | |  | |
| 10. Relationship of course to program objectives: | | | | | |  | |
|  | | | | | |  | |
| This course relates to the program objectives by contributing to the following measurable outcomes at the level indicated for all computer science graduates: | | | | | | | |
| *SLO #1: Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*  *SLO #7: Students will be able to communicate effectively orally and in writing.* | | | | | | | |
|  | | | | | | | |
| 11. Prepared by: | Melisa Hendrata | | | | | | 03/2012 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ABET Course Syllabus**  Department, Course Number, and Course Title: | | | | |
| [Department of Physics and Astronomy](http://ecatalog.calstatela.edu/preview_entity.php?catoid=4&ent_oid=112&hl=%22physics%22&returnto=search) , PHYS 101, Physics (4) | | | | |
|  | | | | |
| Designation: | Required | 🗹 | Elective | 🞎 |
|  | Lower Division | 🗹 | Upper Division | 🞎 |
|  | | | | |
| Course Description: | Mechanics of particles, rigid bodies; gravity; simple harmonic motion. Fundamental treatment of physics without use of calculus. Lecture 3 hours, laboratory 3 hours for each course. | | | |
|  | | | | |
| Prerequisites: | Knowledge of elementary algebra and trigonometry. | | | |
|  | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Text and Materials: | 1. R.A. Serway C. Vuille, & C.A. Bennett.: COLLEGE PHYSICS (8-th Edition) *(Lecture + Lab)* is a calculus-based study on Mechanics. Prerequisites *High School* *Physics, Algebra* *and Trigonometry* are required. The suggested course for chapters 1 through 14 follows the content of the text book. Updated modified versions of syllabi will be available on WebCT. 2. (Supplemental) The supplementary Text book of P. Davidovits on “Physics in Biology and Medicine” (3-rd Edition); ISBN: 0123694116 is required for assigned text material and additional problems. | | |
|  | | | |
| Course Outline: | **Introduction:** Introduction to the Physics Science.  **Dynamics: Newton's Lawas:**Force. Mass. Newton Laws. Free Body Diagram. Problem Solving.  **Friction Force:** Newton Laws. Free Body Diagram. Problem Solving.  **Gravitation:** Newton Gravitation Force. Satellites and "Weightlessness". Keplers’ Laws. Problem Solving.  **Kinematics in One Dimension:** Uniform Motion. Uniform Acceleration. Free Fall. Problem Solving.  **Kinematics in Two Dimensions:** Adding Vectors. Projectile Motion. Problem Solving.  **Work and Energy:** Kinetic Energy. Energy Work Principle. Conservative and Nonconservative Forces. Law of Energy Conservation. Power. Problem Solving.  **Conservation of Energy:** Conservative and Nonconservative Forces. Law of Energy Conservation. Power. Problem Solving.  **Linear Momentum:** Momentum. Impact. Conservation of Momentum. Elastic Collision. Inelastic Collision. Collisions in One and Two Dimensions. Center of Mass. Problem Solving.  **Rotational Motion:** Rotational Dynamics. Rotational Kinetic Energy. Problem Solving.  **Torgue:** Conservation of Angular Momentum. Inertial Forces. Problem Solving.  **Solids and Fluids:** Fluids Statics and Dynamics. Problem Solving. | | |
| Topics Covered: | (in Order of Presentation) | | |
|  |  | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **WEEK** | **Topics** | Sections in Serway and Vuille | **Sections in Davidovits** | **Problems from Davidovits** | | 1 | Mechanics | **Ch. 1** |  |  | | 2 | Motion in One Dimension | **Ch. 2** |  |  | | 3 | Vectors and Two Dimensional Motion | **Ch. 3** | 3-1, 4,5,6,7 | 3-4,6 | | 4 | Laws of Motion | **Ch. 4** |  |  | | 5 | Energy | **Ch. 5** | 3-8 | 3-1,9 | | 6 | Momentum and Collisions | **Ch. 6** |  |  | | 7 | Rotational Motion and Gravity | **Ch. 7** | 4-4,5,6,7 | 4-2,4,5,6 | | 8 | Rotational Equilibrium and Dynamics | **Ch. 8** | 1-3,6 | 1-1,4,7,10 | | 9 | Solids and Fluids | **Ch. 9-7** | 5-1,2,3  7-2,8,11  8-5,7,9,10,11 | 5-1,6,7; 7-2,8,11  8-1,3,4,5,7,9 | | | | |
|  | | | |
| Class Schedule: | Lecture 3 hours per week  Laboratory 3 hours per week |  | |
|  | | | |
| Contribution of course to meeting the professional component: | | |  |
| This course is part of the one year of Basic Sciences. | | | |
|  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ABET Course Syllabus**  Department, Course Number, and Course Title: | | | | |
| [Department of Physics and Astronomy](http://ecatalog.calstatela.edu/preview_entity.php?catoid=4&ent_oid=112&hl=%22physics%22&returnto=search) , PHYS 102, Physics (4) | | | | |
|  | | | | |
| Designation: | Required | 🗹 | Elective | 🞎 |
|  | Lower Division | 🗹 | Upper Division | 🞎 |
|  | | | | |
| Course Description: | Waves, sound, fluids, thermal physics, kinetic theory, electrostatics. Fundamental treatment of physics without use of calculus. Lecture 3 hours, laboratory 3 hours for each course. | | | |
|  | | | | |
| Prerequisites: | PHYS 101. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Text and Materials: | 1. R.A. Serway C. Vuille, & C.A. Bennett.: COLLEGE PHYSICS (8-th Edition) *(Lecture + Lab)* is a calculus-based study on Mechanics. Prerequisites *High School* *Physics, Algebra* *and Trigonometry* are required. The suggested course for chapters 1 through 14 follows the content of the text book. Updated modified versions of syllabi will be available on WebCT. 2. (Supplemental) The supplementary Text book of P. Davidovits on “Physics in Biology and Medicine” (3-rd Edition); ISBN: 0123694116 is required for assigned text material and additional problems. | | |
|  | | | |
| Course Outline: | **Solids and Fluids:** Elasticity Modulus, Pascal′s, Bernoulli′s & Archimedes′s Principles.  **Thermal Physics:** Temperature Scales, Expansion. Ideal Gases, Kinetic Theory, Radiation. Problem Solving.  **Energy in Thermal Processes:** Heat and Internal Energy, Specific Heat and Entropy. Problem Solving.  **Work Diagrams:** 1-st and 2-nd Laws of Thermodynamics, Thermal Processes, Thermal Efficiency, Entropy and Disorder. Problem Solving.  **Vibrations and Waves:** Hook′s Law, Elastic Potential Energy and SHM, Types of Waves, Interference. Problem Solving.  **Sound:** Speed of Sound, Energy and Intensity, Spherical and Plane Waves, Doppler Effect, Standing waves and Resonance. Problem Solving.  **Electric Forces and Fields:** Electrical Charges, Coulomb Forces, Electrical Fields, Gausse′s Law. Problem Solving.  **Electrical Energy:** Electrical Potential, Capacitance. Conductors and Equipotential Surfaces, Combinations of Capacitors, Electrical Energy. Problem Solving.  **Current and Resistance:** Electric Current, Ohm′s Law, Energy and Power, Superconductors. Problem Solving.  **DC Circuits:** Electromotive Force, Combination of Resistors, Kirchhoff′s Law, RC Circuits. Problem Solving.  **Magnetism:** Magnets, Magnetic Field and Forces, RHR and Ampere Law. Problem Solving. | | |
| Topics Covered: | (in Order of Presentation) | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **WEEK** | **Topics** | Sections in Serway and Vuille | **Sections in Davidovits** | **Problems from Davidovits** | | 1 | Solids and Fluids | **Ch. 9-7** | 5-1,2,3  7-2,8,11  8-5,7,9,10,11 | 5-1,6,7; 7-2,8,11  8-1,3,4,5,7,9 | | 2 | Thermal Physics | **Ch. 10** | 9-4,4,5,6 |  | | 2 | Laws of Thermal Processes | **Ch. 11** | 11-1,2,3,4, 5,6,7,8 | 11-3,6,9,11 | | 3 | Laws of Thermodynamics | **Ch. 12** | 10-4,5 |  | | 4 | Vibrations and waves | **Ch. 13** |  |  | | 5 | Sound | **Ch. 14** | 12-3,4 | 12-1 | | 6 | Electric Forces and Fields | **Ch. 15** |  |  | | 7 | Electrical Energy and capacitance | **Ch. 16** |  |  | | 8 | Current and Resistance | **Ch. 17** | 13-1,4  14-2 | 13-1,8  14-1 | | 9 | DC Circuits | **Ch. 18** |  |  | | 10 | Magnetism | **Ch. 19** |  |  | | | | |
|  | | | |
|  | | | |
| Class Schedule: | Lecture 3 hours per week  Laboratory 3 hours per week |  | |
|  | | | |
| Contribution of course to meeting the professional component: | | |  |
| This course is part of the one year of Basic Sciences. | | | |
|  | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ABET Course Syllabus**  Department, Course Number, and Course Title: | | | | |
| [Department of Physics and Astronomy](http://ecatalog.calstatela.edu/preview_entity.php?catoid=4&ent_oid=112&hl=%22physics%22&returnto=search) , PHYS 103, Physics (4) | | | | |
|  | | | | |
| Designation: | Required | 🗹 | Elective | 🞎 |
|  | Lower Division | 🗹 | Upper Division | 🞎 |
|  | | | | |
| Course Description: | Electricity and magnetism, light and optics, relativity, quanta, atoms, nuclei, and fundamental particles. Fundamental treatment of physics without use of calculus; Lecture 3 hours, laboratory 3 hours for each course. | | | |
|  | | | | |
| Prerequisites: | PHYS 102. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Text and Materials: | 1. R.A. Serway C. Vuille, & C.A. Bennett.: COLLEGE PHYSICS (8-th Edition) *(Lecture + Lab)* is a calculus-based study on Mechanics. Prerequisites *High School* *Physics, Algebra* *and Trigonometry* are required. The suggested course for chapters 1 through 14 follows the content of the text book. Updated modified versions of syllabi will be available on WebCT. 2. (Supplemental) The supplementary Text book of P. Davidovits on “Physics in Biology and Medicine” (3-rd Edition); ISBN: 0123694116 is required for assigned text material and additional problems. | | |
|  | | | |
| Course Outline: | * Induced Voltages * And Inductance * Alternating circuits * Reflection and Refraction of Light * Mirrors and Lenses * Wave Optics * Optical Instruments * Quantum Physics * Atomic Physics * Nuclear Physics | | |
| Topics Covered: | (in Order of Presentation) | | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **WEEK** | **Topics** | Sections in Serway and Vuille | **Sections in Davidovits** | **Problems from Davidovits** | | 1 | Induced Voltages  And Inductance | **Ch. 20** |  |  | | 2 | Alternating circuits | **Ch 21** |  |  | | 3 | Reflection and Refraction of Light | **Ch. 22** |  |  | | 4 | Mirrors and Lenses | **Ch. 23** |  |  | | 5 | Wave Optics | **Ch. 24** |  |  | | 6 | Optical Instruments | **Ch. 25** | 15-all | 15-3,4,7,8,9 | | 7 | Quantum Physics | **Ch. 27** |  |  | | 8 | Atomic Physics | **Ch. 28** | 16-6 |  | | 9 | Nuclear Physics | **Ch. 29/30** |  |  | | 10 | Final Exam |  |  |  | | | | |
|  | | | |
|  | | | |
| Class Schedule: | Lecture 3 hours per week  Laboratory 3 hours per week |  | |
|  | | | |
| Contribution of course to meeting the professional component: | | |  |
| This course is part of the one year of Basic Sciences. | | | |
|  | | | |

**ABET Course Syllabus**

**1. Department, Course Number, and Course Title:**

Department of Physics and Astronomy

PHYSICS 211 – Mechanics (5 Units)

**2. Designation:**  Required 🗹 Elective 🞎

Lower Division 🗹 Upper Division 🞎

**3. Course Description:** Motion in one and two dimensions, Newton’s laws of motion, circular motion, work and energy, energy transfer, linear and angular momentum and their conservation, collisions, universal gravitation.

