**ABET**

**Self-Study Report**

**for the**

**Bachelor of Science Degree in**

****

**at**

**California State University, Los Angeles**

**Los Angeles, CA**

**July 1, 2018**

**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.

­­­

Table of Contents

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

A. Contact Information

B. Program History

C. Options

D. Program Delivery Modes

E. Program Locations

F. Public Disclosure

G. Deficiencies, Weaknesses or Concerns from Previous Evaluation

and the Actions taken to Address Them

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

A. Student Admissions

B. Evaluating Student Performance

C. Transfer Students and Transfer Courses

D. Advising and Career Guidance

E. Work in Lieu of Courses

F. Graduation Requirements

G. Transcripts of Recent Graduates

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

A. Mission Statement

B. Program Educational Objectives

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

D. Program Constituencies

E. Process for Revision of the Program Educational Objectives

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

A. Student Outcomes

B. Relationship of Student Outcomes to Program Educational Objectives

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

D. Enabled Student Characteristics

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

A. Student Outcomes

A-1. Assessment measures on CSNS

A-2. Assessment data collection on CSNS

A-3. Analysis of Student Learning Outcomes

A-4. Relationship between Student Learning Outcomes and ABET criteria

B. Continuous Improvement

C. Additional Information

C-1. Constituent Comments

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

A. Program Curriculum

A-1. Course requirements

A-2. Course prerequisite(s) requirements

A-3. Prerequisite flowchart

A-4. Capstone courses

A-5. Course coverage of required areas

A-6. Relationship between courses and Student Learning Outcomes

B. Course Syllabi and Course Journals

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

A. Faculty Qualifications

B. Faculty Workload

C. Faculty Size

D. Professional Development

E.. Authority and Responsibility of Faculty

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

A. Offices, Classrooms and Laboratories

B. Computing Resources

C. Guidance

D. Maintanance and Upgrading of Facilities

E. Library Services

F. Overall Comments on Facilities

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

A. Leadership

B. Program Budget and Financial Support

C. Staffing

D. Faculty Hiring and Retention

E. Support of Faculty Professional Development

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

[**APPENDICES**](#AppendixA)

[Appendix A – Course Syllabi](#AppendixA)

[Appendix B – Faculty Vitae](#AppendixB)

[Appendix C – Equipment](#AppendixC)

[Appendix D – Institutional Summary](#AppendixD)

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

# 

# 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**

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)

* **Jane Dong**

Associate Dean & ABET coordinator

College of Engineering, Computer Science and Technology

5151 State University Drive

Los Angeles, CA 90032

Phone: 323-343-4450

e-mail: [jdong2@calstatela.edu](mailto:jdong2@calstatela.edu)

* **Emily Allen**

Dean

College of Engineering, Computer Science and Technology

5151 State University Drive

Los Angeles, CA 90032

Phone: 323-343-4450

e-mail: [eallen3@calstatela.edu](mailto:eallen3@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 in Computer Science degree program was started in 1980’s and 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 2001-2017.

* 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 program was reaccredited in 2012.
* Many major revisions to the undergraduate curriculum (CS BS) were implemented over the years from 2001 to 2015 before the most recent curricular changes in 2016.
* California State University, Los Angeles converted from a Quarter system of three quarters per academic year (10 weeks of educational instruction per term), to a semester system of two semesters per academic year (15 weeks of educational instruction per term) beginning Fall semester 2016. This conversion provided an excellent opportunity for us to re-examine and strengthen our curriculum. Many significant curricular revisions were adopted. The curriculum now strictly conforms to the current ABET criteria.

The history of activities that have taken place in the Department since the last ABET accreditation visit in Fall 2012 are summarized below in Table 0.1. (See Section B under Criterion 4 for detailed descriptions)

| **Year** | **Comments** |
| --- | --- |
| 2012 | Added the Calculus based Physics sequence as required courses. |
| 2012 – 2013 | Implemented rubric evaluation in CSNS (Computer Science Network Services) to facilitate assessment processes. |
| 2014 | Increased number of units for CS301 to allow for a broader coverage societal, legal and security issues in computing. |
| 2015-2016 | The university’s conversion from Quarters to Semesters allowed us a chance to re-examine and strengthen our program. (i) Many quarter courses are converted to semester courses with an increased time coverage (ii) Many new courses were created to strengthen the program. (iii) All changes that effected the ABET Criteria were documented. (See Assessment Report 2016) |
| 2016 | Created a new CS2148 course in Discrete Structures. |
| 2016 | Increased our programming sequence to three semesters: CS2011-CS2012-CS2013. |
| 2016 | Increased the number of units and strengthened the senior design sequence: CS4961-CS4962. |
| 2016 | Increased the number of units and strengthened the capstone recapitulation course: CS4963. |
| 2016 | Added ENGL2030 as a technical writing course. |
| 2016 | Increased the number of units and strengthened the computer ethics course: CS3801. |
| 2016 | Increased the number of units in the freshmen introductory course (CS1010) to provide an introduction to the computing disciplines. |
| 2016 | All lower division courses were converted to a project based learning format. All of these courses now have a built-in laboratory requirement. |
| 2016 | General Education (GE) requirements have been modified by integrating upper division Natural Science, Humanities, and Social Science GE outcomes into the Computer Science curriculum. |
| 2016 | Increased the number of required electives. The number of elective courses has been expanded to enable students to tailor their program to their individual interests. |
| 2014-2016 | Enhanced the Computer Science Network Services (CSNS) so that it now has all the tools required to be an Assessment Management System. This complements its role as a Learning Management System. |

**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 may be found 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 use CSNS as a learning management system. 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 2012 visit included no deficiencies, weaknesses, or concerns.

**GENERAL CRITERIA**

# CRITERION 1. STUDENTS

1. **Student Admissions**

**A.1 Admission Criteria**

Requirements for admission to California State University, Los Angeles (Cal State LA) are publicized on the University Admission webpage (http://www.calstatela.edu/admissions). Typically students are admitted to Cal State LA in two categories: first-time freshman (FTF) or transfer students, and the admission requirements are summarized below.

First time freshman (FTF) applicants are either graduating high school students, or high school graduates who have not earned college credit beyond the summer immediately following high school graduation. These students need to complete a set of college-preparatory coursework (A-G) with a grade of C or better prior to admission. The A-G curriculum includes four years of English, three years of mathematics, two years each of history and social science, lab science (including one biological science and one physical science), and foreign language, and one year each of arts and college preparatory electives. Across Cal State Universities, “eligibility index” is used to determine an applicant’s eligibility for admission. The index is basically a combination of the applicant’s high school grade point average and his/her highest ACT composite or SAT test score and is calculated as follows:

1) multiply the applicant’s high school grade point average by 800 and add the total SAT score;

2) or multiply the applicant’s high school grade point average by 200 and add ten times the total ACT score.

Students with good performances in honors courses get bonus points toward their index.

For California residents, the admission requirement for FTF is an index of 2900 (SAT) or 694 (ACT). For nonresidents, the requirement is an index of 3502 (SAT) or 842 (ACT). (The minimum Eligibility Index requirement for admission to the Fall 2018 term will increase to 2950 using the SAT for California residents, and to 3570 to nonresidents) California residents with a high school grade point average of more than 3.00 are not required to submit test scores, and the same rule applies to out-of-state applicants with a high school grade point average of 3.61 or more. However, Cal State L.A. urges all students, regardless of grade point average, to take one of the two exams.

Transfer applicants are students who have attempted college units beyond summer immediately after high school graduation. To be considered for admission, the applicants need to complete 60 or more semester transferrable units and meet the following criteria:

* + Earn grades of “C” or better in college-level English Composition, Oral Communication, Critical Thinking, and college-level Mathematics courses:
  + Have an overall minimum transferrable college grade point average of at least 2.00;
  + Be in good standing and eligible to re-enroll at your last college or university attended;
  + Satisfy the major-specific criteria requirements for the Computer Science major (listed below)

1. Required minimum cumulative GPA: 2.0
2. Required major preparation courses (grade of “C” or better required):
   * MATH 2110 – Calculus I
   * CS 2011 – Introduction to Programming I
3. Required General Education courses not covered by major preparation courses listed above:
   * Written Communication
   * Oral Communication
4. Additional recommended preparation for transfer students (grade of “C” or better required)
   * MATH 2120 – Calculus II (required beginning Fall 2019)
   * PHYS 2100 – General Physics I: Mechanics and Thermodynamics (required beginning Fall 2019)
   * CS 2012 – Introduction to Programming II
   * MATH 2550 – Introduction to Linear Algebra
   * PHYS 2200 – Electromagnetics and Optics

**A.2 Admission Process**

The Cal State LA Office of Admissions and Outreach oversees the admission process. Each year, the Office of Admission will publicize available terms for new student admission as well as the associated deadlines. Prospective applicants should submit their application before deadline through Cal State Apply (<http://www.calstatela.edu/admissions/apply>), a portal to online application and evaluation system. After the online application is submitted, the applicant will receive an Acknowledge email which includes the next steps and a preliminary campus ID (CIN). The applicants need to follow the steps to submit requested documents including transcripts and test scores by a term’s established deadlines, and they can use their campus ID to access the Golden Eagle Territory (GET) portal to track the progress of admission application. Evaluation of the applications will be performed based on established admission criteria stated in the previous section.

1. **Evaluating Student Performance**

Students are continuously evaluated from the time they apply for admission 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 is both integrated throughout the curriculum and decentralized by virtue of the fact that each professor chooses his or her own way 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**

Students are monitored as they progress through the Computer Science program. In addition to the mentoring and advising practices (described in Section D below), students are flagged automatically for a variety of reasons and monitored as described below:

1. *Monitoring for Administrative-Academic Probation*

The Office of the Chancellor has made provision whereby students may be placed on administrative-academic probation for any of the following reasons:

* Withdrawal from all or a substantial portion of courses for which they registered in two successive semesters or in any three semesters.
* Repeated failure to progress toward a stated degree or program objective when such failure is within their control.
* Failure to comply, after due notice, with a routine academic requirement or regulation.

1. *Monitoring for Academic Probation*

Students are monitored based on the GPA as indicated in <http://ecatalog.calstatela.edu/>, Procedures and Regulations). Students are placed on academic probation at the end of a semester if either their grade point average at Cal State L.A. or their cumulative grade point average in all college work attempted falls below C (2.0).  They remain on academic probation until their Cal State L.A. and cumulative grade point average is 2.0 or higher or until they are disqualified in accordance with the regulations for academic disqualification.

1. *Monitoring for Academic Disqualification*

Students already on probation or special probation whose Cal State L.A. or cumulative grade point average reaches the following levels are disqualified:

|  |  |
| --- | --- |
| ***Class Level*** | ***Grade Point Average*** |
| Freshman (0-29 units completed) | <1.50 |
| Sophomores (30-59 units completed) | <1.70 |
| Juniors (60-89 units completed) | <1.85 |
| Seniors (90+ units completed) | <1.95 |

1. *Monitoring for Special Probation*

Disqualified students are required to arrange a disqualification interview with their academic advisor to review the reasons for disqualification. These students could be reinstated and flagged as a special probation status. The advisor must specify on the special probation petition the terms and conditions under which the student will be eligible to be returned to and to remain in matriculated status, such as courses to be repeated, courses to be completed, GPA requirements, and any required workshops and/or tutoring.

All disqualified students who are placed on special probation are required to earn better than a C (2.0) grade point average each semester until their grade point average is increased to a level that is higher than that which would normally cause them to be disqualified according to their class level, at which time they would be switched from special probation to probation.

1. *Monitoring for Probation & Disqualification at the College level*

All ECST students who have been placed on academic probation must attend an Academic Development Workshop (ADW) offered by the ECST Student Success Center (ESSC).  ESSC advisors have been working closely with all academic departments to better meet the needs of students demonstrating academic deficiencies. The purpose of the Academic Development Workshop is to:

* Facilitate students’ understanding as to what factors may have contributed to their academic deficiency.
* Help students develop a personalized Action Plan.
* Provide students with helpful information about university policies and procedures, understanding unsatisfactory academic performance, along with tips and strategies on how to maintain good academic standing.
* Provide students with information about campus resources and the types of services available.

1. *Monitoring for Progress in the Major at the College level*

The College is currently engaged in discussions to draft a new policy whereby the students are monitored for progress in the major. This policy provides the conditions necessary for a student in the College of ECST to be considered in good standing in the major. This monitoring can be done by instituting success markers at various levels. This policy is currently being defined and may be approved by the 2018-2019 academic year. Even though the policy is not strictly enforced, the advisors are trying to monitor the system with the new tools that are being created at the university level.

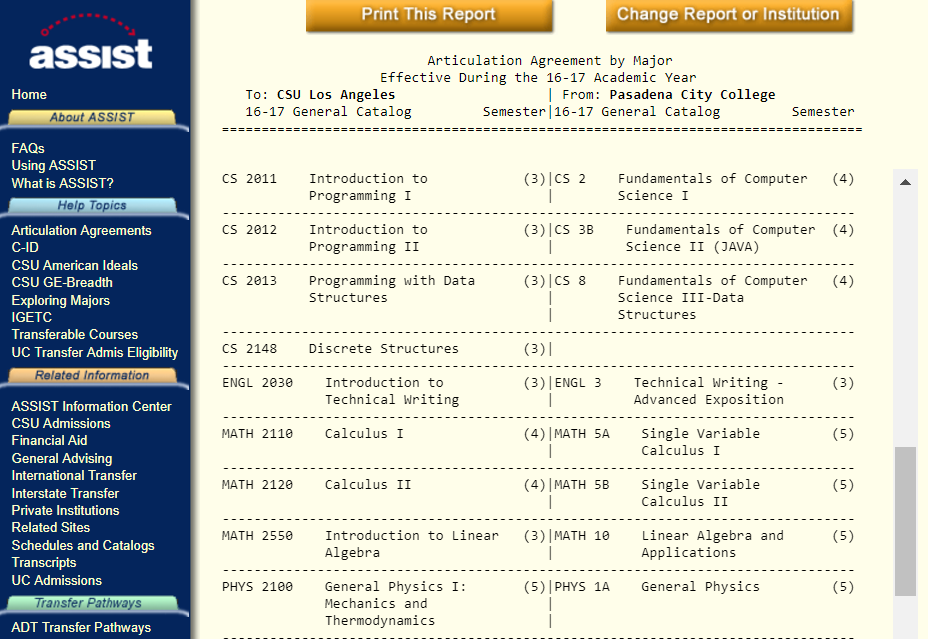
1. **Transfer Students and Transfer Courses**

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 both (a) general education and graduation course requirements met by transfer courses and (b) 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, which shows how course credits earned at one public California college or university can be applied when transferring to another. ASSIST, the official repository of articulation for California’s public colleges and universities, provides the most accurate and up-to-date information about student transfer in California.

A snapshot of articulated courses between Pasadena City 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, 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 confirm the course articulation, which would then be reflected on GET.



**Figure 1.1: Articulation Agreements from PCC**

## Advising and Career Guidance

All students receive academic advisement to help them make informed academic choices. Computer Science majors can seek advisement at both the department and the college levels.

## D.1 Department Undergraduate Advisement

The advisor and student go over the student’s degree progress data available on CSNS and GET. The advisor documents the discussion on CSNS, which is then accessible to the student and advising staff. Student advising in the department is considered either Open Advisement or Mandatory Advisement.

Open Advisement:

1. Advisor/staff office visitation: Office hours for the Advisor and the Department chair are posted in the Department Office. Students meet with their 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. Students may also seek the advice of any faculty member in evaluating career choices.
2. Email advisor: Students may seek advisement via email.
3. CSNS Advisement Forum: Students may post questions to a Forum hosted by CSNS. Responses are provided by other students or an advisor and are monitored by the Department staff and advisor.

Mandatory Advisement:

Every student is required to meet with their advisor at least once a year. At these meetings students plan their course schedule for the upcoming semester and formulate a longer-term road map to complete all remaining requirements. Advisement meetings are scheduled first with entering first time freshmen or transfer students and then with continuing students.

1. Entering first term students: 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 the advising process. Students are given information about the program requirements, which are also 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.
2. Freshmen level in CS1010: Entering freshmen or a first year transfer students must enroll in CS1010 during their first term. CS1010 presents a comprehensive overview of higher education. Topics include: University rules and regulations; general education requirements; major requirements; evaluation of transfer units; sample road maps; individualized quarterly planners; and graduation checks. Instructional videos from CS1010 remain accessible online.
3. Sophomore level in CS2011: CS2011 is a required course usually taken as the first required course at the sophomore level. Students are advised to keep pace with the Math and Physics requirements along with CS requirements. Students’ Individualized Roadmaps are checked and approved if satisfactory. Otherwise students are given assistance in modifying their roadmaps.
4. Junior level in CS3112: CS3112 is usually taken as the first required course at the junior level. Students roadmaps are checked to see if they are on pace to take the senor design next year. Students also make adjustments to their planned electives.
5. Senior level in CS4961. CS4961 is the front end of the senior design sequence. Students do a graduation check with the advisor to ensure that they are on track to graduate by the end of the year.

## D.2 College Undergraduate Advisement

The College of ECST Student Academic Support Services 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.  Advisement is mandatory every quarter for the first two years.

* Advance appointment guarantee advisement time with a Staff Advisor or Peer Advisor. Appointment may be made either by
  + Visiting the advisement office (Engineering and Technology building, A-127),
  + Calling the office front desk at 323-343-4574, or
  + Visiting <http://web.calstatela.edu/academic/ecst/student_services/>.

The advisement office also provides assistance with scholarship and internship opportunities.

Information regarding career choices is offered at 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 by talks by guest speakers (organized by the ACM student chapter and ECST).

## D.3 Transitional advising during Quarter to Semester (Q2S) Conversion:

California State University, Los Angeles converted from a Quarter system of three quarters per academic year (10 weeks of educational instruction per term), to a semester system of two semesters per academic year (15 weeks of educational instruction per term) beginning with the Fall semester 2016. This section pertains to students who began their academic programs before Fall 2016.

## D.3.1 Infrastructure to Support Q2S Advising

To oversee the Q2S conversion process, the University established an Office of Semester Conversion (OSC). OSC works with Colleges and University Curricular and Advisement committees to provides support and guidelines for both curriculum conversion and transitioning advisement. OSC also implemented tools such as a Conversion Guide and a Degree Planner to facilitate curriculum mapping, scheduling, and advising during the Q2S process. The Degree Planner is available on GET (Golden Eagle Territory). It enables the development of individualized advisement plan (IAP). An IAP maps out the academic requirements term by term (ongoing Quarters or future Semesters) and serves as an individualized road map for timely graduation. OSC produced training materials (handouts, video tutorials, and workshops) to help faculty/staff advisors and students develop IAPs. Additional information and supporting tools are provided on the Q2S website: (<http://www.calstatela.edu/semesterconversion>).

At the college level, a Q2S Advisement Coordinator was appointed to work with the Associate Dean, department Chairs, and faculty/staff advisors to oversee Q2S transition. In the College of Engineering, Computer Science and Technology, an Advising Taskforce was formed in 2014 to proactively discuss students’ needs in the Q2S conversion process and to develop college level guidelines and strategies to implement Q2S transitioning advice. The members of Advising Taskforce include department chairs and principal faculty advisors, staff advisors from ECST Student Success Center, and the college Q2S Advisement Coordinator. During the two-year Q2S conversion, the Advising Taskforce was instrumental in establishing a collaborative advising model and an effective process for the development of student IAPs. IAP Clinics helped students develop and review their IAPs.

At the department level, the Chair (Dr. Raj Pamula) and designated faculty advisor (Dr. Elaine Kang) worked closely with the ECST Student Success Center and the Q2S Advisement Coordinator to organize CS-specific IAP clinics, which reached out to students and helped them with their IAPs. The following advisement tools were developed to help students understand the curriculum changes and to guide them in developing IAPs.

1. *Major Curriculum Conversion Map*: Shows the mapping between the quarter and semester courses.
2. *ECST GE Conversion Map*: Helps students understand the differences between GE requirements in the Quarter and Semester systems.
3. *Major Roads Maps (in quarter/semester)*: These reference road maps were developed by major advisors and served as crucial references for students in developing IAP;
4. *Projected Course Schedule in Semester*: Allowed students to layout their IAP based on the actual timeline of course offerings;
5. *Step-by-step Guide to Develop IAP*: Developed collaboratively by the major advisors and the Student Success Center, this Guide provides easy-to-follow steps for students to use the GET Degree Planner to develop their IAP.

Q2S transitioning advising process and tools are described at <http://www.calstatela.edu/ecst/cs/semester-conversion>.

## D.3.2. College and Departmental Q2S Advising

In general, the Q2S advising process consisted of the following steps:

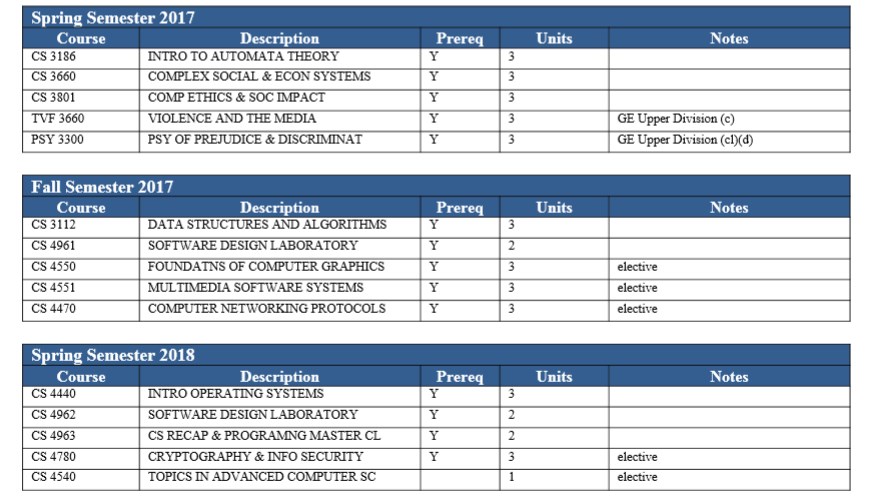
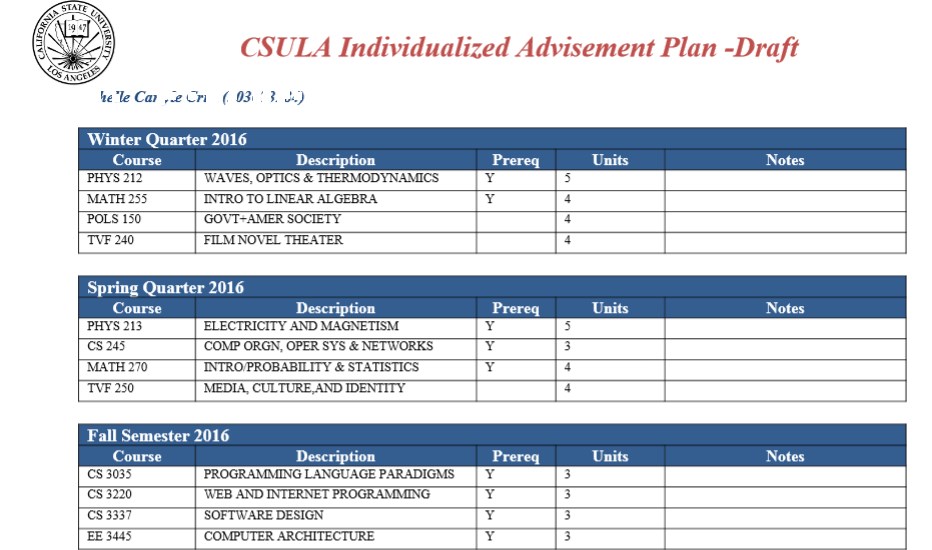
1. The Semester Conversion Office worked with the Registrar Office to identify students who would be affected by Q2S transition based on their degree progress;
2. The department inform the identified students by email (and telephone reminders) to register and attend one of the IAP clinics;
3. The department arranged multiple IAP clinics to conduct Q2S transiting advise in group settings.

Students who began their academic programs under the quarter system (i.e., prior to Fall 2016) were offered two options: (a) complete their requirements with semester course equivalents or (b) switch to the new semester program requirements. During the IAP Clinic, advisors helped students review their degree progress and helped them determine if they should stay with Quarter curriculum or switch to a Semester program. Since the IAP clinic played a critical role in Q2S transitioning advising, its flow is elaborated below.

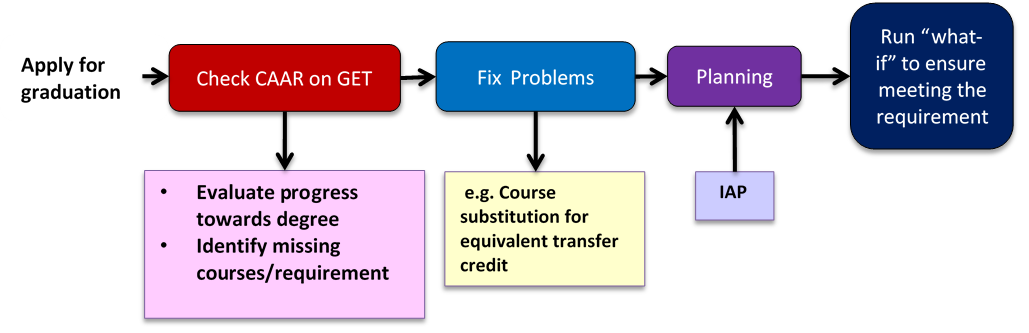
* An advisor (IAP clinic facilitator) explained the function of the IAPs.
* Students reviewed their degree progress with advisors and determined their remaining requirements.
* With guidance from advisors, students decided on the most efficient GE catalog year (quarter versus semester requirements).
* With the guidance from advisors, students selected the most efficient major catalog year (quarter versus semester requirements for the major).
* Using the Degree Planner, students selected generated an academic plan to meet all remaining graduation requirements.
* Both faculty and staff advisors were on site to answer students’ questions regarding both GE and major courses.
* Advisors helped students review their IAPs ensuring that the IAP meets the degree requirements under the selected catalog date.
* An example of a completed IAP is shown in Figure 1.2

Students were advised to check on the Academic Requirements Report on GET and plan the remaining requirements on an ongoing basis. Using a “What-if report” on GET, students and advisors could easily check whether graduation requirements were met under either the quarter or semester curriculum.

The graduation check process (as described in Section F below) remained unchanged. However, the availability of an IAP helped make this process easier for transition students. Figure 1.3 shows the updated graduation check process, which utilizes an IAP to simplify academic planning. It uses a “what-if” report to ensure that a student meets the graduation requirements given the student’s IAP.



**Figure 1.2 Sample IAP**



**Figure 1.3 Q2S advising and graduation process**

By Fall 2017, most of the remaining transition students had been switched over to the semester requirements, with full credit given to courses completed under the quarter system.

## D.3.3. Career Guidance

The College of ECST has a Professional Placement Coordinator who provides a link to the University Career Development Center. Each year, during National Engineers' Week, the College sponsors an industry career day. During this event, representatives from industry, government, and private practice set up information tables and meet with students throughout the day. Typically representatives from 20 to 30 companies are present at these events. This gives students an opportunity to meet prospective employers and to learn about their current engineering needs.

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 6 semester 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, 6 semester units of lower division elective credit are awarded for 1 year or longer of active duty with an honorable discharge, and 3 semester 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 9 semester units.

Students may 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 allowed only if the advisor is relatively certain that the student is competent in the material in introductory 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 described below:

1. Students take note of all the information (application deadlines, fees, diploma, commencement, transcripts, etc.,) as described at <http://www.calstatela.edu/sites/default/files/groups/Graduation/Docs/graduation_application_for_undergrad.pdf>.
2. Students complete the 2-page ***Degree Completion Worksheet*** and 1-page ***Undergraduate Graduation Application*.**

* Students review their Academic Requirements report on GET to complete the Degree Completion Worksheet.
* 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 1.4. Students thus have a general idea of the graduating quarter and discuss the program requirements with the advisor.