**4. Prerequisites:** High school physics or permission of department; MATH 206 (may be taken concurrently).

**5. Text and Materials:** Physics for Scientists and Engineers with Modern Physics **,** 8’th edition, by Serway and Jewett, 2011.

**6. Course Objectives:** Students will be exposed to a first course in Newtonian Mechanics using differential calculus. They will see how the application of Newton's Laws of Motion and the Conservation of Momentum and Mechanical Momentum can yield solutions to complex 1, 2, and 3 dimensional motion, including rotations and problems in elementary Static Equilibrium and Elasticity.

Course Outcomes:

1. Students will come to appreciate the scalar and vector nature of physical quantities.
2. Students will learn to identify and quantify forces which govern dynamics of particle motion.
3. Students will learn how to use the concepts of work, kinetic energy, and potential energy in quantitative descriptions of dynamics.
4. Students will learn how to employ conservation laws to simplify and to understand motion.
5. Students will learn how to use rotational kinematics and dynamics for complex rigid body motion.

**7. Topics Covered:** (in Order of Presentation, Chapters-1 through 14)

* 1D Motion, Vectors,
* 2D Motion,
* Laws of Motion,
* Circular Motion,
* Work and Energy,
* Potential Energy,
* Momentum,
* Rolling Motion and Torque,
* Angular Momentum,
* Moment of Inertia,
* Gravitation and Fluid Mechanics.

**8. Class Schedule:** Number of Sessions per week: 2 lectures; 1 tutorial; 1 laboratory

Duration of each session: Lecture: 3 hrs, 15 min; Tutorial: 2 hrs

Laboratory: 3 hrs

**9. Contribution of course to meeting the professional component:**

This course is part of the lower-division major requirement

Science: 5 units

**10. Relationship of course to program objectives:**

* Knowledge of the fundamentals of mathematics.
* Knowledge of the fundamentals of science.
* In particular, an ability to apply knowledge of advanced mathematics through multivariate calculus and differential equations.
* Capability to analyze and interpret experimental results.
* Ability to communicate effectively orally and in written reports
* A desire to be a flexible and adaptable team player.

**11. Course coordinator:** A. Khachatourian, Updated by Fred Daneshgaran, 05/2012

**ABET Course Syllabus**

**1. Department, Course Number, and Course Title:**

Department of Physics and Astronomy

PHYSICS 212 – Waves, Optics and Thermodynamics (5 units)

**2. Designation:** Required 🗹 Elective 🞎

Lower Division 🗹 Upper Division 🞎

**3. Course Description:** Mechanical vibrations and sound, optics, elementary thermodynamics.

**4. Prerequisites:** PHYS 211; MATH 207 (prerequisite or corequisite)

**5. Text and Materials:**  Physics for Scientists and Engineers with Modern Phyiscs **,** 8’th edition, by Serway and Jewett, 2011.

**6. Course Objectives:** Students will be introduced to the mechanics of continuous media via wave motion. Thermodynamics expands the concept of the conservation of energy from the limited form discussed in Physics 211 ( for conservative forces) to its broadest application to all forms of energy. The kinetic theory of gases shows how Newton's Laws of motion can be fruitfully applied on a microscopic scale. Entropy is a common concept in thermodynamics and statistical mechanics. Students will see that the many varied types of motion and transformations seen in the physical world can often be understood in terms of Newton's Laws of Motion. Students are introduced to light as waves and learn about interference and diffraction effects.

Course Outcomes:

1. Students will learn about oscillatory motion, such as waves on a string and sound waves, and wave phenomena , such as superposition of waves, energy transported by waves, and resonances.
2. Students will be introduced to thermodynamics and its applications, such as thermometry, heat conduction, First Law of Thermodynamics, heat capacity, heat engines, and entropy.
3. Students will be introduced to the microscopic Kinetic Theory of Gases. They will see the connection between topics discussed in thermodynamics and a microscopic theory that explains these topics based on Newtonian mechanics.
4. Students will learn about the Nature of light, image formation, interference of light waves, diffraction and polarization

**7. Topics Covered**: (in Order of Presentation)

Oscillatory motion

Wave motion

Sound waves

Superposition and standing waves

Temperature, thermal expansion and Ideal Gases

Heat and the First law of Thermodynamics

The Kinetic Theory of Gases

Heat engines, Entropy, and the Second law of Thermodynamics

Light, image formation, interference effects, diffraction and polarization

**8. Class Schedule:** Number of Sessions per week: 2 lectures; 1 tutorial; 1 laboratory

Duration of each session: Lecture: 3 hrs, 15 min; Tutorial: 2 hrs

Laboratory: 3 hrs

**9. Contribution of course to meeting the professional component:**

This course is part of the lower-division major requirement

Science: 5 units

**10. Relationship of course to program objectives:**

* Knowledge of the fundamentals of mathematics.
* Knowledge of the fundamentals of science.
* In particular, an ability to apply knowledge of advanced mathematics through multivariate calculus and differential equations.
* Capability to analyze and interpret experimental results.
* Ability to communicate effectively orally and in written reports
* A desire to be a flexible and adaptable team player.

**11. Course coordinator**: Guo-meng (Peter) Zhao, Updated by Fred Daneshgaran, 05/2012

**ABET Course Syllabus**

**1. Department, Course Number, and Course Title:**

Department of Physics and Astronomy

PHYSICS 213 – Electricity and Magnetism (5 units)

**2. Designation:** Required 🗹 Elective 🞎

Lower Division 🗹 Upper Division 🞎

**3. Course Description:** Elementary field theory, basic electricity and magnetism, DC circuits.

**4. Prerequisites:** PHYS 212; MATH 208 (prerequisite or corequisite)

**5. Text and Materials:** Physics for Scientists and Engineers with Modern Phyiscs **,** 8’th edition, by Serway and Jewett, 2011.

**6. Course Objectives:** Students will learn the fundamentals of electrostatics and magnetostatics. Basic direct current circuit analysis with resistive and capacitive elements will be studied

Course Outcomes:

1. Students will learn about Coulomb's law, Electric fields, electric flux and applications of Gauss' law.
2. Students will learn about the concept of the electric potential, capacitance, and dielectrics.
3. Students will learn about Ohm's law, electric currents, microscopic semi-classical models of electrical conduction, and the use of Kirchoff's rules to solve direct current circuits.
4. Students will be introduced to the effect of magnetic fields on charged particles in motion, the Lorentz force. They will learn that the cumulative effect of the Lorentz force on individual charge carriers results in macroscopic forces and torques on current carrying wires.
5. Students will learn how to apply the Biot-Savart law to calculate the magnetic fields generated by current distributions. Ampere's law will be used to demonstrate how symmetries in the current distribution can be used to simplify magnetic field calculations. They will learn how to include the presence of materials in the computation of magnetic fields.

**7. Topics Covered: (in Order of Presentation)**

* Electric Fields
* Gauss’s Law
* Electric Potential
* Capacitance and Dielectrics
* Current and Resistance
* Direct Current Circuits
* Magnetic Fields
* Sources of the Magnetic Field
* Faraday’s Law
* Electromagnetic Waves

**8. Class Schedule:** Number of Sessions per week: 2 lectures; 1 tutorial; 1 laboratory

Duration of each session: Lecture: 3 hrs, 15 min; Tutorial: 2 hrs

Laboratory: 3 hrs

**9. Contribution of course to meeting the professional component:**

This course is part of the lower-division major requirement

Science: 5 units

**10. Relationship of course to program objectives:**

* Knowledge of the fundamentals of mathematics.
* Knowledge of the fundamentals of science.
* In particular, an ability to apply knowledge of advanced mathematics through multivariate calculus and differential equations.
* Capability to analyze and interpret experimental results.
* Ability to communicate effectively orally and in written reports
* A desire to be a flexible and adaptable team player.

**11. Course coordinator:** A. Lee, Updated by Fred Daneshgaran, 05/2012.

|  |  |
| --- | --- |
| **ABET Course Syllabus** | |
| **Course number** | TECH250 |
| **Course name** | Impact of Technology on the Individuals and Society |
| **Credits** | 4 units |
| **Contact hours** | 4 hours/week |
| **Coordinator** | Ben Lee |
| **Text book** | Hjorth, Linda S., Eichler, Barbara A., Khan, Ahmed S., & Morello, John A. (2008). Technology and Society: Issues for the 21st Century and Beyond, 3rd Ed. Upper Saddle River, N.J.: Pearson. [ISBN: 0-13-119443-7] |
| **Course Information** | 1. An exploration of the ways technology impacts individual human development within global social systems. Including an emphasis on lifelong acquisition of technology skills with an optional service-learning component. 2. Prerequisites: - 3. This course is required in the General Education Block E. |
| **Course Goals** | The Student Learning Outcomes that are addressed by the course are:  *SLO #8. Students will have the knowledge, skills, and attitudes for lifelong self-development.*  *SLO #9. Students will have the ability to analyze the local and global impact of computing on individuals and society.*  Other outcomes of instruction: At the end of the course students are able to:   |  | | --- | | * Explain the dynamic interaction among technology, the individual, society; | | * Identify their responsibilities related to the use of various technologies and their related ramifications; | | * Access, synthesize, and apply information from multiple sources to understand technology and its current and potential impact; | | * Assess the positive and negative factors related to the use of technology in multiple situations; | |  | |
| **Brief list of topics to be covered** | |  | | --- | | Ethics and Technology  The Global Impact of Computer Science  The Social Impact of Computing  Lifelong Self-Development  War, Politics, and Technology  Health and Technology  Technology and the Third World  Emerging and Future Technology Issues | |
| **Laboratory Projects** | Each student will be required to submit an individual research paper in essay format and present a PowerPoint presentation. Students will upload their documents (Word and PowerPoint) to CSNS to the assignment labeled “Social Impact of Computing”. |
| **Grading Policy** Lab Projects 15%, large projects 25%, Midterm1 15%, Midterm2 15%, Final 30%  A 90 – 100  B 80 – 90  C 60 – 80  NC below 60  **Academic** Cheating will not be tolerated. Cheating on any assignment or exam will be **Integrity** taken seriously. All parties involved will receive a grade of F for the course and   are reported to the proper authorities.  **ADA Statement** Reasonable accommodation will be provided to any student who is registered   with the Office of Students with Disabilities and requests needed   accommodation. | | |
|  | | |

**University 401 Syllabus**  
Office: University Writing Center

**Objective** University 401 is a course alternative to the WPE. You are eligible to take this course if you have attempted the WPE at least once without success. In this course you will produce a portfolio of work designed to demonstrate your writing ability. You will plan, write, revise, and proofread approximately five essays, two of which will be selected for inclusion in the portfolio. In addition, you will write one essay exam similar to the WPE, which will also be included in the portfolio. The portfolio will be evaluated by two instructors other than your own. A passing portfolio is equivalent to a passing score on the WPE.

**Required Work** There will be weekly reading and writing assignments in this class. You will need to plan ahead carefully in order to complete the following tasks on time:

1. Occasional reading assignments
2. Five or six short essays (2-3 pages)
3. Two or three revised essays (2-4 pages)
4. One final timed essay
5. Weekly assigned readings (provided by the instructor)
6. Out-of-class readings and reading journal (see “Reading Project,” below)
7. Portfolio (see “Your Portfolio,” below)

**Expectations** Regular attendance is essential. Late work is not acceptable.

**Supplies** You will need some regular, lined notebook (8.5 x 11) paper, some dark-ink pens (blue or black). While there is no required text for this course, you should strongly consider purchasing a decent American language dictionary in addition to a foreign-language dictionary you might already have. Also, you might consider purchasing a writing handbook, such as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. [Note: Some sections are held in a computer lab, and students may also be asked to bring a flash drive.]

**Your Portfolio** For this class, ultimately you will be evaluated on the basis of **three** writing samples: two essays written during the quarter, which are revised and edited (with rough drafts attached beneath the revision), and your final timed essay. Due dates for each revision to be used in your portfolio will be announced.

**Reading Project** Because the acquisition of language skills associated with reading is an essential part of writing improvement, the Reading Project is an important part of this course. You will write reading summaries each week on recent newspaper and magazine articles. Here are the rules for selecting articles to complete the weekly reading project assignment:

* You need to find two articles each week, unless instructed otherwise.
* Each article must be taken from a major newspaper or magazine (no web sites, advertisements, textbooks, professional journals, etc.).
* Each article must have been published during the two weeks prior to the assignment due date.
* You can and should read in areas of your own interest. However, rather than summarizing and responding to articles in *PC Week* or *Tiger Beat* every single week, you should vary the source to expose yourself to different writing styles.

After you have found articles that meet the above conditions, you should set aside 15-20 minutes after reading each article to summarize what you’ve read and what the reading made you think about. Here's the procedure you should follow:

1. Write immediately after reading, or as soon as possible after reading.

2. Do not worry about grammar or correctness. I'll be looking at **content** only.

3. Begin each journal entry by noting the following information:

Article Title, Author, and Source (name of magazine or newspaper)

Below this information write your summary and response as follows:

4. **Summary:** In your summary, please tell about the important points the reading makes as well as how the author establishes these points. Give enough information so that I know what’s going on, but please don’t simply copy sections out of the article. Your summary should be about one-half of a typed page or about one handwritten page.

5. **Response:** The response is your opportunity to express what you think about what you’ve read. Try to connect what you’ve read with your own thoughts and experiences. You can disagree with the author, offer your opinion about what is happening in the reading, or write about what the reading reminds you of in your own life. Your response can be any length, but at a minimum it should be about one-half of a typed page or about one handwritten page.

6. Attach a photocopy of the article.

**Plagiarism** As in all university courses, you are expected to do your own work. Using all or part of another person’s writing and presenting it as your own (in other words, without acknowledging the source) is plagiarism. (See the more complete description of plagiarism in the *Schedule of Classes*.) Plagiarism is a form of cheating and is therefore unacceptable. Students who are caught plagiarizing will fail the course and will be referred to the Student Disciplinary Officer for possible action.

**Grades** This course is graded Credit (CR) or No Credit (NC).

# Appendix B

# Faculty Vitae

This section contains ABET-style resumes for each of the Department of Computer Science faculty members. The resumes are presented for full time faculty followed by part time faculty and listed in alphabetical order as indicated in Table B-1. For the sake of brevity, lists of publications are limited to the most significant or recently published since the previous ABET visit in 2006.

| Faculty Name | Faculty Status |
| --- | --- |
| Russell J. Abbott | Full Time |
| Vladimir Akis | Full Time |
| Valentino Crespi | Full Time |
| Huiping Guo | Full Time |
| Jiang Guo | Full Time |
| Eun-Young “Elaine” Kang | Full Time |
| Raj Pamula | Full Time |
| Behzad Parviz | Full Time |
| Chengyu Sun | Full Time |
| Albert Cervantes | Part Time |
| Edmund Gean | Part Time |
| Mahan Hajianpour | Part Time |
| John Hurley | Part Time |
| Jose M. Macias | Part Time |
| Bapa Rao | Part Time |
| John Tran | Part Time |

**Table B.1: Faculty Table**

# Russ Abbott

**http://www.calstatela.edu/faculty/rabbott**

**rabbott@calstatela.edu**

**Education**

* + Ph.D, Computer Science, University of Southern California, 1973
  + MA, Applied Mathematics, Harvard University, 1963
  + BA, Mathematics, Columbia University, 1962

**Academic experience**

* + 1987-1991 – Professor – Probationary – full time
  + 1991-current – Professor – Tenure – full time

**Non-academic experience**

* + 1978 – present. The Aerospace Corporation. Sr. Engr. Specialist. Computer software consultant in many areas. Both full and part time—e,g, full time over the summer..

**Certifications or professional registrations**

* + None

**Current membership in professional organizations**

* + AAAI

**Honors and awards**

* + Best paper. "If a Tree Grows Rings, is it Telling the Time?" International Journal of Unconventional Computation, July 2008. Adapted from paper presented at the Conference on Unconventional Computation, September 2006.

**Service activities**

* + I am the course coordinator for CS 332F, CS 332L, CS 454UI, CS 460, CS 461.
  + I contribute to academic governance every quarter by participating in department/college/university/system committees (up to 9 hours per week)
* Academic Senate (2010-2011)
* College RTP (2010-2011)
* Dept. Assessment committee (ongoing).

**Publications/Presentations**

* + "[Energy and Information in Complex Systems](https://sites.google.com/site/russabbott/Energyintro.pptx?attredirects=0)*," GECCO 2010* and *Swarmfest 2011.*
  + "[Abstract data types and constructive emergence](https://sites.google.com/site/russabbott/Abbott-Abstractdatatypesandconstructiveemergence.pdf?attredirects=0)" [*Newsletter on Philosophy and Computers*](http://www.apaonline.org/documents/publications/v09n2_Computers.pdf)*of the American Philosophical Society*, v9-n2, Spring 2010, pp 48-56.
  + "If a Tree Grows Rings, is it Telling the Time?" *International Journal of Unconventional Computation*, July 2008.
  + "Abstraction abstracted" *Proceedings of the Workshop on the Role of Abstraction in Software Engineering*, May 2008.
  + "Bits don't have error bars" selected papers from the *Workshop on Philosophy and Engineering*, October 2007.

**Other professional development activities**

* + Swarmfest 2011
  + UI Workshop on Computational Intelligence, 2010
  + GECCO 2010
  + Conference on Engineering and Philosophy, 2009
  + Conference on Unconventional Computation, 2008

**Areas of interest, i.e., areas in which I am interested in teaching courses, advising student projects, and (in some cases) doing research**

* + Artificial intelligence, including constraint programming and rule-based/expert systems
  + Complex systems, including agent-based modeling and evolutionary programming
  + Computing and philosophy, including multi-level ontologies and downward entailment
  + Data science, including data mining, computational intelligence, and machine learning
  + Programming languages, including functional and logic programming
  + User interface technologies, including user interface design and client- side functionality.

# Vladimir N. Akis

# <http://www.calstatela.edu/faculty/vakis>

# [vakis@calstatela.edu](mailto:vakis@calstatela.edu)

**Education**

* MFA, Art, University of California, Los Angeles, 1994
* Ph.D., Mathematics, University of California, Davis, 1982
* MA, Mathematics, University of California, Berkeley, 1977
* BA, Mathematics, California State University, Sacramento, 1976

**Academic Experience**

* 1978 – 1982, Teaching Associate, University of California, Davis
* 1982 – 1986, Assistant Professor, California State University, Los Angeles
* 1986 – 1999, Associate Professor, California State University, Los Angeles
* 1999 – 2012, Professor, California State University, Los Angeles

**Current membership in professional organizations**

* + ATINER (Athens Institute for Education and Research)

**Honors and awards**

1. Bennett Annual Lecture Award, Auburn University
2. Invited Speaker, Spring Topology Conference, University of Utah
3. Invited Speaker, International Topology Conference, Patras, Greece

**Service activities**

* Academic Organizer, 1St - 6th ATINER International Conferences of Mathematics and Statistics, Athens, Greece. 2007, 2008, 2009, 2010, 2011, 2012.
* Principle Editor of the Conference Proceedings, 1St - 6th ATINER International Conferences of Mathematics and Statistics, Athens, Greece. 2007, 2008, 2009, 2010, 2011, 2012.

**Publications/Presentations**

PUBLICATIONS

* "On the Variation of Vector Fields and Fixed Points of Analytic Maps." Preprint.
* Book Editor “Essays in Mathematics and Statistics” ATINER 2009.
* Book Editor “Essays in Mathematics and Statistics vol. 2”, ATINER 2012.

PRESENTATIONS

* ATINER, International Conference of Computer Science and Information Systems, Athens, Greece, Presenter and Session Chair.
* Conference and Workshop on the Moore Method of teaching, invited panelist.
* International Topology Conference, Invited Speaker, Patras, Greece.
* ATINER, International Conference of Mathematics and Statistics, Athens, Greece, Academic Organizer and Session Chair. 2007, 2008, 2009, 2010, 2011, 2012.

**Other professional development activities**

* Director of the Mathematics and Statistics Research Unit at the Athens Institute for Education and Research (ATINER), Athens, Greece. 2007 - present.

**Areas of interest, i.e., areas in which I am interested in teaching courses, advising student projects, and (in some cases) doing research**

* + Computer Graphics.
  + Analysis of Algorithms.
  + Formal Languages and Automata Theory.
  + Topology.
  + Mathematical Foundation of Computer Science.

# Valentino Crespi

# <http://www.calstatela.edu/faculty/vcrespi>

# [vcrespi@calstatela.edu](mailto:vcrespi@calstatela.edu)

**Education**

* Ph.D, Computer Science, University of Milan and Turin, 1997.
* Laurea Degree (summa cum Laude), University of Milan, Italy, 1992.

**Academic experience**

* 1998 – 2000: Eastern Mediterranean University, Assistant Professor.
* 2000 – 2003: Dartmouth College, Research Faculty.
* 2003 – 2009: CSULA, Assistant Professor.
* 2009 – present: CSULA, Associate Professor.

**Current membership in professional organizations**

* ACM

**Service activities**

* Course coordinator for CS 312, CS 386, CS 486, CS 488.
* Chair of the College Instructional Affairs Committee (IAC, since 2003).
* Member of the Graduate Studies Committee.
* Member of the Part-time faculty Review Committee (meets once a year).
* Liaison for the Department to the Campus Library.
* Member of the Research Task Force (2011).

**Publications/Presentations (since 2008 only, more on my web page)**

* Valentino Crespi, Aram Galstyan and Kristina Lerman. Top-Down vs Bottom-up Methodologies in Multi-Agent System Design. Journal of Autonomous Robots, Volume 24, Number 3, pages 303-313, 2008.
* Valentino Crespi, George Cybenko and Guofei Jiang. *The Theory of Trackability with Applications to Sensor Networks*. ACM Transactions on Sensor Networks, Volume 4, Number 2, pages 4-42, 2008.
* G. Cybenko, V. Crespi. Learning Hidden Markov Models using Non-Negative Matrix Factorization. IEEE Transactions on Inf. Theory, volume 57, Number 6, June 2011.
* V. Crespi, G. Cybenko, A. Giani. Attacking and Defending Covert Channels and Behavioral Models. Submitted for publication to IEEE-TDSC, 2011 (at second review).
* Flocon2008 Conference, January 2008, Savannah, GA.
* AFOSR PI Review Meeting, Washington DC, June 2008.
* Viterbi School of Engineering, USC, Invited Talk, 2/2/2011.
* CINQA Seminar Series, Invited Talk, 1/27/2010.
* ISI-USC, Los Angeles, CA, Invited Talk, 11/06/2009.