**Figure 1.4: Academic Requirements Report on GET**

1. Students make an appointment with the Advisor to discuss the Academic Requirements Report and to resolve/plan out all remaining requirements (which are indicated as RED flags). The Advisor reviews the *Degree Completion* Worksheet and the *Undergraduate Graduate Application*. Both applications are signed off by the Advisor.
2. Students submit the signed Graduation Application and Degree Completion Worksheet to the Cashier's Office (ADM 128) and pay the Application ($20) and Diploma ($10) fees. The Cashier's Office forwards the forms to the Graduation Office for processing and final audit.
3. Degree dates are posted at the end of the semester 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: | (date) |
| Plan: | Computer Science |

1. Student who donot graduate during the term declared on their **Graduation Application**, must file a **Request to Change Graduation Term.** Students must take the following steps before registering for classes beyond the previously declared graduation term.

* Make an appointment with the Advisor to determine the correct term at the end of which all graduation requirements will have been completed.
* Complete and sign the **Request to Change Graduation Term** form; available at [**h**](http://www.calstatela.edu/graduation)**ttp://**[**www.calstatela.edu/graduation**](http://www.calstatela.edu/graduation)
* Pay the $25 late filing fee to the Cashier’s Office

The Cashier's Office will then forward the form to the Graduation Office for processing.

1. After the degree is posted on GET, CSNS imports the list of all graduated students and updates them to alumni standing.
2. **Transcripts of Recent Graduates**

As requested by the Team Chair a random selection of transcripts will be provided to the visiting ABET team prior to or during the site visit. Academic Requirements Reports on GET for can be generated at the time of the visit for the selected students.

# CRITERION 2. PROGRAM EDUCATIONAL OBJECTIVES

1. **Mission Statement**

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

**The University's Mission**

The Mission Statement (also provided below) of California State University Los Angeles is available online at <http://www.calstatela.edu/csula-missionstatement.php>.

*California State University, Los Angeles has one of the most diverse student populations of any college or university in the nation. Building on the strengths of this rich diversity, our University prepares students for success in advanced studies, in their careers, and throughout their lives. California State University, Los Angeles graduates constitute a major leadership force in Greater Los Angeles, a microcosm of the global society.*

*The University is committed to free scholarly inquiry, to high-quality teaching, and to academic excellence in undergraduate, graduate, and other post-baccalaureate and extended education programs. This commitment underlies strong educational programs that are sensitive to the needs of the University's uniquely diverse student body. These programs include research, scholarship, creative activity, and community service. With the support of the administration, staff, alumni, and community, highly qualified faculty are the keystone of the University and the basis for the excellence of our programs.*

*As a comprehensive university, California State University, Los Angeles offers a broad range of liberal and professional programs designed to encourage student excellence, achievement and well being. Facilitated by close interaction between faculty and students, educational programs are designed to foster habits of disciplined inquiry and critical thinking while helping students master a body of knowledge. The University strives to promote understanding of and respect for diversity, and to serve the changing needs of a global society. Recognizing its commitment to teaching, research, scholarship, creative activities, and service, the University supports an effective library and the use of new technologies that enrich the instructional process and provide effective access to information in its various forms.*

*The University is committed to providing students with a balanced and well-rounded educational experience including co-curricular activities that contribute to personal enrichment, leadership development, and institutional pride. Student organizations, campus residence life, artistic events, multicultural events, intercollegiate athletics, and intramurals are designed to be a significant part of this experience.*

*The close proximity of the University to civic, cultural, and economic centers enables it to foster strong cooperative relationships with alumni, community, business, scientific, educational, cultural, and government constituencies. Partnerships with these constituencies will continue to grow for the mutual enhancement of academic programs and the community.*

*California State University, Los Angeles is committed to fostering collegial relationships among faculty, administration, students, and staff. The principles of academic freedom and professional ethics are the responsibility of the entire academic community. We take pride in our continuing evolution as the University serving the Los Angeles Basin.*

**The College of Engineering, Computer Science and Technology’s Mission**

Our college mission is to successfully prepare the next generation of engineering, computer science and technology professionals for Los Angeles and beyond.

Figure 2.1 shows the college Vision and Mission statements as developed through a 2014 comprehensive strategic planning process involving ECST faculty, staff, and students.



**Figure 2.1: College of ECST Vision and Mission Statements.**

**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.*

1. **Program Educational Objectives**

Program Educational Objectives are broad statements that describe what graduates are expected to attain within a few years of graduation. They provide curricular guidelines that offer a vision for the program.

The Program Educational Objectives of the undergraduate program in Computer Science at California State University, Los Angeles are as follows.

1. *Students who had entered 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 had continued 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 knowledge and skills.*

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 had entered 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 *“To prepare the next generation of … computer science… professionals..”* and the University Mission Statement “prepares students for success … in their careers, and throughout their lives”.

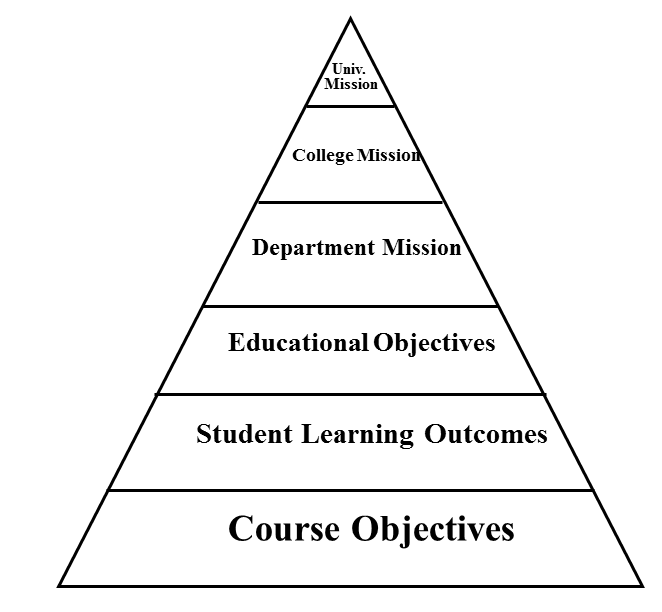
The **second Program Educational Objective** “*Students who had continued in academia will have been successful in pursuing advanced degrees ...”* is consistent with Department’s Mission statement “…*who are prepared to meet the challenges …”,*  the College’s Mission statement “… *prepare the next generation of ..computer science.. professionals …”*  and the University Mission Statement *“prepares students for success in advanced studies, in their careers...*”.

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 “*prepare the next generation of … computer science… professionals for Los Angeles and beyond …”*  and the University Mission Statement *“prepares students for success …. throughout their lives … . committed to providing students with a balanced and well-rounded educational experience … that contribute to personal enrichment, leadership development.”*

The consistency of Mission statements, Program Educational Objectives, Student Learning Outcomes, and Course Objectives is best described by the pyramid shown in Figure 2.2.

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 university.



**Figure 2.2: 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. *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.
    2. *Industry Advisory Board*: The Industry Advisory Board (IAB) consists of industry leaders whose advice is critical in preparing our students. In addition, the IAB provides feedback and suggestions for formulating our Program Educational Objectives. The IAB typically meets once a year in the Spring as part of the Senior Design project presentations. After conversion to semesters in 2016, a second meeting in Fall was added to enhance discussions with the IAB. A formal survey is taken to record feedback after every IAB meeting. (<http://csns.calstatela.edu/wiki/content/assessment/iab/>).
    3. *Faculty*: Faculty design and deliver the undergraduate curriculum. Their responsibility is to ensure that the curriculum is up-to-date and reflective of changing technologies and the skills expected of our graduates. An annual faculty retreat is conducted during the Winter/Spring terms.

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

Besides these constituencies the program adheres to the following frameworks.

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

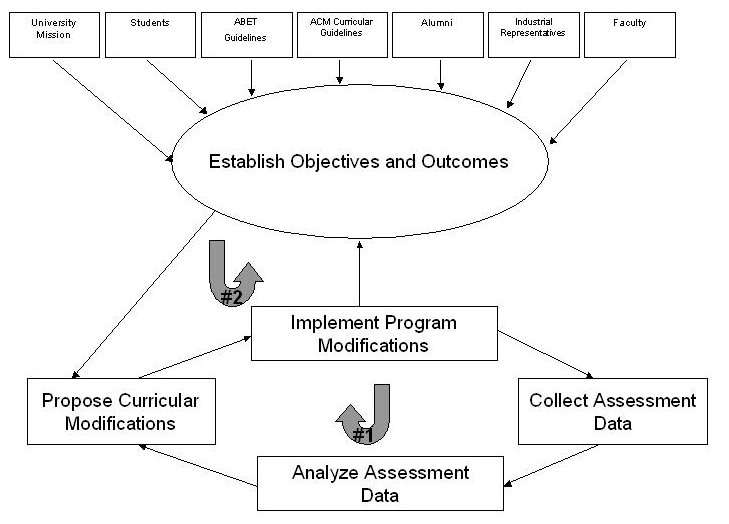
*ABET guidelines:* As the external accreditation agency, ABET assures the program’s effectiveness by setting accreditation standards. These standards provide an important input in guiding the Program Educational Objectives and Student Learning Outcomes.

*ACM curricular guidelines:* ACM/IEEE curriculum guidelines provide important advice with respect to course curricula. The Department engages in an ongoing discussion of curriculum, which ensures that the program is up to date.

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

The Department established an assessment process for the establishment and revision of the Program Educational Objectives. As depicted in Figure 2.3, the process consists of two-loops. The outer loop, Loop #2, shows the periodic process involved in establishing and revising the Program Educational Objectives. Loop #2 is a relatively slow loop; assessment occurs usually once every three to five years. Loop #2 can be accelerated if needed based on assessment data.

Program Educational Objectives which were in effect in the last ABET CAC visit in Fall 2012 were revised in Winter 2016. The actual mechanism for revision of Program Educational Objectives is as follows.



**Figure 2.3: Assessment Process**

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. Feedback from surveys
      1. Email surveys (Alumni, Faculty and IAB) are conducted annually during Loop #1 activities. In addition, periodically (typically once every 3 years) additional questions are asked concerning LOOP #2 activities. These questions deal with the effectiveness of the PEOs. Stakeholders are asked to make suggestions regarding PEO changes.
   6. IAB meetings
      1. The Industry Advisory Board consists of current employers of our alumni and potential employers of our current students. They are asked to provide input to the PEOs.
2. Faculty retreat. During the annual retreat, faculty consider possible revisions to the Program Educational Objectives as proposed by the Assessment Committee. Faculty vote on whether to adopt proposed PEO changes.

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

|  |  |
| --- | --- |
| **Date** | **Activities** |
| 2012 to 2016  (Spring Quarter) | Annual constituency surveys are conducted |
| 2012 to 2016  (Winter Quarter) | Annual faculty retreats |
| 2012-2016  (Spring Quarter) | Annual IAB meetings |
| Winter 2016 | Assessment Committee proposed revised Program Educational Objectives. Adopted by faculty. |
| Spring 2016 | Constituency (alumni, IAB and faculty) surveys were conducted to validate the importance of Program Educational Objectives |
| 2016 to 2017  (Fall Semester) | Annual IAB meetings |
| 2016 to 2018  (Spring Semester) | IAB meeting + Senior Design project presentations |
| 2016 to 2018  (Spring Semester) | Annual constituency surveys conducted |
| 2016 to 2018  (Spring Semester) | Annual Faculty retreats |

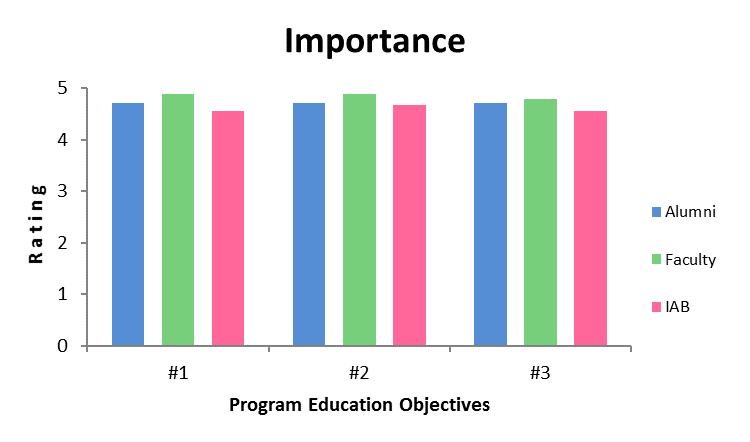
**Table 2.1: Program Educational Objectives review timeline**

Prior to Spring 2016, our Program Educational Objectives were as follows:

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.*

As described in Table 2.1, the Program Educational Objectives were revised in Winter 2016. Minor revisions were made to clarify the meanings to PEO #1 and PEO #2 (See Constituent comments below)

Program constituency surveys were conducted in Spring 2016 to validate the importance of the revised Program Educational Objectives. This step obtains final approval from all the constituencies and completes the process of revising the Program Educational Objectives.



**Figure 2.4: Importance of Program Educational Objectives**

As shown in Figure 2.4, all three Program Educational Objectives are deemed important. Assuming all constituencies are weighted equally, all the Program Educational Objectives received a ranking higher than 4.0 out of a possible 5.0.

* Program Educational Objective #1 received an average ranking of 4.72;
* Program Educational Objective #2 received an average ranking of: 4.76;
* Program Educational Objective #3 received an average ranking of: 4.68;

Thus, the Program Educational Objectives are acceptable to the program constituencies.

**Constituency Member Comments**

A few pertinent comments are given below.