**Other professional development activities**

* *Scientific Collaborations*: Actively involved in scientific collaborations with Dartmouth College, Hanover, NH, with the ISI, Marina del Rey, CA, and with UC Berkeley, Berkeley, CA.
* Spent Winter 2011 quarter in *Sabbatical* at UC Berkeley.
* *Scientific Journal and Conference Reviewer*: IEEE Transactions on Information Theory, IEEE Transactions on Mobile Computing, ACM Transactions on Sensor Networks, Journal of Distributed Computing. Invited to be an official reviewer of Mathematical Reviews.
* *Publisher Reviewer*: Among the reviewers of one of Horstmann's Java books being published in 2012.
* *Mini Projects in Grant Proposals*: written mini research proposal on Mathematical Methods in Biology for the RISE/MARC Program (education/training), promoted and led by Prof. Dr Carlos Gutierrez.
* Involved in the following *grant projects and related activities*:
  + Title: “Engineering Awareness”, role: **PI**, Type: research grant, Agency: Air Force, Status: **expired**, Period: 2007-2009. Purpose: develop mathematical methods and technologies to monitor effectively the environment.
  + Title: “CSULA Partnership to Enhance G6-12 STEM in LAUSD District Five through Graduate Teaching Fellows”, Role: **research advisor**, Type: educational, Agency: NSF, Status: **currently funded**, Period: 2007-2012. Advised Student (2007): Ankur Bansal.
  + Title: “CREST Center for Energy Sustainability”, Role: key personnel, Status: **currently funded**, Sponsoring Agency: NSF, Type: research, Period: 2009-2014. Obtained supplemental funds in 2011 for the creation of the new Modeling Component. **Current leader** of the Modeling Component.
  + Title: “CINQA Center for Interdisciplinary Quantitative Analysis”, Role: co-PI, Status: **currently funded**, Sponsoring Agency: NIH, Type: educational, research, Period: 2008-2013. ***Developed interdisciplinary minor in Bioinformatics (BINF)*** to be activated in Fall 2012: http://www.calstatela.edu/centers/cinqa/binf/index.php

**Areas of interest, i.e., areas in which I am interested in teaching courses, advising student projects, and (in some cases) doing research**

* + Trackability, Complexification and Machine Learning of Hidden Markov Models.
  + Sensor Networks.
  + Computer Security
  + Social Media Analytics.
  + Complex Systems.
  + Communication and Information Theory.
  + Multi-Target Tracking, Process Query Systems and Stochastic Modeling.
  + Distributed Control, Surveillance, Sensor Networks and UAVs.
  + Combinatorial Optimization, Matrix Computation and Graph Theory.

# Huiping Guo

# <http://www.calstatela.edu/faculty/hpguo>

# [hpguo@calstatela.edu](mailto:hpguo@calstatela.edu)

**Education**

* + Ph.D, Computer Science, University of Ottawa, 2003
  + MA, Computer Science and Engineering, Nanjing University of Aero.& Astro., P.R.China, 1999
  + BA, Computer Science and Engineering, Nanjing University of Aero.& Astro., P.R.China, 1992

**Academic experience**

* + 2011-current – Association Professor – Tenure – full time
  + 2005-2011 – Assistant Professor – Probationary – full time
  + 2003-2005 – Post-Doctoral researcher – full time

**Non-academic experience**

* + 1992 – 1996. software engineer, Luoyang Electro - Optical Technology Development Center (LEODC), P.R. China

**Current membership in professional organizations**

* + ASEE

**Honors and awards**

* Co-PI, Enhancement of Computer Networking Curriculum through OPNET PBL, funded by NSF, 09/2010 **–** 08/2012
* Co-PI, 2009 HP Innovations in Education , 09/2009 – 08/2011

### Co-PI, IMPACT LA – Improving Minority Partnerships and Access through CISE-related Teaching, funded by NSF, 05/2008 – 04/2013

**Service activities**

* + I am the course coordinator for CS 447, CS470, CS 480
  + I contribute to academic governance every quarter by participating in department/college/university/system committees

**Publications/Presentations**

1. "Achieving Simultaneous Distribution Control and Privacy Protection for Internet Media Distribution"  *ACM Transactions on Multimedia Computing, Communications and Applications(TOMCCAP),* Vol. 4, No. 2, 2008
2. "A Multiple Bits Watermark for Relational Data", *Journal of Database Management* , 19(3), pages 1-21,  July-September, 2008
3. Chaining Watermarks for Detecting Malicious Modifications to Streaming Data”, *Information Sciences* , Vol 177, No. 1, 2007, pp281-298
4. “Using Digital Images to Teach Abstract Math and Inspire Students towardsCareers in Computer Science and Engineering”, *ASEE 2011 Annual Conference & Exposition*, June 26-29 2011, Vancouver Canada
5. “Engaging Underrepresented Middle School Students in Engineering and Science through a Two-day Summer Camp”, *ASEE 2011 Annual Conference & Exposition*, June 26-29 2011, Vancouver Canada
6. “Implementing Collaborative Project-Based Learning Using the Tablet PC to Enhance Student Learning in Engineering and Computer Science Courses”, *FIE 2010,* Oct 28, Washington D.C
7. “Web Usage Mining with Fine-Grained Browsing Activity Tracking”. [*IKE 2010*](http://www.informatik.uni-trier.de/%7Eley/db/conf/ike/ike2010.html#ShindeSG10) *11-17,* 12-15 July, Las Vegas, 2010
8. “Internet Voting Protocol Design and Implementation”. *International Conference on Security and Management (SAM’10)*: 406-412, 12-15 July, Las Vegas, 2010
9. “A New Approach to Detecting Malicious Modification to Streaming Data Using Integer Stuffing”. *International Conference on Security and Management (SAM’10)*: 393-399, 12-15 July, Las Vegas, 2010
10. “Strengthening the K-20 Engineering Pipeline for Underrepresented Minorities”*, 2010 ASEE Annual Conference*, June 24th, 2010, Louisville, KY (***best paper award***)
11. “Multiple Watermarking for Relational Data”, *International Conference on security and management (SAM’09),* 13-16 July, Las Vegas, 2009
12. “Tamper Detection and Localization for Categorical Data Using Fragile Watermarks for Multi Owner Databases”,  *International Conference on security and management (SAM’09)*, 13-16 July, Las Vegas, 2009
13. “An improved indexing scheme for range queries”,  *International Conference on security and management (SAM’08)*, 14-17 July, Las Vegas, 2008
14. "The Price of Security: A Performance Study of Web Application Security", *The 2008 International Conference on Internet Computing (ICOMP'08)*, 14-17 July, 2008, Las Vegas

**Other professional development activities**

* + Program Committee Member: CSIE2011, WCECS2007-2010,
  + Session chair: SAM08-09
  + Reviewer: Journal of Multimedia and Application, IEEE Transaction on Image Processing, IEEE Transaction on Signal Processing, International Journal of Intellectual Property Management.

**Areas of interest, i.e., areas in which I am interested in teaching courses, advising student projects, and (in some cases) doing research**

* + Database security and privacy
  + Digital rights management
  + Wireless sensor network security
  + Smartphone security
  + Multimedia communications

# Jiang Guo

# <http://www.calstatela.edu/faculty/jguo>

# [jguo@calstatela.edu](mailto:jguo@calstatela.edu)

**Education**

* + Ph.D, Computer Science, Beijing University of Aeronautics and Astronautics, 1996
  + M.S, Computer Science, Computing Center of Chinese Academy of Sciences, 1992
  + B.S, Computer Science, University of Science and Technology of China, 1989

**Academic experience**

* + 2002-2007 – Assistant Professor – Probationary – full time
  + 2007-2011 – Associate Professor – Tenure – full time
  + 2011-current – Professor – Tenure – full time

**Non-academic experience**

* + None

**Certifications or professional registrations**

* + Microsoft Certified Product Specialist MCPID 358142

**Current membership in professional organizations**

* + IASTED

**Honors and awards**

* + Outstanding Computer Science Professor of the Year

ECST Student Council, College of Engineering, Computer Science, and Technology, California State University Los Angeles. (2005, 2006, 2009)

**Service activities**

* + I am the course coordinator for CS 245, CS345, CS 337, and CS 437.
  + I contribute to academic governance every quarter by participating in department/college/university/system committees (up to 9 hours per week)
* Department RTP (2008-2010)
* Dept. Assessment committee (ongoing).

**Publications/Presentations**

* + "Context-Aware System: CSULA Smart Parking (CSP)," *In Proceedings of the 15th IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA 2011), Washington, DC, USA, May 16 – 18, 2011.*
  + “Implementing Collaborative Project-Based Learning Using the Tablet PC to Enhance Student Learning in Engineering and Computer Science Courses,” *In Proceedings of the 40th Annual Frontiers in Education Conference (FIE 2010), Arlington, Virginia, USA, October 27–30, 2010.*
  + “Design and Implementation of a Telematic Management System” *In Proceedings of the 2010 ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM’10), August 20-22, Yangzhou, China, 2010.*
  + “A Three Layer Network Management System”, *In Proceedings of the 2009 World Congress on Computer Science and Information Engineering (CSIE 2009), Los Angeles/Anaheim, USA, March 31 to April 2, 2009.*
  + *“An Adaptive Architecture for Mobile Wireless Sensor Networks",* Journal of Micronanoelectronic Technology, Volume 44, No. 7/8, 2007.

**Other professional development activities**

* + International Conference on Software Engineering and Applications, 2011
  + [International Conference on Software Engineering](http://www.iasted.org/conferences/home-642.html), 2010
  + International Conference on Granular Computing, 2009

**Areas of interest, i.e., areas in which I am interested in teaching courses, advising student projects, and (in some cases) doing research**

* + Software Engineering, including software re-engineering, object-oriented modeling
  + Web-based Systems, including web-based system performance management and web services
  + Operating Systems, including web-based operating systems and distributed systems
  + Wireless Sensors and Network, including context-aware systems, mobile computing and wireless sensor networks

# Eun-Young Elaine Kang

# <http://www.calstatela.edu/faculty/eykang>

# [eykang@calstatela.edu](mailto:eykang@calstatela.edu)

**Education**

* + Ph.D, Computer Science, University of Southern California, 2003
  + MS, Computer Science, Hongik University, Seoul Korea, 1995
  + BS, Mathematics, Sejong University, Seoul, Korea, 1992

**Academic experience**

* + 1995-1996 – Lecturer and Research Associate – full time, Sejong University, Korea
  + 2004 Jan.-2004 Jun. – Lecturer – part time, University of Southern California
  + 2004-2009 – Assistant Professor – Probationary – full time, Cal. State LA
  + 2009-current – Associate Professor – Tenure – full time, Cal. State LA

**Non-academic experience**

* + None

**Current membership in professional organizations**

* + ACM (Association for Computing Machinery)
  + IEEE (Institute of Electrical and Electronics Engineers)
  + KSEA (Korean-American Scientists and Engineers Association)
  + KOCSEA (Korean Computer Scientists and Engineers Association of America)

**Honors and awards**

* + Professor of the Year, Computer Science Department, College of Engineering, Computer Science, and Technology, California State University, February 2007
  + Academic Achievement Award, University Of Southern California, 2003
  + Honor Student Fellowship, Sejong University, Seoul Korea, 1988-1992

**Service activities**

* + I am the course coordinator for CS 242, CS 332C, CS 450, CS 451, CS 464 (Game).
  + I contribute to academic governance every quarter by participating in department/college/university/system committees
* Univ. External Awards Nominating (2006), Student Educational Equity Advisory Board (2006-2008), Academic Advisement Subcommittee (Spring 2011)
* College Student Affairs (2010-current), Instructional Affairs (Spring 2011)
* Dept. RTP (2010-2011), Industry Advisory Board (IAB), Instructional Affairs, and Part-time Faculty Review (ongoing).

**Publications/Presentations (selected publications of the last two years)**

* V. Mejia and E.Y. Kang, “Automatic Moving Object Detection Using Motion and Color Features and Bi-Modal Gaussian Approximation”, to appear In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, Alaska, October 9-11, 2011.
* Pallavi Bhole and Eun-Young Elaine Kang, “Analysis of Super Resolution Reconstruction Based on Multi-frame Interpolation Using Different Orthonormal Wavelets”, In Proceedings of the 2011 International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV'11), Las Vegas, USA, July 18-21, 2011.
* Milan Mijic, E. Y. E. Kang, T. Longson, "Classical Cosmology Through Animation Stories," Bulletin of the American Astronomical Society, vol. 42. no. 3, p. 885, (journal abstract), 2010.
* N. Warter-Perez, J. Dong, A. Abramyan, M. Castillo, H. Guo, and E. Kang, " Stengthening the K-20 Engineering Pipeline for Underrepresented Minorities," (2010 ASEE Annual Conference & Exposition Best Paper Award), The ASEE (American Society for Engineering Education) Annual Conference, Louisville, Kentucky, Jun. 20-23, 2010
* E.Y. Kang, K. Chansamorn\*, M. Mijic, T. Longson, and J. Woo, "Project and Asset Management System for Multi-Disciplinary Scientific Visualization Projects", the Proc. of the 2010 International Conference on Internet Computing (ICOMP'10), Las Vegas, Nevada, July 12-15, 2010.

**Other professional development activities**

* Reviewer
  + Intl. Symposium on Optical Engineering and Photonic Technology: OEPT 2009
  + IASTED Intl. Conf. on Signal and Image Processing (SIP), 2006, 2007, 2008
  + IASTED Intl. Conf. on Internet and Multimedia Systems and Applications (IMSA), 2007, 2008
  + IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2006
  + Elsevier Image and Vision Computing Journal, 2003
* Conference and Symposium
  + KOCSEA (the Korean Computer Scientists and Engineers Association of America), Editor 2007-2009, Secretary 2010.
  + Publication/Session Chair, KOCSEA Technical Symposium, 2007, 2008, 2010.
  + Program Committee, IASTED International Conference on Signal and Image Processing (SIP), 2006, 2007, 2008
  + Program Committee, IASTED International Conference on Internet and Multimedia Systems and Applications (IMSA), 2007, 2008

**Areas of interest, i.e., areas in which I am interested in teaching courses, advising student projects, and (in some cases) doing research**

* + Computer Vision, including structure from motion, super-resolution and video analysis.
  + Computer Graphics and 3D Game Programming
  + Multimedia Software Systems, including image processing and digital data compression.
  + Artificial Intelligence, including rule-based/expert systems and machine learning

# Raj S Pamula

# <http://www.calstatela.edu/faculty/rpamula>

# [rpamula@calstatela.edu](mailto:rpamula@calstatela.edu)

**Education**

* + Ph.D, Computer Science, Southern Illinois University, Carbondale, 1987
  + MS, Computer Engineering, Indian Institute of Technology, Kharagpur, India, 1982
  + BS, Electronics and Telecommunications, Jawaharlal Nehru Technological University, India,1980

**Academic experience**

* + 1987-1993 – Professor – Probationary – Assistant Professor, full time
  + 1993-2000 – Associate Professor – Tenure – full time
  + 2000- current – Professor and Chair – Tenure – full time

**Non-academic experience**

* + None

**Certifications or professional registrations**

* + None

**Honors and awards**

* + *“Favorite Computer Science Professor Award”* in 2001, 2002, and 2011 and 2012 as voted by the students of Engineering, Computer Science and Technology.

**Service activities**

* + I am the course coordinator for CS101,CS160, CS190, CS290,CS 370 and CS490.
  + I was one of the four founding faculty members of the Computer Science Department. Since its founding the Department has: (i) developed and instituted a now thriving MS program, (ii) recruited five new full-time faculty, (iii) upgraded our hardware and software resources, (iv) modified our undergraduate program, and (v) received ABET accreditation.
  + I have participated at all levels of academic governance during the last five years:

**University.** General Education Subcommittee, Recruitment Taskforce Subcommittee, WASC Assessment Committee

**College.** Instructional Affairs, Faculty Affairs, Student Affairs, Outreach, EagleFest,

**Department.** Instructional Affairs, Assessment, Recruitment, Industry Advisory Board, Retention Tenure and Promotion, Course Coordinator, Webmaster, ProgFest, Part-Time Faculty Review.

My Post Tenure Evaluation reports from both the Department committee in 2006 and 2011 was highly appreciative in all the three categories.

**Publications/Presentations**

* + "Automatic Integration of Web Services", in Proceedings of the Int'l Conference on Internet Technology and Applications (iTAP2011), Wuhan, China, August 2011.
  + “A Three-Layer Network Management System.” World Congress on Computer Science and Information Engineering (CSIE 2009), March 2009, Los Angeles/Anaheim, USA.
  + “Static Analysis Based Software Architecture Recovery,” The Special Issue of Lecture Notes on Computer Science (LNCS), Volume 3982, Springer-Verlag, 2006.
  + “Using Category Theory to Model the Dependencies of the Web Services,” Proceedings of the 2006 International Conference in Computational Science and Its Applications (ICCSA 2006), IEE, Glasgow, Scotland, 2006.
  + “Continuous Improvement of the Computer Science Undergraduate Program at CSULA”, International Conference on Frontiers in Education: computer Science and Computer Engineering (FECS'06), Las Vegas, Nevada, July 2006.

**Other professional development activities**

* + CSISSE: Computer Science, Information Science, Software Engineering chairs of California State University system. (2006-present)
  + LDTP: Lower Division Transfer Pattern evaluation team for California State University. (2001-2008)
  + iTAP2011: Int'l Conference on Internet Technology and Applications, 2011
  + CSIE 2009: World Congress on Computer Science and Information Engineering, 2009

**Areas of interest, i.e., areas in which I am interested in teaching courses, advising student projects, and (in some cases) doing research**

* + Cluster Computing
  + Parallel Algorithms
  + Computer Networks
  + Information Security
  + Web Systems

# Behzad Parviz

# <http://www.calstatela.edu/faculty/bparviz>

# [bparviz@calstatela.edu](mailto:bparviz@calstatela.edu)

**Education**

* + Ph.D., Computer Science, State University of New York, Binghamton, NY 1986
  + M.S., Computer Science, State University of New York, Binghamton, NY 1979
  + M.S., Managerial Science, Marywood University, Scranton, PA, 1976
  + B.S., Cost Accounting, College of Accounting, Tehran, Iran, 1974

**Academic experience**

* + 1986 - 1994 – Assistant Professor – Probationary (full time)
  + 1994 - 2007 – Associate Professor (full time)
  + 2007 - Present – Professor (full time)

**Non-academic experience**

* + 1987 – 1996 Universal Software Engineering. President and professional software development consultant. Part time.

**Certifications or professional registrations**

* + None

**Current membership in professional organizations**

* + None

**Honors and awards**

* + None

**Service activities**

* I am the course coordinator for CS 201, CS 202 and CS203.
* I was one of the four founding faculty members of the Computer Science Department.
* I am the principal graduate advisor.
* I contribute to academic governance every quarter by participating in department/college/university/system committees (1 to 9 hours per week)

**Publications/Presentations**

* + “ A ***New Approach to Detecting Malicious Modification to Streaming Data Using Integer Stuffing,”*** Proceeding of the 2010 International Conference on Security and Management (SAM 2010), Las Vegas, Nevada, USA, 2010.
  + ***“Internet Voting Protocol Design and Implementation,”*** Proceeding of the 2010 International Conference on Security and Management (SAM 2010), Las Vegas, Nevada, USA, 2010.
  + **“*Building a Peer to Peer Message Passing Environment by Utilizing Reflection in .NET,”*** Proceedings of the 2006 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’06), Las Vegas, Nevada, USA,2006.
  + ***“A Collaboration-Oriented Software Architecture Modeling System-JArchiDesigner,”*** Proceedings of the 13th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2006), Potsdam, Germany, 2006.
  + ***“A Performance Validation Tool for J2EE Applications,”*** To be published in the Proceedings of the 13th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2006), Potsdam, Germany, 2006.

**Other professional development activities**

* + Conference on Security and Management, 2010
  + Conference on Parallel and Distributed Processing Techniques and Applications, 2006

**Areas of interest, i.e., areas in which I am interested in teaching courses, advising student projects, and (in some cases) doing research**

* + Programming languages
  + Complex systems analysis
  + Operating Systems
  + Software engineering
  + Database systems

# Chengyu Sun

# <http://www.calstatela.edu/faculty/csun>

# [csun@calstatela.edu](mailto:csun@calstatela.edu)

**Education**

* + Ph.D., Computer Science, University of California, Santa Barbara, 2004
  + BE, Electronic Engineering, Tsinghua University, P.R. China, 1992

**Academic experience**

* + 2010-current – Associate Professor – Tenure – full time, Computer Science Department, California State University, Los Angeles
  + 2004-2010 – Assistant Professor – Probationary – full time, Computer Science Department, California State University, Los Angeles

**Non-academic experience**

* + 1995 – 1996. Network Engineer, SinoChem-Hero Intelligent System Co., Beijing, P.R. China

**Current membership in professional organizations**

* + Association for Computing Machinery (ACM)

**Honors and awards**

* ICDE 2002 Best Research Paper Award. *Exploring Spatial Datasets with Histograms*. In Proceedings of the 18th International Conference on Data Engineering, 2002.

**Service activities**

* + I am the course coordinator for CS120, CS122, CS320, CS422, CS496A, CS496B, and CS496C.
  + I contribute to academic governance by participating in a number of department, college, and university committees
* Academic Senate (Fall 2006 and Summer 2009)
* University Student Policy Committee (Fall 2007 and Winter 2012)
* College Assessment Task Force (2004-2006)
* College Program Assessment Committee (2008)
* Department Assessment Committee (ongoing).