* *Make a change in tense to the past tense in the statements. They will then reflect objectives being met 3-5 years after graduation. i.e., (i) Change PEO #1 from "Students who enter...." to "Students who had entered....." (ii)Change PEO #2 from "Students who continue..." to "Students who had continued...."*
* *I'd suggest changing "new skills and new technologies" to "new knowledge and skills".*
* *The long term goal 3 to 5 years out are captured by the thee listed.*
* *A lot of the essential knowledge and skills come from the more advanced courses. Students should be encouraged to pursue an advanced degree.*
* *A strong foundation while in school should help the students with all the objectives.*

# 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 had entered 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 both a theoretical foundation and the skills necessary to succeed as a software professional. Thus, students who are successful in 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 had continued in academia will have been successful in pursuing advanced degrees and in demonstrating their ability to master advanced areas of computer science.*

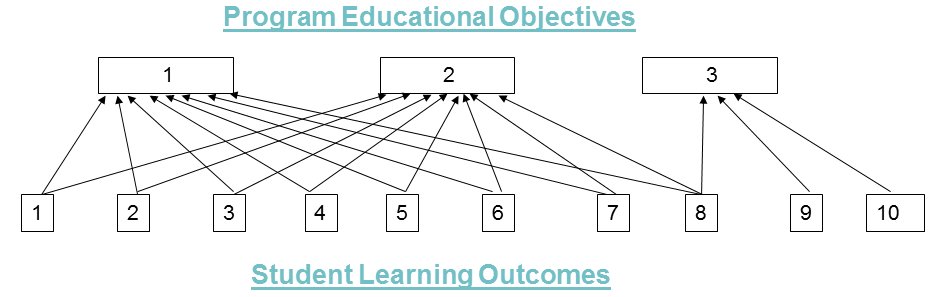
The following Student Learning Outcomes are largely concerned with the development of the theoretical understanding and skills that are necessary to succeed as a graduate student.

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 knowledge and skills*

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 Establishing and Revising Student Learning Outcomes**

Our Department established an assessment process for establishing and revising Student Learning Outcomes. This process is similar to that for the Program Educational Objectives, as described in Section E under Criterion 2.

1. Our Assessment Committee (Raj Pamula, Chengyu Sun, Russ Abbott) develops possible revisions 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. Feedback from surveys
      1. Email surveys (Alumni, Faculty and IAB) are conducted annually during Loop #1 activities. In addition, periodically (typically once every 3 years) additional questions are asked concerning LOOP #2 activities. These questions deal with the effectiveness of the SLOs. Stakeholders are asked to suggest possible changes to them.
   6. IAB meetings
      1. Industry Advisory Board consists are current employers of our alumni and potential employers to our current students. They are asked to provide input to the SLOs.
2. Faculty retreat. Faculty consider revisions to the SLOs as proposed by the Assessment Committee. Faculty vote to adopt any changes to the SLOs

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

|  |  |
| --- | --- |
| **Date** | **Activities** |
| 2012 to 2016  (Spring Quarter) | Annual constituency surveys are conducted |
| 2012 to 2016  (Winter Quarter) | Annual faculty retreats |
| 2012-2016  (Spring Quarter) | Annual IAB meetings |
| Spring 2016 | Constituency (alumni, IAB and faculty) surveys were conducted to validate the importance of SLOs |
| 2016 to 2017  (Fall Semester) | Annual IAB meetings |
| 2016 to 2018  (Spring Semester) | IAB meeting + Senior Design project presentations |
| 2016 to 2018  (Spring Semester) | Annual constituency surveys are conducted |
| 2016 to 2018  (Spring Semester) | Annual Faculty retreats |

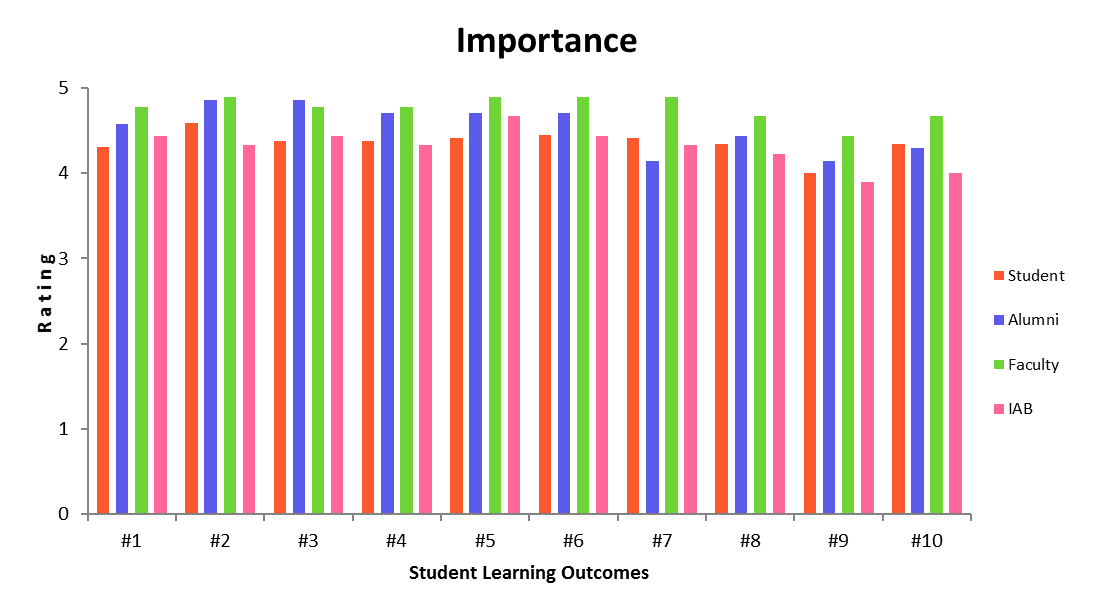
**Table 3.1: Student Learning Outcomes review timeline**

The current Student Outcomes were in place for the Fall 2012 ABET visit. During the 2016 cycle, the faculty considered switching to ABET criteria defined by “a” thru “k” as the new SLOs. The ABET criteria “a” thru “k” aligned well with our existing SLOs. However, we decided to wait as the ABET CAC commission was considering revising the “a” thru “k” criteria to a new set of required SLOs.

Constituency Surveys were conducted to validate the importance of Student Learning Outcomes in Spring 2016.

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 4.53/5;
* Student Learning Outcomes #2 received a ranking of 4.67/5;
* Student Learning Outcomes #3 received a ranking of 4.62/5;
* Student Learning Outcomes #4 received a ranking of 4.55/5;
* Student Learning Outcomes #5 received a ranking of 4.67/5;
* Student Learning Outcomes #6 received a ranking of 4.62/5;
* Student Learning Outcomes #7 received a ranking of 4.44/5;
* Student Learning Outcomes #8 received a ranking of 4.42/5;
* Student Learning Outcomes #9 received a ranking of 4.12/5;
* Student Learning Outcomes #10 received a ranking of 4.33/5;



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

The Student Learning Outcomes are acceptable to the program constituencies. 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 industry needs.

**D.    Enabled Student Characteristics**

Our required courses enable students to attain all ABET characteristics (a) thru (k) by the time of graduation. This is described in two ways:

1. Describing how each of the ABET characteristics is enabled in the core courses.
2. Mapping the ABET CAC criteria to the Student Learning Outcomes
3. This section describes how each of the ABET characteristics are enabled in the core courses.
   * + - 1. *An ability to apply knowledge of computing and mathematics appropriate to the discipline.*

***Relevant Courses:*** Math core; CS core

By the time students take the capstone CS 4963 course, Students have taken:

* 5 math core classes: Calculus I, Calculus II, Linear Algebra, Discrete Structures, a lower division MATH elective in Probability/Statistics.
* 11 computer science core classes: Programming, Data Structures, Algorithms, Automata Theory, Operating Systems, Computer Systems, Database systems, Software Engineering, Computer Ethics, Software Project I,II).
* 6 technical Computer Science elective courses.
* About 15% of their required units in mathematics and around 55% of their units in computing.

CS 4963 provides a recapitulation of the primary concepts in Mathematics and Computer Science.

* + - * 1. *An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.*

***Relevant Courses:*** CS 2013, CS 3220, CS 3337, CS 4440, CS 4961, CS 4962

In CS 3337, groups of students choose their own projects. The deliverables include a high-level requirements, technical specifications, detailed design document and presentation materials.

In CS 4961 and CS 4962, teams of students are given a project, usually by an outside stakeholder. There is typically a weekly or bi-weekly discussion between the student team and a stakeholder liaison to ensure that the requirements are appropriate and that the design and development are on track. Each team works with a faculty advisor, who oversees the project. All of the groups are required to submit a requirements document, a technical specifications, and a detailed design document.

These projects demonstrate the students’ ability to analyze a problem and to identify and define the computing requirements appropriate to its solution.

* + - * 1. *An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.*

***Relevant Courses:*** CS 3220, CS 3337, CS 4961, CS 4962

In CS 3337, groups of students choose their own projects. The deliverables include a high-level requirements document, a technical specification, a detailed design document, and an implementation of a prototype along with presentation materials.

In CS 4961 and CS 4962, teams of students are given a project, usually by an outside stakeholder. Each group is required to create a detailed design document and complete the project implementation. The outside stakeholders and the faculty advisor approve each of the deliverables and ensure that the project is on track.

Each of the deliverables in CS3337 and CS4961 enables students to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

* + - * 1. *An ability to function effectively on teams to accomplish a common goal.*

***Relevant Courses:*** CS 3337, CS 4961, CS 4962

Students work in groups both in CS 3337 and in the Senior Design Project sequence (CS4961-CS4962) to design and develop a software project. Students evaluate each other based on a rubric with five indicators: Participation, Problem-solving, Attitude, Contribution, and Interaction.

Students learn such team building skills such as:

* Planning and scheduling individual and team efforts to complete required work by the established deadlines
* Working productively within a team despite inevitable conflicts.
* Assuming various specialized roles within the team.
  + - * 1. *An understanding of professional, ethical, legal, security and social issues and responsibilities.*

***Relevant Courses:*** CS 3801, CS 4961, CS 4962

CS 3801 covers various topics on Societal and Ethical issues in Computing.

* Professional and ethical issues
* Ethical thinking in general
* Issues found in the ACM Code of Ethics,
* Software developer’s professional responsibility regarding cybersecurity
* Ethical outcomes regarding the effects of computers on people.
* Possibility of strong AI (might machines have rights?)
* Privacy and data sharing
* Autonomous vehicles
* Legal issues
* Intellectual property law
* Lawsuits and intellectual property disputes in the tech industry
* Social issues:
* Cyber age
* Social media
* Hacktivism
* Free/open source software
* Automation and employment

These principles are taken into account in the Senior Design sequence (CS4961-CS4962).

* + - * 1. *An ability to communicate effectively with a range of audiences.*

***Relevant Courses:*** Gen. Ed., CS 3337, CS 4961, CS 4962

Oral communication is emphasized early in a General Education required course such as COMM 1100. This course teaches effective oral communication in a public speaking forum.

Oral communication is also made an important factor in many Computer Science courses. It is emphasized especially in CS 3337, CS 4961, and CS 4962. Students are evaluated based on a rubric with five indicators: Logical organization, English language, Technical vocabulary, Presentation aids, and Audience interaction.

* CS3337 has final group presentations on their project.
* In CS4961-CS4962, every student on the team will give several presentations over the course of the project.
* CS4962 has a culminating event referred to as an Expo. Student teams make a formal presentation of their project in front of the Department Industry Advisory Board and project liaisons.

In addition, many Computer Science core courses integrate oral presentations in the courses.

* + - * 1. *An ability to analyze the local and global impact of computing on individuals, organizations, and society.*

***Relevant Courses:*** CS 3801, CS 4961, CS 4962

CS 3801 covers various topics on Societal and Ethical issues in Computing including the following aspects of the impact of computing.

* The effects of social media on politics
* Cyberbullying
* The effects of automation on employment
* The effects of globally available information
* The effects of automation on consumers, standards of living
* The use and effects of software in law enforcement
* The “Dark Web”
* The effects of hacktivist groups like Anonymous
* Warfare in the cyber age

These principles covered are taken into account in the senior Software Design sequence (CS 4961-CS 4962).

* + - * 1. *Recognition of the need for and an ability to engage in continuing professional development.*

***Relevant Courses:*** CS 4961, CS 4962, CS 4963

In CS 4961-CS 4962, students work with the stakeholders to determine the best solution to a given need. This typically requires that students learn new tools and technologies, thereby providing an important lesson about continuing professional development. Students write a Lifelong Learning essay in CS4962. The essay prompt requires student to discuss:

* What was learned (software library, framework, programming language, software tools, software systems etc.,)
* How it was learned (web research, software practice, consult someone etc.,)
* What they have learned about learning (What do you know now about learning new technologies that you didn't know before? How will this experience influence your approach learning other technologies?)
* What are their realistic career goals (short-term and long-term) after graduation?

In CS 4963, every student completes a mock technical interview with a group consisting of faculty and industry representatives. The goal is to expose students to the technical interview process. Students attend workshops and/or review videos to advance their professional skills. A few topics are listed below:

* How to Communicate in Technical Interviews & What to do When you're Stumped
* What is ‘Project Work’ and what Does it Look Like on a Resume?
* Job Opportunity Spotlight: The Software Engineer, Tools & Infrastructure New Grad Role
* How to Transition Your Academic Skills to the Tech Industry.

The CS 4961-CS 4962-CS 4963 capstone courses encourage students to recognize the importance of Lifelong Learning and the ability to engage in continuing professional development.

In addition each year, during National Engineers' Week, the College sponsors a major industry career day. During this event, representatives from industry, government and private practice are set up information tables, and meet with students throughout the day. Typically representatives from 20 to 30 companies are present at these events. This gives students an opportunity to meet prospective employers and learn their current engineering needs.

* + - * 1. *An ability to use current techniques, skills, and tools necessary for computing practice.*

***Relevant Courses:*** CS 201X, CS 3220, CS 3337, CS 4961, CS 4962

In almost every Computer Science course, students practice working with the tools and developing the skills necessary for computing. Students are exposed to:

* Multiple integrated development environments (IDEs),
* Many software packages (database management systems, web, etc.,),
* Operating system tools and frameworks
* Software Project Management Tools, such as Microsoft Project
* Software Requirement Tools, such as Workspace
* Software Version Control Tools, such as SVN, GitHub
* Many programming languages (Java, C++, Haskell, JavaScript, Python, etc.,)
* Lynda.com (the online software training resource that provides thousands of video tutorials on the latest software tools and techniques) as a paid subscription from the university.

By the time students take CS4961 and CS4962 in their senior year, they are capable of learning new techniques, skills, and tools necessary to complete their projects.After completion of CS4962, students are ready for entry-level positions in software development. (Some students are considerably more advanced.)