**Publications/Presentations**

* Sweta Shinde, Chengyu Sun. *Web Usage Mining with Fine-Grained Browsing Activity Tracking*. In Proceedings of the 2010 International Conference on Information and Knowledge Engineering (IKE'10), 2010.
* Cheralyn Cofer, Chengyu Sun. *SQL Tuner: A Utility to Assist in the Query Tuning Process*. In Proceedings of the 2009 International Conference on Information and Knowledge Engineering (IKE'09), 2009.
* Chengyu Sun, Huiping Guo. *The Price of Security: A Performance Study of Web Application Security*. In Proceedings of the 2008 International Conference on Internet Computing (ICOMP'08), 2008.
* Nagender Bandi, Chengyu Sun, Divyakant Agrawal, Amr El Abbadi. *Fast Computation of Spatial Selections and Joins Using Graphics Hardware*. In Information Systems, Volume 32, Number 8, pages 1073-1100, 2007.
* Chengyu Sun, Nagender Bandi, Divyakant Agrawal, Amr El Abbdi. *Exploring Spatial Datasets with Histograms*. In Distributed and Parallel Databases, Volume 20, Number 1, pages 57-88, July, 2006.
* Nagender Bandi, Chengyu Sun, Hailing Yu, Divyakant Agrawal, and Amr El Abbdi. *New Hardware Support for Database Operations*. In IEEE Data Engineering Bulletin, Volume 28, Number 2, pages 23-30, June, 2005.

**Other professional development activities**

* + External reviewer for international database conferences and journals: SIGMOD, VLDB, PODS, ICDE, and TODS.
  + Developer and maintainer of CSNetwork Services (CSNS), a web-based system that facilitates teaching, learning, student administration, and program assessment at the Computer Science Department, CSULA.

**Areas of interest, i.e., areas in which I am interested in teaching courses, advising student projects, and (in some cases) doing research**

* + Database systems
  + Web development
  + Web data mining
  + Mobile computing
  + Cloud computing

**Albert F. Cervantes**

**acervan5@calstatela.edu**

**Education**

|  |
| --- |
| Masters of Science Electrical Engineering, Spring 2006, California State University, L.A. Bachelor of Science Computer Engineering, Summer 2004, University of California, Irvine |

**Skills**

|  |
| --- |
| Innovative entrepreneur with strong leadership qualities. Takes initiative whenever opportunity presents itself. Identifies and resolves issues effectively and efficiently. Able to learn and perfect new subject matter quickly. Works well within diverse groups, and independently. Meets deadlines under extreme pressure while maintaining the highest quality of work. |

**Languages**

|  |
| --- |
| C/C++/C#, BASH, HTML, CSS, Javascript, Java, Python, OpenGL, MFCs, MIPS |

**Work & Research Experience**

|  |
| --- |
| **Senior Software Engineer,** Desktop Development Tools, Western Digital Corporation. April 2010 – Present. Develop desktop and Web-based applications to facilitate in the firmware development process, as well as the generation of secure signatures based on X509 Certificates.  **Adjunct Professor,** Computer Science Department, California State University, Los Angeles. Spring 06 – Present. Provide instruction to students on computer science related topics including: computer programming, client and server side development, planning, object oriented design, debugging, and testing.  **Research Fellow,** “Global Registration from a Video Sequence”, California State University, Los Angeles. Winter 04 – Spring 06. Developed a Global Registration algorithm for use in aligning frames of a video sequence in real-time.  **Developer,** Property Disposition Department, Northrop Grumman Space Technologies, October 2005 – March 2007. Designed and implemented a Web-based solution to facilitate Northrop Grumman Space Technology’s property disposition sales. Developed a streamlined process for extracting property data from Northrop Grumman’s database into a public-facing Web Application for consumption by prospective buyers.  **Developer**, PYRE, California Institute of Technology. Summer 2004 – 2005. Integrating PYRE, a framework for efficiently managing massively parallel scientific simulations, into the Computational Materials Design Facility (CMDF) to allow for efficient and effective distributed computing within the CMDF.  **Engineer**, Extreme Performance Computational Chemistry on Minimal Cost Graphical Processing Units, California Institute of Technology. Summer 2004 - 2005. Implementing quantum monte carlo (QMC) equations on a graphical processing unit (GPU) to obtain an 8-11x performance gain over central processing units (CPUs). Engineering a QMC/GPU supercomputer consisting of 100 GPU/CPU units generating a 10 Tera FLOPS peak performance at a price less than $200K. |

**Professional Memberships**

|  |
| --- |
| Sigma Alpha Epsilon Fraternity  Society for Advancement of Chicanos and Native Americans in Science  Circle K International |

**Posters and Publications**

|  |
| --- |
| A.F. Cervantes, M.J. Buehler and W.A. Goddard, III. "The Computational Materials Design Facility (CMDF) and Parallelization with Pyre: A new Multi-scale Multi-Paradigm Simulation Framework”, Computational Nanotechnology and Molecular Engineering Workshop, Cali, Colombia, <http://atlas.puj.edu.co/PASI/index.php>. February 15–18, 2005.  A.F. Cervantes, M. Sainz. “Multi-Modal Image Registration and Segmentation of the Visible Human Dataset”. Computational Nanotechnology and Molecular Engineering Workshop, Cali. Colombia. <http://atlas.puj.edu.co/PASI/index.php>. February 15-18, 2005.  J. Dodson, A.F. Cervantes, M.J. Buehler, W.A. Goddard III. “Development of a New Central Data Structure for the Computational Materials Design Facility Based on OpenBabel”, Annual Materials and Process Simulation Center Research Conference, California Institute of Technology, Pasadena, CA. March 17-18, 2005. |

**Honors & Fellowships**

|  |
| --- |
| CSU-LSAMP Fellowship, Cal State L.A. and National Science Foundation. 2004-2006. CAMP Researcher of the Year, California Alliance for Minority Participation, University of California, Irvine. 2004.  Extreme Computing Finalist, School of Information and Computer Sciences, University of California, Irvine. 2004.  Special Merit Award in Physical Sciences and Engineering, University of California and the National Science Foundation. 2004. |

**Presentations at Professional Meetings**

|  |
| --- |
| CAMP Statewide Symposium, Irvine, California. 2004. |

**EDMUND GEAN**

**egean@calstatela.edu**

**EDUCATION**

University of Southern California, L.A., California.

MS Computer Science, Dec., 1991.

University of California Berkeley, Berkeley, California.

BS Electrical Engineering & Computer Science, Dec., 1985.

**ACADEMIC EXPERIENCE**

Part-time lecturer in Department of Computer Science since 1998.

**NON-ACADEMIC EXPERIENCE**

**2002-present. CALIFORNIA STATE UNIVERSITY, LOS ANGELES**

**Network Security Administrator**

* assisted in configuration and troubleshooting of Cisco Catalyst switches such as Cisco Catalyst 6509, 5509, 4506, 3750, 3500, and 2950.
* assisted in configuration and troubleshooting of Cisco routers such as Cisco 7204, 4700, 2801, 2610, 1760 and 2500
* assisted in installation and configuration of Juniper Netscreen firewall, Cisco PIX firewalls, and Cisco FWSM consistent with campus security policy
* designed redundant WAN network with high availability using routers running BGP and HSRP and using redundant firewall for stateful failover.
* upgraded remote access server to Cisco Access Server 5350 to support remote dial-in users with AAA authentication via TACACS+ and accounting via RADIUS
* troubleshooted problems related to Cisco VPN, Cisco URT, Cisco PIX, Cisco ACS
* configured and installed Packeteer Packetshaper bandwidth manager
* assisted in migration of campus backbone from FDDI to Gigabit Ethernet
* troubleshooted Ethernet and frame relay installations
* troubleshooted TCP/IP routing protocols such as RIP and OSPF.
* Assist in providing support for Aruba controllers and Aruba 802.11 wireless access points with 802.1x authentication
* Monitored network using Aruba Airwave, Statseeker, and NetQos
* conducted security assessment, network vulnerability and penetration tests
* installed network IDS Snort
* responded to incidents related to network or computer security

**11/93-2002. CALIFORNIA STATE UNIVERSITY, LOS ANGELES**

**Network Administrator and Unix/NT system administrator.**

* configured multiprotocol routers connected via FDDI backbone supporting over 4000 nodes via 40 Ethernet and Token-Ring
* subnets running TCP/IP and AppleTalk. Troubleshoot Larscom CSU/DSU units connected via T1 line
* monitored network with aid of Sun Net Manager, Cabletron Spectrum, and network analyzer.
* assisted in implementation of network security using router access control lists (ACL) and Checkpoint firewall.
* installed and administered various remote access servers (Cabletron Cyberswitch 7000, Xylogics/BayNetworks Remote Access 4000, Sun Terminal Server) supporting SLIP and PPP connections for remote dial-up users. Installed authentication servers such as RADIUS.
* wrote scripts using Perl, awk, C shell, Bourne shell, and Expect to automatically monitor the health (eg. packet collision rate, network bandwidth utilization, server
* load, and disk usage) of network resources and to provide reports via WWW web pages for management to aid in capacity planning.
* maintained Sun Enterprise servers and Sparc workstations running Solaris 2.x. Assisted in administration of Domain Name Service(DNS), Network Information Service(NIS), Network File System(NFS), and DHCP.
* installed and configured network printers using HP JetAdmin software.
* used Legato's Budtool for automated tape backup of of Unix file systems. Used Veritas Backup Exec to backup NT servers.
* Installed Windows NT server 4.0 as backup domain controller on Dell PowerEdge4300 system with hardware RAID 5. Installed Microsoft Exchange and created mailboxes for 800 users.
* made recommendations on the reconfiguration of existing network architecture and design of new networks to accommodate growth.
* troubleshooted connectivity problems with networked PCs running PC-NFS protocol and Macintoshes running Mac/TCP.
* supervised work of network technician staff. Scheduled tasks and resources to meet project goals and objectives. Attended staff meetings to review work in progress.
* setup network bandwidth monitoring software to aid in capacity planning.
* installed network bandwidth monitoring software(MRTG) to aid in capacity planning; network traffic statistics collected via SNMP were displayed in graphical form on web page in real-time.
* tracked down denial-of-service attacks such as ping floods and IP spoofing.

**10/91-11/93. COMPUTER SCIENCES CORP., El Segundo, CA.; Systems Engineer.**

**5/91-10/91. OPTICOMP, INC. Los Alamitos, CA.; Network Engineer.**

**7/89-8/90. THERMOMETRICS, INC. Edison, NJ; Software Test Engineer.**

**CERTIFICATIONS**

Cisco Certified Network Associated (CCNA)

Microsoft Certified Professional Engineer

# Hajianpour, Mahan

**mhajianpour@gmail.com**

1. **Education**
   1. Master of Science, Computer Science, University of Southern California, 2007
   2. Bachelor of Science, Computer Science, California State University, LA, 2005
2. **Academic experience**
   1. California State University, LA, Adjunct, Lecturer, 2007-Present, Part-time
   2. The Aerospace Corporation (Institute), Lecturer, 2010-2011
3. **Non-academic experience**
   1. The Aerospace Corporation, Member of Technical Staff Sr., Software Engineer, Designer, Developer, 2005-Present, Full-time
   2. Glendale Unified School District, Technical Staff, IT Technician/Administrator, 2001-2005, Part-time
4. **Certifications or professional registrations**
   1. None
5. **Current membership in professional organizations** 
   1. Corporate Member of AFCEA International
6. **Honors and awards**
   1. Many internal Aerospace corporate awards related to space systems and defense programs
7. **Service activities** 
   1. None
8. **Most important publications and presentations from the past five years – title, co-authors if any, where published and/or presented, date of publication or presentation**
   1. Patent application: System and Methods for End-to-End Location and Media Content Tracking
      1. Publication date: 06/30/2011
      2. Patent application number: 20110161005
      3. Patent Pending
   2. IEEE Conference paper: STARS (Spacelift Telemetry Acquisition and Reporting System) Limit Checking System
      1. Publication date: 07/24/2006
      2. Print ISBN: 0-7803-9545-X
      3. Published and presented
9. **Most recent professional development activities**
   1. Task related to software development, engineering and acquisition.
      1. (Unfortunately, I can’t name programs from Aerospace)

**John W. Hurley**

**hurley\_j@sbcglobal.net**

**Education**

MS in Computer Science expected, California State University at Los Angeles, 2012

MA in American History, Harvard University, 1991

BA with High Honors in English, University of California at Berkeley, 1987

**Academic Employment**

**Teaching Associate in Computer Science, California State University-Los Angeles** 10/2010-Present

Teach introductory courses in web development, databases, and Java programming

**Adjunct Instructor in American History, Various Community Colleges**

9/1993 -6/1998

* Adjunct Instructor in American History at Solano Community College, Pasadena City College, and City Colleges of Chicago-US Military Distance Learning program, 1993-1998

**Teaching Fellow in History, Harvard University, 1991-1998**

**Other Employment**

**Estate Planning Representative, Caliri Insurance Group** 10/2008-10/2010

* Consulted with business owners and high-income individuals on estate planning and retirement benefits
* Designed life insurance and retirement plans

**Director of Advanced Sales, Innovative Solutions Insurance Services** 5/2003-10/2008

* Promoted financial planning strategies using life insurance to sales force
* Educated sales representatives in technical aspects of life insurance, financial planning, and taxation

**Internal Wholesaler, Transamerica Insurance and Investments Group** 8/2000-4/2003

* Promoted Transamerica products to independent insurance agents

**JOSE M. MACIAS**

**jose.m.macias@jpl.nasa.gov**

**EDUCATION**

* Ph.D., Applied Mathematics, CLAREMONT GRADUATE SCHOOL; Claremont, California, 1998
* M.A., Applied Mathematics, CLAREMONT GRADUATE SCHOOL; Claremont, California, 1993
* M.S., Computer Science, UNIVERSITY OF SOUTHERN CALIFORNIA (USC), Los Angeles, California, 1984
* M.B.A, Major: Finance, CALIFORNIA STATE UNIVERSITY, LOS ANGELES; Los Angeles, California, 1980
* M.S., Electrical Engineering, UNIVERSITY OF CHILE; Santiago, Chile, 1971

**ACADEMIC EXPERIENCE**

1999-Present UNIVERSITY OF SOUTHERN CALIFORNIA; Los Angeles, California

**Computer Science and Mathematics Instructor**

**.** Advanced Compiler Design (csci565), Advanced Database Design(csci585), Automata Theory (csci301), Advanced Theory of Algorithms (csci570), Analysis and Design of Algorithms (csci303) and Discrete Mathematics (csci271).

1981-Present CALIFORNIA STATE UNIVERSITY, LOS ANGELES; Los Angeles, California, Department of Mathematics and Computer Science

**Computer Science and Mathematics Instructor**

**.** Software Engineering, Database Analysis and Design, Data Structures, Programming Languages (C, C++, Assembly, Fortran, Pascal, Lisp), Object-oriented Programming, Operating Systems, Advanced Operating Systems, Computer Networks, Computer Architecture, Advanced Computer Architecture, Numerical Calculus, Software Engineering, Automata Theory, Theory of Algorithms, Logic Programming using Prolog.

**AWARDS**

NASA Medal for Exceptional Achievement as a Mission Assurance Manager for Epoxi and Stardust-Next

Many Team Awards.

**EXPERIENCE**

1978-Present JET PROPULSION LABORATORY; Pasadena, California

8/04-Present **Mission Operations Assurance Manager** (MOAM). MOAM for various NASA projects including Spitzer, DI-Epoxi, STD-NexT, Rosetta, Grace, Voyager. Past Projects included: Mars Odyssey (MO1), Gravity Recovery and Climate Experiment - Grace, Active Cavity Radiometer Irradiance Monitor - Acrimsat, Rosetta, Tropospheric Emission Spectrometer – TES, Atmospheric Infrared Sounder – AIRS, Microwave Limb Sounder – MLS, Voyager, Ulysses. Special Accomplishments include:

Spitzer: In record time cleaned up all Crit 2 ISAs, redid the Top Risk list for the project and significantly simplified the overall work for the team reagrding risk and ISAs.

Epoxi & NexT: I was instrumental to management and the teams at making these two mission a fantastic JPL/NASA success.

Rosetta: Instrument Alice slit installed in a non-optimal position. Dealt with the issue visiting Germany dynamics experts & discussing with Alice Mission Mgr and PI. Presented results to NASA & JPL. Issue closed to all parties satisfaction.

Rosetta: Rosetta’s instrument MIRO had had two major software failures and the hope to find their cause was remote. Presented to JPL and NASA the extreme need of funds to complete the MIRO test bed. Funds were obtained from NASA and the problem was finally determined.

Grace: For several years the project had been trying to identify a candidate with AACS, Systems Engineering, Grace familiarity and Software skills to lead the development of the on-board protection software. Visited Germany and Palo Alto experts and led a team with members in all three institutions (2 locations in Germany, 1 in Palo Alto and 1 in JPL). The software was delivered in about 14 months almost a month before the spacecraft failed and required its use.

Voyager: The project was in a difficult risk situation with mostly unknown risk status on both spacecraft. ISAs open for many years (more than 8 years in average) had been largely ignored. Took action leading the team at closing the ISAs and in less than 2 weeks all except two of the most pressing ISAs were closed. In the remaining 2 months all ISAs have been closed and the risk status of the project has been well established and publicized. A document was written (reviewed by the 5X Director) to report NASA on the Voyager risk status.

Acrimsat: Helped the project to reduce costs by alleviating payments for unused services. This allowed the project to survive for almost an additional year before getting additional funds.

Proposal: Mars Volcanic and Emission Life (MARVEL). Coordinate with Project manager and Mission Manager the safest procedures and activities to satisfy the Office of Safety and Mission Success requirements for projects in Operations (phase E).

04/04-08/04 **System Engineer** for JIMO’s Level 3 Space System Requirements. Assessment and detailed analysis of Spacecraft level 3 requirements for JIMO Space System. Essential contribution to generate one of the most important Requirements documents in the spacecraft area.

5/03-04/04 **Verification & Validation Engineer** for Mars Reconnaissance Orbiter Requirements. Coordinated low level testing requirement with the Lockheed Martin contractor building and operating the spacecraft.

12/02-03/05 **Software Development Engineer** for Deep Impact Spacecraft. Translation and implementation to JPL’s AMMOS system of Telemetry pages described in Ball’s OASIS format.

# Kotcherlakota V. Bapa Rao, Ph.D.

**bapa.rao@gmail.com**

### *Education*

### *Ph.D. in Computer Science (Database Systems), University of Southern California, 1987*.

***Master’s*** *in Electrical Engineering*. ***Rice University***, 1980.

***Bachelor’s*** *in Electrical Engineering*. ***Indian Institute of Technology***, Madras, 1978

### *academic Experience*

### *California State University, Los Angeles. Computer Science Lecturer. 2002-present. Part-time.*

### *University of Southern California. Computer Science Lecturer, 1985-2002. Part-time.*

### *Non-academic Experience*

AT&T Interactive. Senior Member of Technical Staff. Software Development and Research. 2007-present. Full-time.

Machinima.com. Director of Technology. Software and Systems Management and Development. 2006-2007. Full-time.

Independent Consultant. Software Development and Research. Web development and software outsourcing 2002-2006. Full time.

E-Connections, Inc. Director of Technology. Technology Evaluation. 2001. Full-time.

Enterworks, Inc. Product Architect. Data Warehouse product implementation. 2000. Full-time.

Network Associates Labs. Senior Computer Scientist. Network Security Research. 1997-2000. Full-time.

Platinum Technologies, Inc. Consulting Member of Technical Staff. Security and System Management Software Development. 1995-1997. Full-time.

Advanced Computing Systems Company. Senior Member of Technical Staff. Database Systems Research and Software Development. 1990-1995. Full-time.

University of Southern California Advanced Computing Support Center. Research Scientist. Database Systems Research and Software Development. 1987-1990. Full-time.

*CERTIFICATIONS*

Cloudera, Inc. Certified Hadoop Professional. 2010.

*Current membership in professional organizations*

Institute of Electrical and Electronics Engineers.

*RECENT PRESENTATIONS OF INTEREST*

Location and Relevance. Proceedings of LocWeb '09. Second International Conference on Location and the Web. In Conjunction with CHI 2009. Boston, MA. April 2009. (With P. Ehlen and R. Zajac)

# John J. Tran

**johnjtran@gmail.com**

1. **Education**

BS Computer Engineering, University of Notre Dame, 1996

MS Computer Science, University of Notre Dame, 1998

1. **Academic experience**

Lecturer, California State University Los Angeles, 2007 – Present

Lecturer, San Jose State University, 1999 – 2001

Lecturer, Santa Clara University, 1998 – 2002

1. **Non-academic experience**

Aerospace Corporation, Member of Technical Staff, 2011 – Present

The Jet Propulsion Laboratory, Software Engineer 2008 – 2011

University of Southern California, Software Analyst, 2002 - 2008

Stanford Linear Accelerator, Computer Research, 2001 – 2002

Safetopia, CTO/VP Engineering, 1999 – 2001

1. **Certifications or professional registrations**

ACM, IEEE

1. **Honors and awards**

Notre Dame Scholar, Intel Product On-time Award, Air Force Meritorious Service Medal, Air Force Commendation Medal, Air Force Achievement Medal, I/ITSEC Graduate Scholarship.

1. **Service activities**

Teaching and volunteering in local community on technology related consulting

1. **Publications**

J. Tran, J. Smith, W. Davies, V. Adame, D. Davis. Building a Technology Center for Iraqi Air Force Communications Training. Interservice/Industry Training, Simulation, and Education Conference. Orlando, Florida 2010.

D.M. Davis, R.F. Lucas, G. Wagenbreth, J.J. Tran, J. Agalso  
, T.D. Gottschalk. FLOPS per Watt: Heterogeneous-Computing's Approach to DoD Imperatives. Interservice/Industry Training, Simulation, and Education Conference . Orlando, Florida 2009.

R.F. Lucas, D.M. Davis, G. Wagenbreth, J.J. Tran. Operational Use of a Large GPGPU-Enhanced Linux Cluster. in the Proceedings of the High Performance Computing Modernization Program Users' Group Conference , San Diego, California, 2009.

A. Hart, J. Tran, D. Crighton, K. Anton, H. Kincaid, S. Kelly, J. Hughes, C. Mattmann. An Extensible Biomarker Curation Approach and Software Infrastructure for the Early Detection of Cancer. International Conference on Health Informatics . Porto, Portugal, January 2009.

C. Mattmann, J. Tran, H. Kincaid, D. Crichton, A. Hart, K. Anton, J. Dahlgren, M. Thornquist, D. Stelling, S. Reid, C. Patriotis, and S. Srivastava. The eCAS Model for Scientic Data Warehousing of Biomarker Data. In Proceedings of the 6th EDRN Scientic Workshop , August 2009. Bethesda MD.

D. Crichton, M. Thornquist, S. Kelly, A. Hart, H. Kincaid, C. Mattmann, J. Dahlgren, K. Anton, D. Stelling, G.Warnick, S. Reid, C. Edelstein, J. Tran, C. Patriotis, and S. Srivastava. Providing Integrated Access to Scientic Information and Knowledge in Cancer Biomarker Research. In Proceedings of the 6th EDRN Scientic Workshop , August 2009. Bethesda MD.