* + - * 1. *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.*

***Relevant Courses:*** CS 2148, CS 3112, CS 3168, CS 3445, CS 4440, CS 4961, CS 4962

Students are exposed to mathematical foundations in CS 2148, to algorithmic principles in CS3112, to computer science theory in CS3186, and to the design of computer-based systems in CS3337.

The Senior Design sequence (CS4961, CS4962) gives students an opportunity to design a software system that addresses a realistic problem. Student teams make presentations at the end of both CS4961 and CS4962. Students address various topics/issues in their presentations and explain their design choices. They address such design issues as:

* Which programming language, data structures, operating platform were chosen and why were they chosen.
* Which architecture and design patterns were chosen and why were they chosen.
* What tradeoffs, if any, were involved in their design choices.
* What challenges and obstacles they had to overcome.
  + - * 1. *An ability to apply design and development principles in the construction of software systems of varying complexity.*

***Courses:*** CS 3337, CS 4961, CS 4962

CS 3337 covers methodologies and tools for the requirements analysis and design of large complex software system.

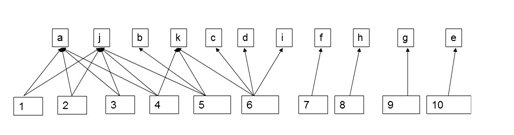
In the Senior Design sequence (CS4961, CS4962), students design and develop a software project that addresses a realistic problem.

The main goals of these project-oriented courses is to require that students undertake complex software projects and see them through to completion—from pre-requirements to delivery. This process entails:

* Development of a precise problem statement
* Development of a written project plan
* Development of an implementation of the design
* Demonstration of the results of their work
* Learning new technologies and utilizing resources available for the completion of the project.

1. This section describes the mapping of ABET CAC criteria to the Student Learning Outcomes

Figure 3.3 and Table 3.2 show the mapping of ABET criteria (“a” thru “k”) to the Student Learning Outcomes (1 to 10). Each ABET criterion is thus related to one or more Student Learning Outcomes. By demonstrating the achievement of Student Learning Outcomes (See Criterion 4), the undergraduate B.S degree program in Computer Science demonstrates its compliance with all the ABET characteristics.



**Figure 3.3: 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. | 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. |
| (b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution. | SLO# 5. 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. | SLO# 6. 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. | SLO# 6. 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. | SLO# 10. 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. | SLO# 7. 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. | SLO# 9. 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 | SLO# 8. 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. | SLO# 6. 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. | 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.  SLO# 5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions. |
| (k) An ability to apply design and development principles in the construction of software systems of varying complexity. | SLO# 4. Students will have a fundamental understanding of computer systems.  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. |

**Table 3.2: ABET criteria – Student Learning Outcomes Map**

# CRITERION 4. CONTINUOUS IMPROVEMENT

Continuous improvement is best accomplished when the assessment process is integrated with an underlying course-level Learning Management System. This section describes Computer Science Network Services (CSNS), a system developed by Prof. Chengyu Sun to meet this objective.

CSNS is a web-based software system that integrates program assessment with learning management. It simplifies and improves the processes involved in the collection, analysis, and presentation of assessment data. In addition, the assessment functions that are built into the learning management system, which faculty and students use on a daily basis, encourage and facilitate a continuous and sustainable assessment process.

CSNS provides the following program assessment features.

* Managing and mapping of student learning outcomes and courses.
* Embedding assessment activities and data collection in gateway courses where it is particularly important to gauge student progress.
* Managing rubrics that can be used both in class and for program assessment; supporting rubric evaluation by instructors, external reviewers, and peers.
* Managing surveys that can be used both in class and for program assessment.
* Importing and analyzing data from external standardized tests such as Major Field Tests (MFT).
* Aggregating and presenting rubric, survey, and MFT data as graphs, which provide visual summaries of the student learning outcomes measurements.
* Facilitating the creation of course journals and making their management more convenient for faculty.
* Supporting mailing lists, which are automatically updated when the academic standing of students change. This helps the department keep in touch with both students and alumni.
* Hosting program assessment documents on the CSNS wiki and file manager, which makes it easier to share and collaborate on those documents.
* Providing a program assessment dashboard page, which connects everything together and presents an overview of the program assessment process.

CSNS is a significant advance over previously available program assessment systems. Dr. Sun has written and presented two papers describing it. (See Appendix 1.x)

* “Building Assessment Functions into LMS for Efficient and Sustainable Assessment Processes,” *ABET Symposium* 2016
* “Combining Program Assessment with Learning Management for Efficient and Sustainable Accreditation Processes,” *ASEE* 2016

The Department’s continuous improvement methodology includes an evaluation of the Student Learning Outcomes (SLOs) to ensure continuous improvement. The program assessment process is described by *Loop #1*, the inner loop in Figure 2.5. The outcomes are assessed annually (Section A below), which prompts any needed curricular adjustments (Section B below).

***A: Student Outcomes***

This section describes the ongoing assessment process of student outcomes at the detailed level and the data collected to support it. It consists of the following subsections.

**A.1:** Mapping of the required Computer Science courses to SLOs

**A.2:** Description of measures used in data collection

**A.3**: Data collection process

**A.4**: Analysis of the MFT results

**A.5**: Analysis of the CS4963 results

**A.6:** Evaluation of achievement of SLOs

**A.1: Mapping of required Computer Science courses to SLOs**

Table 4.1 indicates which of the required Computer Science courses contribute to the achievement of SLOs.

Courses in which SLOs are either introduced, reinforced or emphasized are marked by the letter “X”. In some of those courses, assessment data is evaluated for the achievement of SLOs as described in Section A.6.

|  | SLO #1 | SLO #2 | SLO #3 | SLO #4 | SLO #5 | SLO #6 | SLO #7 | SLO #8 | SLO #9 | SLO #10 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CS 1010 |  | X |  |  |  |  | X | X | X | X |
| CS 1222 |  | X |  | X |  |  |  |  |  |  |
| CS 2011 |  | X | X |  | X | X |  |  |  |  |
| CS 2012 |  | X | X |  | X | X |  |  |  |  |
| CS 2013 |  | X | X |  | X | X |  |  |  |  |
| ENGL 2030 |  |  |  |  |  |  | X |  |  |  |
| CS 2148 | X |  | X |  |  |  |  |  |  |  |
| CS 3035 |  | X |  |  |  |  |  |  |  |  |
| CS 3112 | X |  | X |  |  |  |  |  |  |  |
| CS 3186 | X |  |  |  |  |  |  |  |  |  |
| CS 3220 |  | X |  | X |  |  |  |  |  |  |
| CS 3337 |  |  |  |  | X | X | X |  |  |  |
| EE 3445 |  |  |  | X |  |  |  |  |  |  |
| CS 3801 |  |  |  |  |  |  |  | X | X | X |
| CS 4440 |  |  |  | X |  |  |  |  |  |  |
| CS 4961 |  |  |  |  | X | X | X | X | X | X |
| CS 4962 |  |  |  |  | X | X | X | X | X | X |
| CS 4963 | X | X | X | X |  |  |  | X |  |  |

**Table 4.1: SLOs to Courses Map**

**A.2: Description of Measures in data collection**

We use the following measure tools to evaluate SLOs.

***Rubrics***

A rubric measures student work. Each rubric has a number of performance indicators. Each indicator is evaluated on a scale of 1 to 5. For each rating a description characterized the sort of work for which that rating applies. We developed the following rubrics as documented at <https://csns.calstatela.edu/department/cs/rubric/list>

* Ethics in the Computer Age
* Lifelong Learning
* Oral Communication
* Software Engineering – Requirements
* Software Engineering – Design
* Software Engineering – Implementation
* Team Work
* Written Communication

See Appendix x.x for a description of these rubrics. The rubrics are evaluated on CSNS in the relevant courses.

***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. In some cases, the artifacts collected from these measures are evaluated by a rubric.

***Major Field Test***

The Major Field Tests (MFT) are designed by the Educational Testing Service (ETS) to measure the level of knowledge and understanding attained by students. The MFT exams are currently utilized by over 200 institutions and more than 9,100 students.

ETS offers comprehensive national comparative data, which enables us to evaluate student performance and to compare our program's effectiveness to programs at similar institutions. In addition, ETS provides a subscription service that compares our students’ results to 10 or more peer institutions. Our selected comparison peer institutions includes five other California State University campuses as well as five non-CSU campuses. We first subscribed to this service beginning in Fall 2016.

The MFT provides three indicators:

* Indicator #1: Programming and Software Engineering
* Indicator #2: Discrete Structures and Algorithms
* Indicator #3: Systems: Architecture/Operating Systems/Networking/Database

ETS only reports the Assessment Indicators as an aggregate for the groups of students taking the test at the same time. These indicators are essentially class averages and are also closely tied to our learning outcomes.

Students taking CS4963 Computer Science Recapitulation, the capstone course, are required to take the MFT. CSNS imports both student scores and the national score distribution. It provides functions to manage, analyze, and visualize the data.

***Surveys***

Surveys are indirect measures that gather perceptions, opinions, and reflections on learning. Surveys also provide a means to ask qualitative open-ended questions. Every year we survey four of our constituencies (students, faculty, alumni, and industry partners) on their satisfaction with our learning outcomes. The feedback indicates how well we achieve each learning outcome.

* These surveys are conducted annually on CSNS.
* CSNS mailing lists are used to conduct student and alumni surveys.
* Faculty surveys are conducted following the annual faculty retreats.
* The surveys of industry partners are conducted following the annual Industrial Advisory Board (IAB) meetings.
* The survey results are aggregated, tabulated, and charted in CSNS.

**A.3: Data collection process**

Data is collected from measures taken in a number of courses. All required courses are scheduled for data collection at least once a year. Surveys are collected annually and reported on a two year cycle.

SLO data collection is described in Table 4.2. A few characteristics of Table 4.2 are as follows.

* Rows describes the information for each SLO
* Columns provides the information on:
  + Data – Indicates what was collected
  + Type – Indicates the type of data
  + Target - Refers to the threshold or 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.
  + Description – Provides a brief description as to where/how the data is collected. See the expanded description in the course syllabus (Appendix x.x)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SLO** | **Data** | **Type** | **Target Thresholds** | **Description** |
|  | 1.CS4963 Assessment Indicator #1  2.MFT Assessment Indicator #2  3.[SLO-1 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo1) | Assignments & Rubric  MFT  Survey | 3 or higher  50th percentile or higher  3 or higher | Instructor conducts assignments and exams. They are added and normalized on a 5 point scale for each student. (Rubric K)  Assessment Indicator AI-2 on MFT provides the national percentile the institution is in based on the mean score of the students.  Constituent surveys for this SLO. |
|  | 1.[CS4963 Assessment Indicator #2](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs490_ai2)  2.[MFT Assessment Indicator #1](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/mft_ai1)  3.[SLO-2 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo2) | Assignments & Rubric  MFT  Survey | Class average > 3.0  50th percentile or higher  3 or higher | Instructor conducts assignments and exams. They are added and normalized on a 5 point scale for each student. (Rubric K)  Assessment Indicator AI-1 on MFT provides the national percentile the institution is in based on the mean score of the students.  Constituent surveys for this SLO. |
|  | 1.[CS4963 Assessment Indicator #3](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs490_ai3)  2.MFT Assessment Indicator #2  3.[SLO-3 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo3) | Assignments & Rubric  MFT  Survey | Class average > 3.0  50th percentile or higher  3 or higher | Instructor conducts assignments and exams. They are added and normalized on a 5 point scale for each student. (Rubric K)  Assessment Indicator AI-2 on MFT provides the national percentile the institution is in based on the mean score of the students.  Constituent surveys for this SLO. |
|  | 1.[CS4963 Assessment Indicator #4](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs490_ai4)  2.[MFT Assessment Indicator #3](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/mft_ai3)  3.[SLO-4 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo4) | Assignments & Rubric  MFT  Survey | Class average > 3.0  50th percentile or higher  3 or higher | Instructor conducts assignments and exams. They are added and normalized on a 5 point scale for each student. (Rubric K)  Assessment Indicator AI-3 on MFT provides the national percentile the institution is in based on the mean score of the students.  Constituent surveys for this SLO. |
|  | 1.[CS3337 & CS4961 Requirements](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs337_pra)  2.[SLO-5 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo5) | Rubric  Survey | 3 or higher on each indicator  3 or higher | Requirements documents evaluated by Instructor (Rubric R)  Constituent surveys for this SLO. |
|  | 1.[CS3337 & CS4962 Design](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs337_pra)  2. [CS3337 & CS4962 Implementation](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs437_dev)  3.CS3337,CS  4961,CS4962 Team  4.[SLO-6 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo6) | Rubric  Rubric  Rubric  Survey | 3 or higher on each indicator  3 or higher on each indicator  3 or higher on each indicator  3 or higher | Design documents evaluated by Instructor (Rubric D)  Project Implementation evaluate by Instructor (Rubric I)  Student and Instructor – rubric evaluations (Rubric T)  Constituent surveys for this SLO. |
|  | 1. [CS3337,CS 4961,CS4962 Oral](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs437_dev)  2.[CS4961, CS4962](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs491b_dev) Written  3.[SLO-7 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo6) | Rubric  Assignment & Rubric  Survey | 3 or higher on each indicator  3 or higher on each indicator  3 or higher | Project Presentation evaluated by Instructor– (Rubric O)  Writing assignments evaluated by Instructor –(Rubric W)  Constituent surveys for this SLO. |
|  | 1.CS4962 Life Long Learning  2.[SLO-8 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo8) | Assignment & Rubric  Survey | 3 or higher  3 or higher | Instructor – Rubric evaluations (Rubric L)  Constituent surveys for this SLO. |
|  | 1.[CS3801 Computing](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs437_dev)  2.[SLO-9 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo8) | Assignments & Rubric  Survey | 3 or higher on each indicator  3 or higher | Instructor conducts assignments, exams and presentations. The scores are added and normalized on a 5 point scale for each student. (Rubric E)  Constituent surveys for this SLO. |
|  | 1.[CS301 Ethics](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/cs437_dev)  2.[SLO-9 Satisfaction Survey](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/slo8) | Rubric evaluations  Survey | 3 or higher on each indicator  3 or higher | Instructor conducts assignments, exams, and presentations. The scores are added and normalized on a 5 point scale for each student. (Rubric E  Constituent surveys for this SLO. |
| [**1-4**](http://csns.calstatela.edu/wiki/content/assessment/undergrad/SLO/)  **Over-all** | 1.[MFT Median Score Percentile](http://csns.calstatela.edu/wiki/content/assessment/undergrad/Learning_Outcomes_Graphs/mft_median) | MFT score | 50th Percentile or higher | MFT Median data comparison of CSULA students when compared to all other students |

**Table 4.2: Data Collection**

**A.4 Evaluation of MFT results**

This section presents and analyzes our MFT results. To review, the MFT provides individual test scores and three group indicators:

* *Indicator #1 (AI-1):* Programming and Software Engineering
* *Indicator #2 (AI-2):* Discrete Structures and Algorithms
* *Indicator #3 (AI-3):* Systems (Architecture, Operating Systems, Networking, and Databases)

These indicators are essentially class averages and are also closely tied to our learning outcomes. Since 2006, CSULA has required all graduating students to take the MFT. The tables and graphs below shows the results for the current evaluation period since 2012.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Year | # OF STUDENTS | # OF QUESTIONS CORRECT | | | National Percentiles | | |
|  |  | AI-1 | AI-2 | AI-3 | AI-1 | AI-2 | AI-3 | |
| 2012 | 22 | 54 | 47 | 55 | 69 | 82 | 96 | |
| 2013 | 28 | 56 | 50 | 47 | 75 | 88 | 84 | |
| 2014 | 32 | 55 | 49 | 46 | 73 | 87 | 91 | |
| 2015 | 28 | 60 | 45 | 51 | 87 | 74 | 93 | |
| 2016 | 62 | 57 | 46 | 47 | 62 | 64 | 75 | |
| 2017 | 65 | 54 | 43 | 47 | 49 | 51 | 75 | |

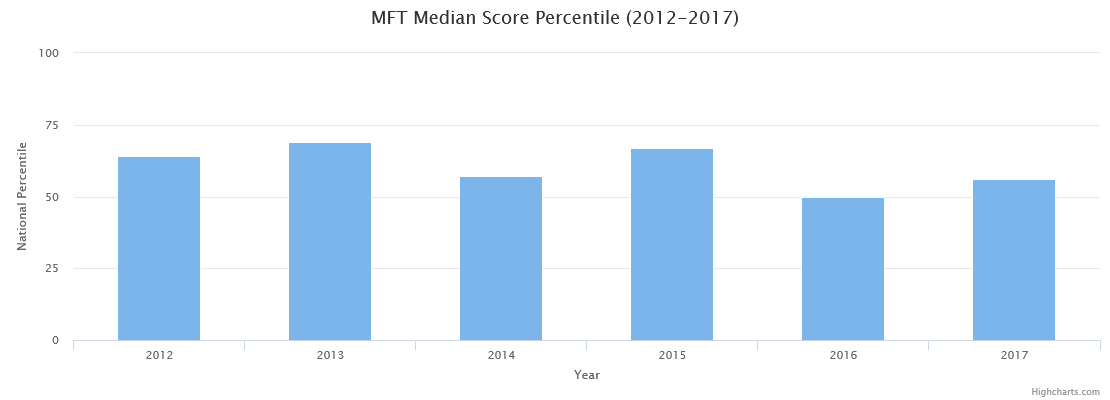
**Table 4.3 Assessment Indicators (yearly)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Period** | **2012 – 2015 (Test-4LMF)** | | | **2016 – 2017 TEST-4HMF)** | | |
| **Sample** | **CSULA** | **Peer** | **National** | **CSULA** | **Peer** | **National** |
| **# of students** | 110 | 571 | 7741 | 127 | 549 | 5294 |
| **Median score** | 154 | 147 | 147 | 151 | 145 | 148 |

**Table 4.4 Median Scores (test cycle)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Period** | **2012 – 2015 (Test-4LMF)** | | **2016 – 2017 TEST-4HMF)** | |
| **Sample** | **Peer** | **National** | **Peer** | **National** |
| **CSULA median student percentile** | 61% | 61% | 61% | 56% |

**Table 4.5 Median Score Percentiles (test cycle)**



**Figure 4.1 Median Score Percentiles (yearly)**

**Figure 4.2 Median Score Percentiles (test cycle)**

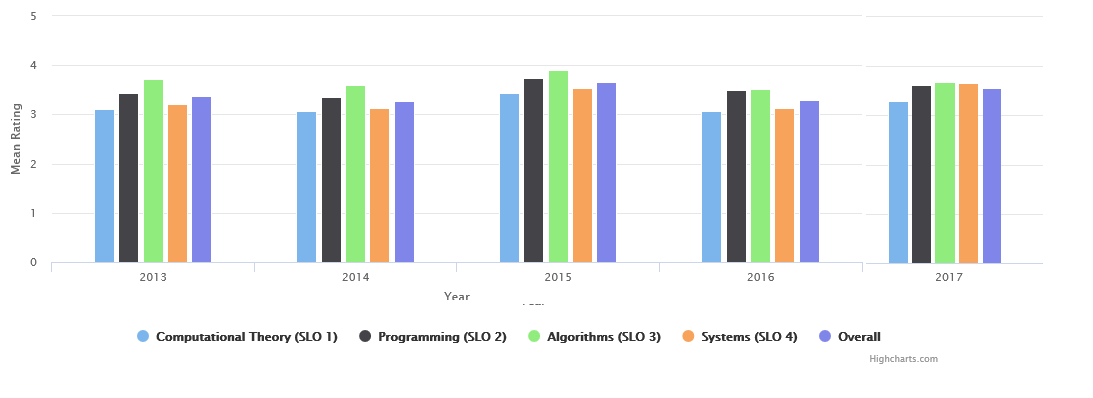
**Figure 4.3 Assessment Indicators (yearly)**

**Figure 4.4 Assessment Indicators (test cycle)**

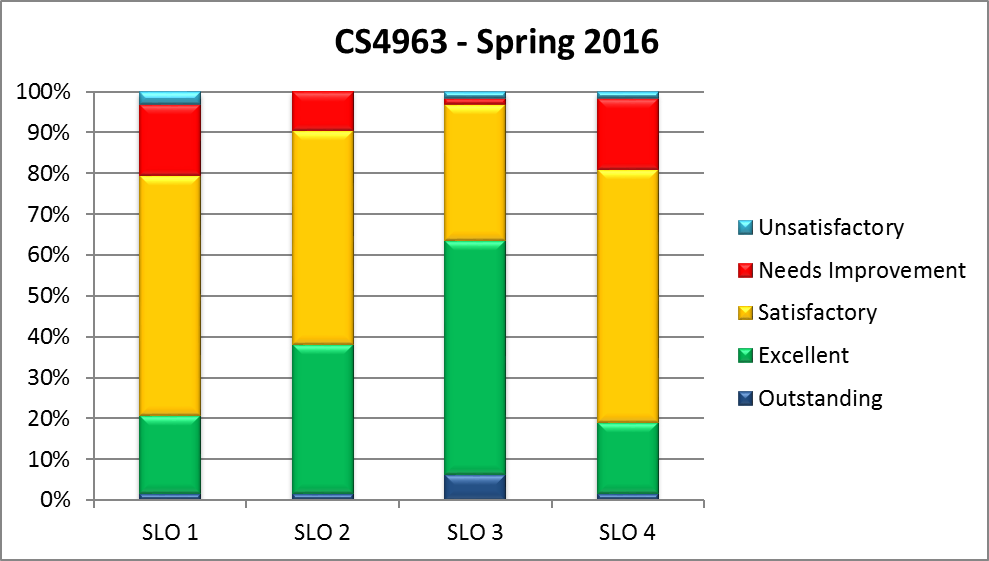
A few characteristics of the results:

* *Number of students in the graduating class.* Table 4.3 shows the sizes of our graduating class over the past 6 years.Compared to the following years, 2012 had a fairly small graduating class. (But it was typical of the classes that preceded it.) The following three years had classes that were more than 25% larger. Then in 2016 and 2017, the size of the graduating class more than doubled. (It will double again in 2018!)
* ETS changes the test every 3 to 5 years. The Computer Science test for the 2012-2015 is labelled as 4HMF and the newer test from 2016 is labelled as 4LMF. The national distributions are quite consolidated for the cycle ending 2015 on the 4HMF test. The newer 4LMF test has a smaller concentration of national data and is being updated yearly.
* Although there have been ups and downs, our results have been fairly consistent over the years. The data is thus comparative rather than normative, since the sample of institutions and students varies for each test cycle.
* Our median student placed consistently higher than the median of the national population, which is our stated target.
  + During the most recent testing period of 2016-2017, our student median was at the 56th percentile compared to the National median and 61st percentile compared to the peer group of similar state universities. (See Figure 4.2)
  + A comparative analysis with a group of ten peer institutions (including all six California State University campuses that use the MFT) indicates better results in the most recent 2016-2017 year. (See Figure 4.2, Table 4.4, and Table 4.5.)
* All results exceed the target levels, although there is still considerable room for improvement. During semester conversion, we modified most of our courses and increased the coverage of core topics. With these modifications, we hope the results will improve further.

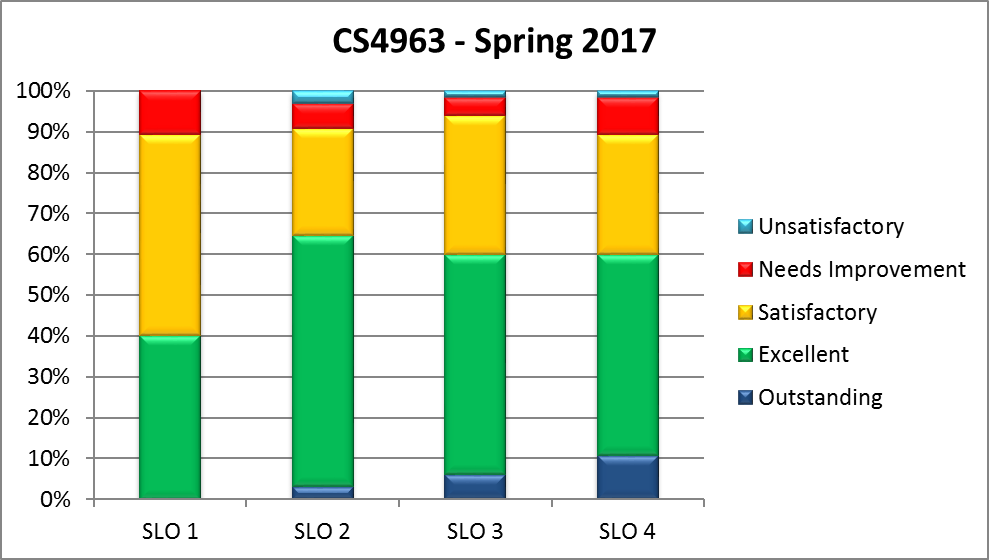
**A.5: Evaluation of CS4963 results.**



**Figure 4.5: CS4963 SLO Evaluation Results**

****

**Figure 4.6: CS4963 SLO Evaluation Results**

****

**Figure 4.7: CS4963 SLO Evaluation Results**

CS 4963 provides a recapitulation of the primary concepts and is the expanded version of CS490 that existed prior to Fall 2016. In CS490 and CS4963, students are assessed based on assignments targeted towards SLO#1 - SLO#4. These evaluations are conducted internally similarly to the external MFT. CS4963 student results are evaluated on a 5-point scale.

* The class average is consistently above 3.0 as shown in Figure 4.5.
* More than 80% of the students have achieved satisfactory performance as indicated in Figure 4.6 and Figure 4.7.
* It is also encouraging to note that student performance was better in Spring 2017 (semester courses) than in Spring 2016 (quarter courses).

**A.6: Evaluation of achievement of SLOs**

Data collected from various measures is indicated in Table 4.2. The data is then analyzed to determine whether the achievement target for each Student Learning Outcome is met.

In many instances the data is presented in a graphical format where the labels on the X-axis refer to the term and year when the data was collected. Labels on the Y-axis refer to the scale that was employed—either a 5 point scale or a 100% scale.

**Student Learning Outcome #1**

*Students will be able to apply concepts and techniques from computing and mathematics to both theoretical and practical problems.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| MATH2110, MATH2120, CS2148 | MATH2550, CS3112, CS3186 | CS4963 |

CS 4963 provides a recapitulation of the primary concepts and students are assessed based on assignments targeted towards this learning outcome. MFT Assessment indicator AI-2 (Discrete Structures and Algorithms) is relevant to this SLO.

**Figure 4.8: Survey Results**

***Summary of Results***

* MFT Assessment Indicator (AI-2) in the area of Discrete Structures and Algorithms. (ETS provides yearly data and aggregate test data.)
  + During the 2012-2017 cycles, our students mean scores were consistently above the 50th percentile (compared to the mean scores of all the institutions) for MFT Assessment Indicator (AI-2). See Figure 4.3 and Table 4.3 for the yearly distribution of scores.
  + A comparative analysis with the national cohort indicates satisfactory results. A comparison with a cohorted group of ten peer institutions (including all six California State University campuses that use the MFT) indicates our median student is placed closer to 60th percentile (See figure 4.4, Table 4.4, Table 4.5)
  + Our students’ mean scores on this MFT assessment indicator improved from 38% (2006-2011, not shown) to better than 46% (2012-2018). (See Table 4.3.)
* The results on the internal assessment in CS490/CS4963 indicate that the student average is around 3.25/5. See Figure 4.5 for the yearly distribution of scores.
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory. (See Figure 4.8)
* All results exceed the target levels. There is still considerable room for improvement. With the modification of CS2148, we hope the results will improve further. We are also considering a new prerequisite to CS2148.

**Student Learning Outcome #2**

*Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| CS2011, CS2012, CS2013 | CS3035, CS3220 | CS4963 |

CS 4963 provides a recapitulation of the primary concepts and students are assessed based on assignments targeted towards this learning outcome. MFT Assessment indicator AI-2 (Programming and Software Engineering) is relevant to this SLO.