H. Kincaid, A. Hart, K. Anton, C. Patriotis, M. Thornquist, J. Dahlgren, C. Mattmann, D. Crichton and J. Tran. Curation of EDRN Cancer Biomarker Research. In Proceedings of the 6th EDRN Scientic Workshop , August 2009. Bethesda MD.

A. Hart, C. Mattmann, J. Tran, D. Crichton, H. Kincaid, J. S. Hughes, S. Kelly, K. Anton, D.Johnsey, C. Patriotis. Enabling Eective Curation of Cancer Biomarker Research Data. In Proceedings of the 22nd IEEE International Symposium on Computer-Based Medical Systems (CBMS), Albuquerque, NM, August 3rd-4th, 2009.

J. Tran, G. Wagenbreth, D. Davis, K-T Yao, D. Bakeman, and R.F. Lucas. A High Performance Route-Planning Technique for Dense Urban Simulations. Interservice/Industry Training, Simulation, and Education Conference . Orlando, Florida 2008.

D.M. Davis, R.F. Lucas, G. Wagenbreth, J.J. Tran, E ectively using a Large GPGPU-EnhancedLinux Cluster. In the Proceedings of the High Performance Computing Modernization Program Users' Group Conference , Seattle, WA, 2008.

J. Tran and K-T Yao. Harnessing Virtualzation Technology: A Novel Approach to Managing Distributed Computing Resources. DREN Networkers Conference . Boston, Massaschusett 2007.

R.L. Grossman, A. Anand, S. Connelly, Y. Gu, M Handley, M. Sabala, R. Sulo, D. Turkington, L. Wilkinson, I. Foster, T. Leggett, M. Papka, M. Wilde, J. Mambretti, R.F. Lucas, and J. Tran. Angle: Detecting Anomalies and Emergent Behavior from Distributed Data in Near Real Time. Supercomputing , Reno, Nevada 2007.

D.M. Davis, R.F. Lucas, G. Wagenbreth, J.J. Tran, and J.R. Moore. A GPU-Enhanced Cluster for Accelerated FMS. HPCMP User Group Conference . Pittsburgh, Pennsylvania 2007.

P. Colon, J.J. Tran, K-T Yao, M.D. Anhalt, and J.M. Curiel. Modeling Human Perception of Situation Awareness During Constructive Experimentation. Interservice/Industry Training, Simulation.

**Appendix C: Equipment**

**I. Current Resources**

Computing laboratories are the primary computing resources for the Computer Science program. There are many Computer and Smart Classrooms that are dedicated to the programs within the College. The Computer Science programs have the first use of many of the classrooms, especially those defined in ETA210, ETA220, ETA309, ETA310, and ETA255G (See Table C.1). These are closed classrooms with networked workstations, internet connection, and projection system that facilitate faculty-student interactions. Some of the classrooms are referred to as Mediasite classrooms which have the capability of streaming classroom presentations. (See Table C.2).

The university provides Open Access Labs (OAL) for use by the students. These labs are made available to assist students to accomplish their academic goal for instruction and research. OALs are equipped with hardware and software that serves all the disciplines within the university. The OALs are made available to students in all disciplines for University related work. Though all of the OALs provide identical services to the campus community, they offer slightly different equipment, hardware and software depending on various needs in each geographical area on campus. The OAL listed as ECST Link (<http://www.calstatela.edu/its/desktop/oal/>) is better equipped to serve the Computer Science discipline.

Computer classrooms are available during the class time while the university OALs is accessible for extended hours. ETA309 laboratory is dedicated for exclusive use by the senior students completing their capstone projects.

Computer Science program also uses a number of servers as described in Table C.3. Some of the courses are taught exclusively using these servers. All the servers are accessible on campus and also from off campus.

Computer stations described in the above labs are installed with a software image that serves all Computer Science courses. The faculty makes extensive use of high quality commercial and free software for instruction and scholarly activities that is described in Table C.4.

Since most of the software is open source, students are encouraged to develop the software image on their personal desktop/laptop that mimics the software image installed in the computer classrooms.

In addition, The Engineering, Computer Science and Technology Computer Productivity Center (CPC) supports the computing needs of engineering, computer science and technology programs in the College as described in Section 5 under Appendix D.

|  |  |
| --- | --- |
| Room | Description |
| ETA 210 | 30 student stations, 1 instructor station |
| ETA 220 | 30 student stations, 1 instructor station |
| ETA A309 | 30 student stations, 1 instructor station |
| ETA A310 | 8 stations, conferencing facilitiesTwo Microsoft XBOX 360’s with KINECT (Motion Sensor) to facilitate game programming classes1510.1" Samsung Galaxy Tab with Android 3.1 to facilitate mobile computing. |
| ETC 255DEG | 3 labs each with 24 students stations and 1 instructor station |
| ETB 9 | 24 students stations (PC + MAC) and 1 instructor station (PC + MAC); |
| ETC245 | 8  Cisco1600 & 4  Cisco 2500 series routers12 Cisco2900 or Cisco3500 series Ethernet switches 8 Cisco501 firewalls25 desktop computers |

**Table C.1: Computer Laboratory Classrooms**

| Room | Description |
| --- | --- |
| ETA 210 | Manual or preset recording |
| ETC 255E | Manual or preset recording |
| ETA 331 | Manual or preset recording |
| ETA 332 | Manual or preset recording |
| Portable Recorder | Available for recording presentations in room |

**Table C.2: Mediasite Classrooms**

| **Server** | **Description** |
| --- | --- |
| csns.calstatela.edu | Used by the students and faculty as a Learning Management System. |
| cs.calstatela.edu | Used by the Department for serving Web pages and other Department-related administrative purposes. |
| cs1.calstatela.edu | Redhat Linux server. Used for lower level web, Unix, and databases programming classes such as CS120, CS122, CS245, CS 345, CS 422. Services offered: Apache Web Server, MySQL database, Postgres Database, Linux Shell |
| cs2.calstatela.edu | Windows 2003 Server. Used for senior design projects that require Windows platform. Services offered: IIS Web Server, .NET Programming |
| cs3.calstatela.edu | Ubuntu Linux ServerUsed for higher level web, Unix, and databases programming classes such as CS320, CS520. Services offered: Apache Tomcat Server, MySQL database, Postgres Database, Linux Shell |
| csproject.calstatela.edu | Redhat Linux server. Used primarily for Senior Design projects as well as any other Special Projects that come up. Services Offered: Apache Web Server, Subversion, Database (Mysql, Postgres), Apache Tomcat Web Server |
| oscar.calstatela.edu | 10 node cluster (1 head node, 9 child nodes). Used primarily for courses devoted to Parallel Programming |
| acm.calstatela.edu | Used by the ACM Student Chapter Computer Club. Services offered: Web Server |
| Cal State L.A. UNIX Servers | Nine planet servers:mars.calstatela.edu, neptune.calstatela.edu,…. |
| Network Storage Server | A 10 TB backup storage server that supports faculty workstations and other CS servers. |
| Blade Server | 16 Blade Server for remote login for specialized applications in certain courses. |

**Table C.3: Servers**

| Software | Description |
| --- | --- |
| Integrated Development Environments (IDE) | Various IDEs (eclipse, JBuilder, Netbeans, Textpad, Notepad++) to support programming environments: Java, C++, Haskell, Prolog etc.All of the software is Open Source and students can set it up on their personal computers. |
| Adobe Creative Suite CS5 (Master Collection) | Adobe licensed software available on ETB9 PC Computers. |
| Unity 3D & Panda 3D Game Engine | Unity 3D is Licensed software available on ETB9 PC ComputersPanda 3D game engine is open source software. |
| Adobe Creative Suite CS4 (Design Premium) | Licensed software available on ETB9 MAC Computers. |
| Microsoft | The Microsoft Campus Agreement provides all the needed software (Operating System, Office, Visual Studio, Visio, Project, etc.,) for every PC computer on campus.The MSDN Academic Alliance allows all Computer Science majoring students to get free software distribution of all Microsoft software other than Microsoft Office.MSDNA Alliance is a renewable annual license. |
| MATLAB, MATHCAD | Licensed software titles are available via network license and can be installed to any ECST Computer Lab. This is available for remote access on the Blade Server. |
| OPNET | OPNET modeler is a network simulation software package which allows students with little background to create their own network scenarios and run simulation at different network layers. In addition, OPNET modeler can be used to support both system level and node level design and simulation.60 OPNET licenses, which are renewable every 6 months. ( Flow Analysis Module, OPNET Modeler University Teaching Program, University AppTransaction Xpert Module,University Terrain Modeling Module, University Wireless Module)This is available for remote access on the Blade Server. |
| VMWare | Virtualization software for desktops and servers. Installed on ET C245 workstations and on some servers. |

**Table C.4: Computer Science Software**

**II. Five Year Maintenance Plan**

Program Computing Facilities described above includes computer classrooms, server platforms, and software. All computer classrooms are coved by the university baseline plan and refreshed accordingly. The rest of the hardware and software needs of the program are maintained by the College. It is estimated that the server platforms will be refreshed/upgraded on a three year cycle. All software will be renewed annually or bi-annually as per the licensing agreements. The ITC staff does an excellent job keeping track of software upgrades as per the licensing agreements.

Table C.5 identifies the approximate Maintenance Plan over the next five years.

| **Equipment** | **Estimated Amount** |
| --- | --- |
| Upgrade/Replace servers in Table C.3 = 16x5000 | $80,000 |
| Faculty baseline upgrade (Memory, Disk, Docking Station) = 9x1000 | $ 9,000 |
| Network lab E&T C245 lab upgrade routers (Cisco 2611XM with WIC-IT = 13\*$200), firewalls, wireless access points (=3\*150) workstations (25\*$1000), SSL VPN (Juniper SA 700 =1\*$500) | $ 28,550 |
| Senior Design E&T A310 upgrade = 10x3000 | $30,000 |
| Other Hardware upgrades (Tablets, Gaming devices) | $10,000 |
| Software licensing Microsoft MSDANAA  Microsoft SQL server for infrastructure use  Microsoft Windows server for infrastructure use  OPNET  MATLAB  Unity 3D  Adobe  Mediasite  VMWare  VPN for students  Total Software | $500  $600  $400  Annual (Free)  $6,200  $5,400  $12,500  $15,200  $5,000  Paid by Campus  $45,800 |
| Total | $203,350 |

**Table C.5: Five Year Maintenance Plan**

# Appendix D – Institutional Summary

Institutional summary is compiled and bound as a common document by the College of Engineering, Computer Science and Technology.

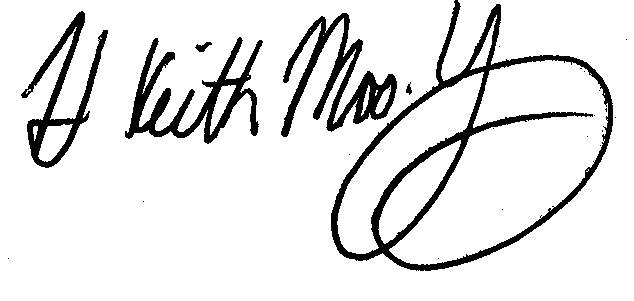
# Signature Attesting to Compliance

By signing below, I attest to the following:

That *Computer Science undergraduate program (CS BS)* has conducted an honest assessment of compliance and has provided a complete and accurate disclosure of timely information regarding compliance with ABET’s *Criteria for Accrediting Computing Programs* to include the General Criteria and any applicable Program Criteria, and the ABET *Accreditation Policy and Procedure Manual.*

Keith Moo-Young .

Dean, College of Engineering Computer Science, and Technology

****

**June 15th, 2012**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature Date**