**Figure 4.9: Survey Results**

***Summary of Results***

* The results on the MFT Assessment Indicator (AI-1) in the area of Programming and Software Engineering have been encouraging. ETS provided yearly data and 3-year aggregate data.
  + During the 2012-2017 cycles, our students mean score was at or above the 50th percentile compared to the mean scores of all institutions. See Figure 4.3 for the yearly distribution of scores.
  + A comparative analysis with the national cohort and a cohorted group of ten peer institutions (including all six California State University campuses that use the MFT) indicates satisfactory results. (See Figure 4.4)
  + We consider Programming/Software Engineering to be the bread and butter of computer science. We are satisfied that our students’ mean percentiles on this MFT assessment since 2012 averages slightly below the 70th percentile. The most recent two years have been slightly lower, which we are striving to improve.
* The results on the internal assessment in CS490/CS4963 indicate that the student average is around 3.5/5. See Figure 4.5 for the yearly distribution of scores.
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory. (See Figure 4.9)
* All results exceed the target levels. Considering the scores in 2016 and 2017, there is considerable room for improvement. During semester conversion we made a number of changes to improve this outcome . (See Section B).

**Student Learning Outcome #3**

*Students will have a strong foundation in the design, analysis, and application of many types of algorithms.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| CS2011, CS2012, CS2013 | CS2148, CS3112 | CS4963 |

CS 4963 provides a recapitulation of the primary concepts. Students are assessed based on assignments targeted towards this learning outcome. MFT Assessment indicator AI-2 (Discrete Mathematics and Algorithms) is relevant to this SLO.

**Figure 4.10: Survey Results**

***Summary of Results***

* MFT AI-2, (Discrete Mathematics and Algorithms) is discussed under SLO #1.
* The results on the internal assessment in CS490/CS4963 indicate that the student average is around 3.25/5. See Figure 4.5 for the yearly distribution of scores.
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory. (See Figure 4.10)
* All results exceed the target levels. There is still considerable room for improvement.

**Student Learning Outcome #4**

*Students will have a fundamental understanding of computer systems.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| CS2011, CS2012, CS2013 | CS2148, CS3112 | CS4963 |

CS 4963 provides a recapitulation of the primary concepts. Students assessment is based on assignments targeted towards this learning outcome. MFT Assessment indicator AI-3 (Systems, consisting of Architecture, Operating Systems, Networking, and Databases) is relevant to this SLO.

**Figure 4.11: Survey Results**

***Summary of Results***

* The results on AI-3 (Systems, consisting of Architecture, Operating Systems, Networking, and Databases) are encouraging. ETS provided yearly data and 3-year aggregate data.
  + During the 2012-2017 cycles, our students mean score was placed at or above the 75th percentile (compared to the mean scores of all institutions). See Figure 4.3 for the yearly distribution of scores.
  + A comparative analysis with the national cohort and a cohorted group of ten peer institutions (including all six California State University campuses that use the MFT) indicates satisfactory results. (See Figure 4.4)
  + We are satisfied that our students’ mean score on this MFT assessment exceeds the 85th percentile since 2012.
* The results on the internal assessment in CS490/CS4963 indicate that the student average is around 3.25/5. See Figure 4.5 for the yearly distribution of scores.
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory. (See Figure 4.11)
* All results exceed the target levels. There is still considerable room for improvement.

**Student Learning Outcome #5**

*Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| CS2011, CS2012, CS2013 | CS3337 | CS4961, CS4962 |

In both CS3337 and in CS4961/4962 (Senior Design) students develop projects from requirements to implementation. The former is a 1-semester course that uses internally generated projects. The latter is a full academic year in which projects are requested by outside stakeholders who serve as customers for the teams.

In both cases, a team of (typically 5) students works together on a project. A faculty advisor oversees the work but does not participate in any of the actual development.

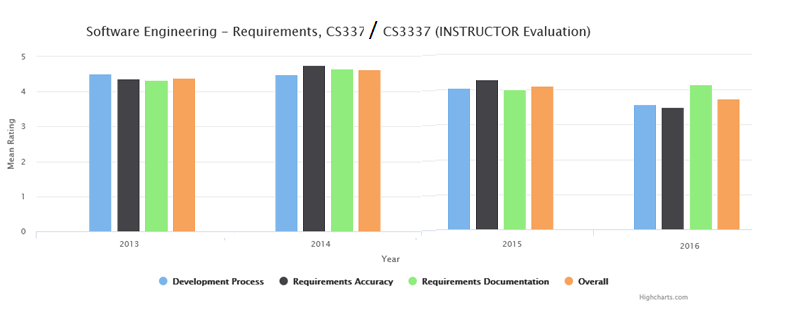
A primary goal of these courses is to achieve SLO #5 and SLO #6. In both cases we use a relaxed waterfall approach: requirements, followed by at least a preliminary design, and then by implementation and testing. The processed is *relaxed* waterfall rather than *rigid* waterfall in that throughout the project customers often realize that their initial problem descriptions were incomplete and that additional requirements are needed. It is up to the teams whether to accept the new requirements or not.

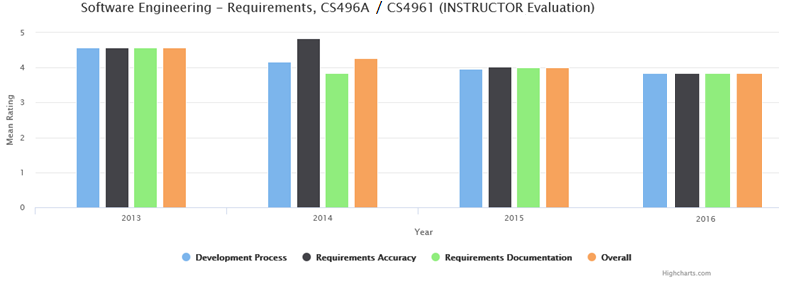
The first portions of CS3337 and CS4961 focus on analyzing the problem as presented in customer terms and developing a clear statement of requirements. As indicated, the requirements document may be updated if a team agrees to new requirements in the course of project. This portion of these projects focusses on SLO #5.

In both CS3337 and CS4961, the groups are required to submit a set of deliverables that includes a high-level requirements document and presentation materials. Students are given sample documents so that they have concrete examples of what is expected.

Project deliverables are graded based on rubrics designed to evaluate how well the teams define the project requirements. The Requirements Rubric is based on three assessment indicators: Development Process, Requirements Accuracy, and Requirements Documentation.

***Summary of Results***

**Figure 4.12: CS337/CS3337 Rubric Evaluation (SE-Requirements)**

**Figure 4.13: CS496A/CS4961 Rubric Evaluation (SE-Requirements)**

**Figure 4.14: Survey Results**

* Students provide detailed requirements specification documents; hence the scores are high on the Software Engineering – Requirements rubric evaluations. (See Figures 4.12 – 4.13).
* Documentation for all senior design projects (CS4961-CS4962) can be found at <https://csns.calstatela.edu/department/cs/projects>.
* Senior design projects have received considerable positive feedback from students, faculty, and participating customers, of whom are members of our Industry Advisory Board.
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.14).
* All results exceed the target levels. We are satisfied that students are achieving SLO #5.
* We continue to seek externally sponsored funded projects to give students a real-world experience.

**Student Learning Outcome #6**

*Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| CS2011, CS2012, CS2013 | CS3337 | CS4961, CS4962 |

This outcome is the focus of the second half of the project sequences: CS3337 and in CS4961/4962 (Senior Design). See the description of these project sequences under **SLO #5** above.

In the second portion of CS3337 and in CS4962, the teams create detailed design documents and implement the project in code.

Although some groups use an agile methodology, all of the groups develop a detailed design document in some form.

The outside stakeholders interact on a regular basis with the teams either in person or via telecom or other communication link. The frequency of these meetings ranges from once a week to once a month. These meetings allow the stakeholder to track the team’s progress and to provide guidance when the team seems to be stuck or heading in the wrong direction.

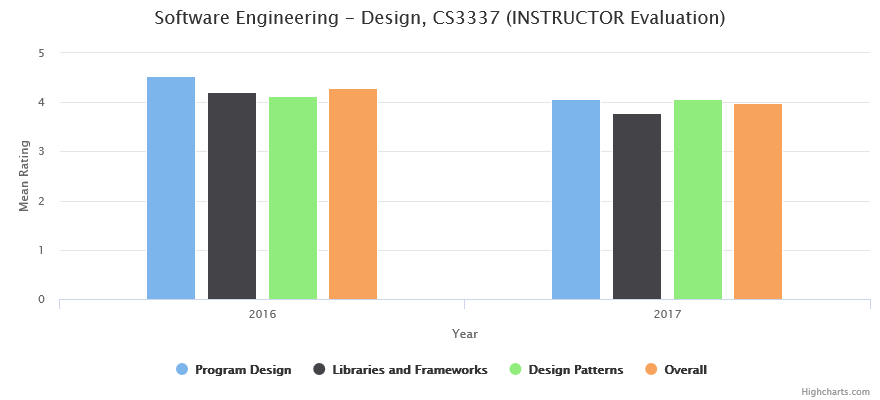
The project deliverables include a project design document and implementation report. As with the requirements, sample documents are provided so that students have concrete examples of what is expected.. Project deliverables are graded based on rubrics considering how well the team completes the project. In addition, peer evaluation is conducted to assess the collaborative nature of all the other members of the group.

The Design rubric is assessed using three assessment indicators: Program Design, Libraries/Frameworks, and Design Patterns.

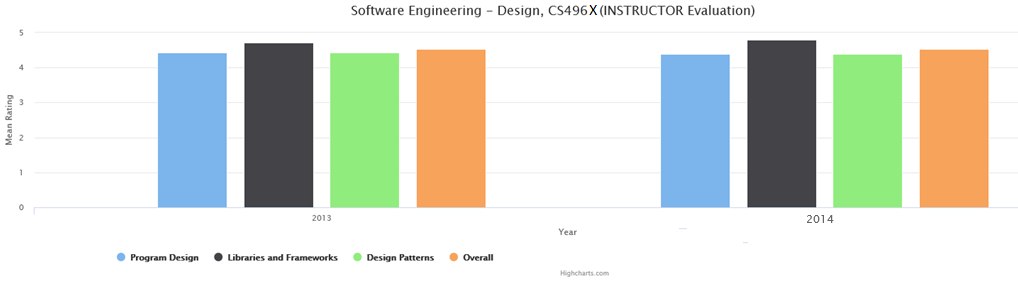
The Implementation rubric is assessed using three assessment indicators: Programming Paradigms, Functions/Methods, and Testing.

The Teamwork rubric is assessed by the peers and the advisor using five assessment indicators: Participation, Problem-solving, Attitude, Contribution, and Interaction.

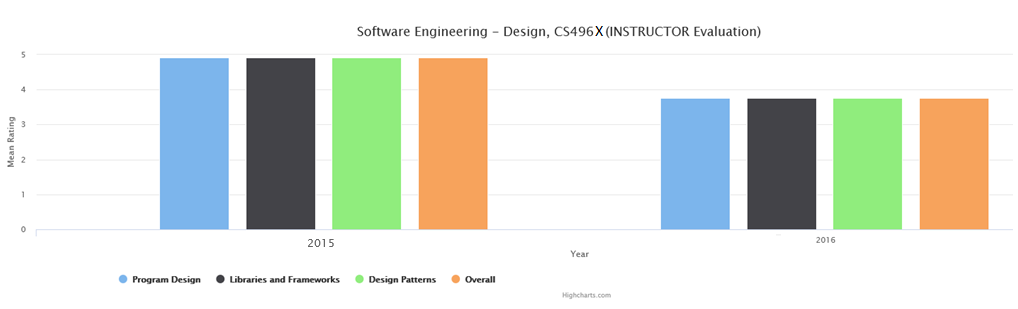
***Summary of Results***



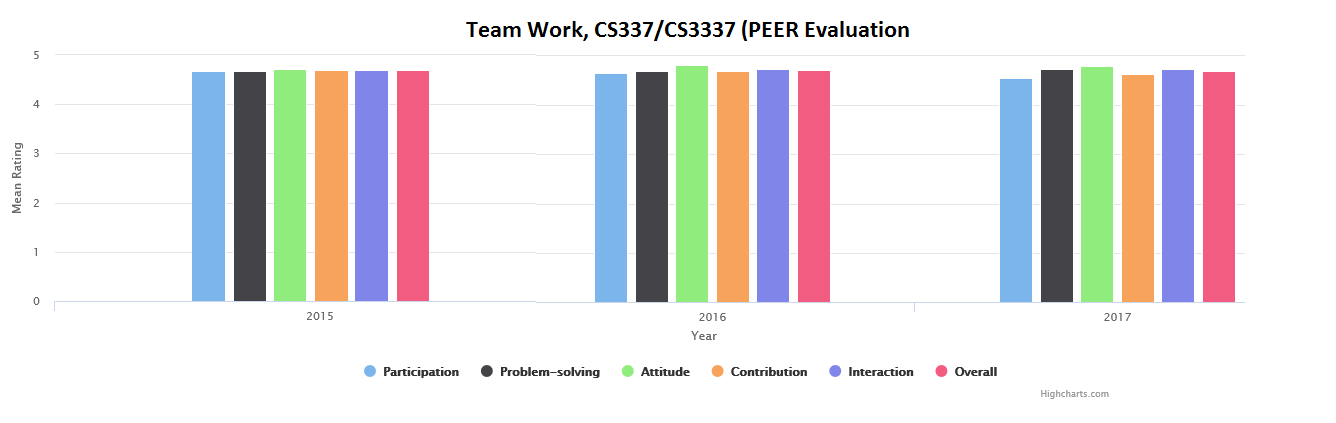
**Figure 4.15: CS3337 Rubric Evaluation (SE-Design)**



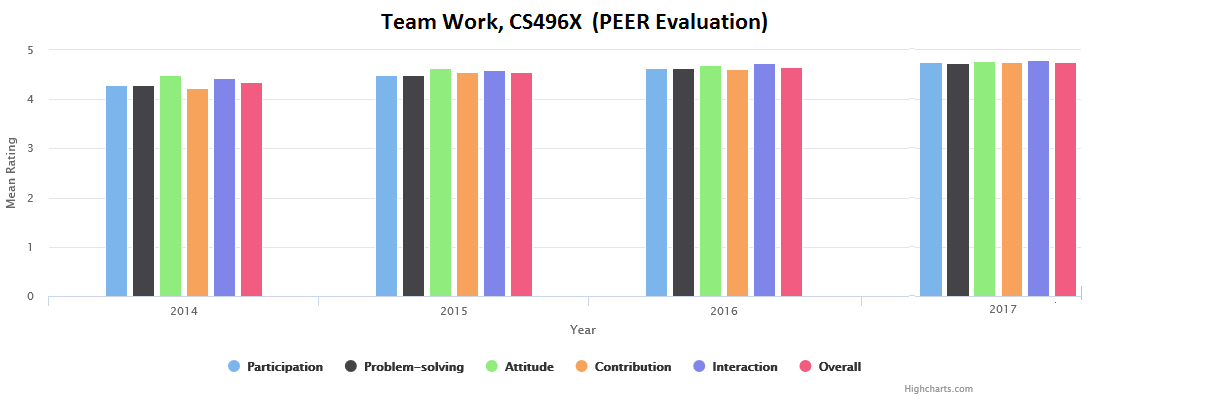
**Figure 4.16: CS496X Rubric Evaluation for 2013-2014 (SE-Design)**

****

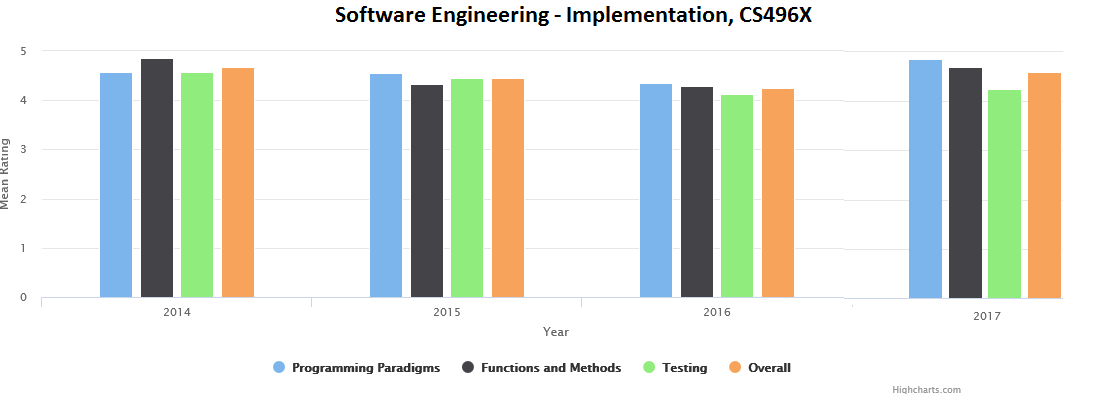
**Figure 4.17: CS496X Rubric Evaluation for 2015-2016 (SE-Design)**

****

**Figure 4.18: CS337/3337 Rubric Evaluation for 2015-2016 (Team Work)**

****

**Figure 4.19: CS496X Rubric Evaluation for 2015-2016 (Team Work)**

****

**Figure 4.20: CS496X Rubric Evaluation (SE Implementation)**

**Figure 4.21: Survey Results**

* Students provide detailed design documents; hence the scores are high on the rubric evaluations. (See Figures 4.15 – 4.17).
* All senior design projects (CS4961-CS4962) available at <https://csns.calstatela.edu/department/cs/projects>.
* Senior design projects have received considerable positive feedback from students, faculty, and participating customers, many of whom are members of our Industry Advisory Board (IAB).
* Evaluations on Design, Implementation, and Teamwork rubrics in CS337/CS3337 and CS491A/CS4961 is satisfactory. (See Figures 4.15 – 4.20).
* Teamwork rubric evaluations are quite high in our project courses. Team members typically form effective working relationships with each other, as well as with their faculty advisors and project sponsors.
* Students, Alumni, Faculty, Employer, and IAB surveys have all been satisfactory (See Figure 4.21).
* All results exceed the target levels. We are satisfied that students are achieving this outcome.
* Our Senior Design projects are currently sponsored by AT&T. Boeing, Aerospace, JPL, Optic Arts, City of Los Angeles and a number of other local organizations. We continue to seek externally sponsored funded projects to give students a real-world experience.

**Student Learning Outcome #7**

*Students will be able to communicate effectively orally and in writing.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

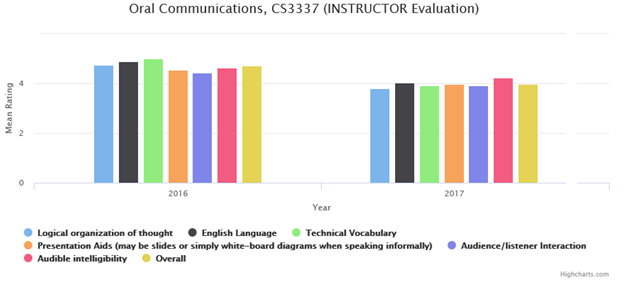
| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| General Education  (Oral Comm., Written Comm.) | ENGL2030, CS3337 | CS4961, CS4962 |

Oral and Written communication are both introduced in early required General Education courses.

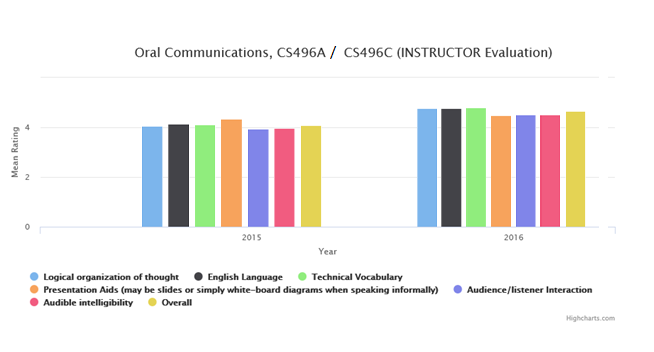
In CS4961-CS4962, every student gives several presentations over the course of the project. Oral communication is assessed during the final presentations in CS3337, CS4961, and CS4962. CS4962 has a culminating event referred to as an Expo. Student teams make formal presentations of their project in front of the Industry Advisory Board and project liaisons.

In CS3337, CS4961, and CS4962, the groups are required to submit presentation slides used in project presentations. Students are evaluated based on Oral Communication rubric with five indicators (Logical organization, English language, Technical vocabulary, Presentation aids, and Audience interaction).

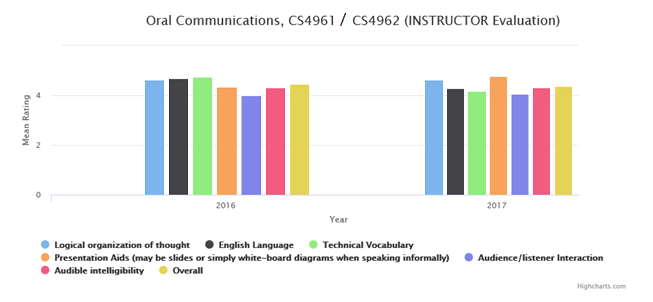
Written communication is assessed through formal written assignments in both CS4961 and CS4962. After a first draft of a document is submitted, students are given feedback. They are required to make the necessary corrections and submit revised versions. Document evaluation is based on the Written Communication rubric. It consists of five assessment indicators: Document Organization, Section Content, Sentence Structure, Technical Vocabulary, and Document Format.

****

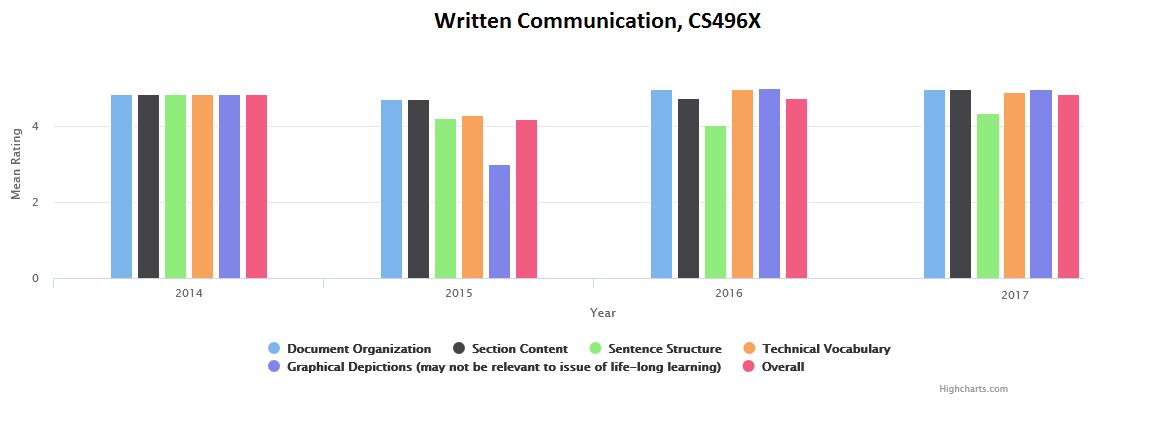
**Figure 4.22: CSCS3337 Oral Communication Rubric Evaluation**

****

**Figure 4.23: CS496A/CS496C Oral Communication Rubric Evaluation**

****

**Figure 4.24: CS4961/CS4962 Oral Communication Rubric Evaluation**



**Figure 4.25: CS496X Written Communication Rubric Evaluation**

**Figure 4.26: Survey Results**

***Summary of results***

* It has been difficult for our students to transition from pure English classes to technical writing. As a result, we have added a technical writing course in the major. We will be working with English department to create sections for our majors focusing on software engineering technical writing.
* One of the assessment indicators in both the Oral Communication and the Written Communication rubric was modified to give better feedback to the students.
* The Written Communication revised rubric evaluations are satisfactory. (See Figure4.24)
* Student teams make a formal presentation of their projects at the end of each semester. Oral Communication is evaluated at each of those presentations. The presentations at the end of the second semester (CS 4962) is a formal college-level event, known as the Senior Design Expo. To help establish its importance, the Expo is held either in the Campus Golden Eagle Ballroom or off campus. Lunch is provided, and the IAB and all project liaisons are invited. Students are encouraged to dress appropriately, e.g., as for a job interview. Students generally receive higher evaluations during the Expo than when compared to their (less formal) presentations at the end of CS4962. (See Figures 4.23 – 4.25)
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.26).

**Student Learning Outcome #8**

*Students will have the knowledge, skills, and attitudes for lifelong self-development.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| CS1010 | CS3801 | CS4961, CS4962, CS4963 |

The program capstone courses—CS4961/CS4962 (Senior Design) and CS4963 (Recapitulation)—encourage students to recognize the importance of Lifelong Learning and the importance of engaging in continuing professional development.

The CS4961-CS4962 projects almost always require students to learn new tools and technologies to be successful. To help students appreciate how much they were forced to learn during their projects, students are required to write a Lifelong Learning essay at the end of CS4962. The essay prompt requires student to discuss the following.

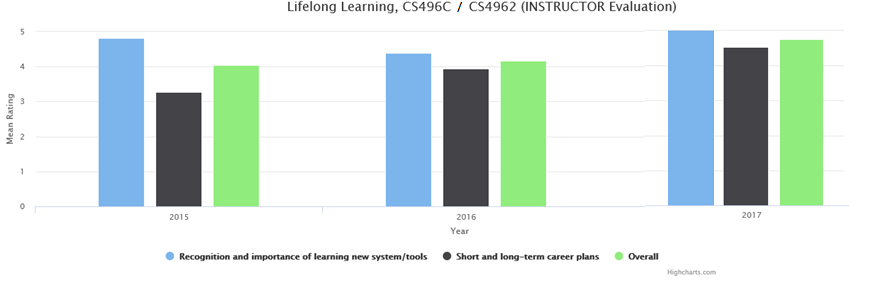
* The technologies they learned: software libraries, frameworks, programming languages, software tools, software systems etc.,)
* How they learned this material: web research, use of online tutorials, software practice, consulting with knowledgeable people, the use of forums and sites like Stack Overflow, etc.,)
* What they learned about learning. They are asked to review what they know about learning new technologies that they didn't know before. They are asked how this experience will influence their approach to learning new technologies in the future.
* Their short-term and long-term career goals are and how they plan to achieve them.

The lifelong learning essay is graded using a rubric of two indicators: Recognition/importance of learning new systems/tools, short/long term career plans.

In CS4963, every student completes a mock technical interview with a group consisting of a faculty member and industry representatives. The goal is to expose students to the technical interview process, given them the opportunity to speak *ad hoc* in front of people who know more than they do, and make interview less anxiety provoking. Students attend workshops and/or review videos to advance their professional skills. As preparation for these interviews the following topics are discussed in class.

* How to communicate in technical interviews and what to do when stumped by a technical question.
* What is ‘Project Work’ and what it looks like on a resume.
* What to expect and what is generally expected of them on their first job.
* How to transition their academic skills to the industry.

***Summary of results***

**

**Figure 4.27: Lifelong Learning Rubric evaluation**

**Figure 4.28: Survey Results**

* The results of the Lifelong Learning Rubric evaluation has been satisfactory (See Figure 4.26).
* The mock interview process in CS4963 provides students with important insights into how to approach problem solving in a face-to-face interaction.
  + In many cases faculty and industry representatives were happy with how the students presented themselves.
  + A few other students needed help to improve their performance.
  + This process was very informal (not mandatory) during the 2015-2016 cycle and established more formally during the 2016-2017 academic year. Google Engineers who participated in both years indicated that students need to have a sound knowledge of Data Structures and to be able to think of problem solving in a language independent way. In our newly introduced 3-semester programming sequence, we have established the CS2013 Data Structures course as language independent. This was echoed by all the faculty and the Industry Advisory Board.
  + We will be developing a formal evaluation rubric for future mock interviews.
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory. (See Figure 4.28).
* By the time they graduate, most students have sufficient understanding the fundamentals of computer science that they can be productive and successful.
* The College of ECST has established a Placement Coordinator position whose occupant will interact with the students, conduct workshops, discuss internship opportunities, and provide placement services.

**Student Learning Outcome #9**

*Students will have the ability to analyze the local and global impact of computing on individuals and society.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| CS1010 | CS3801 | CS4961, CS4962 |

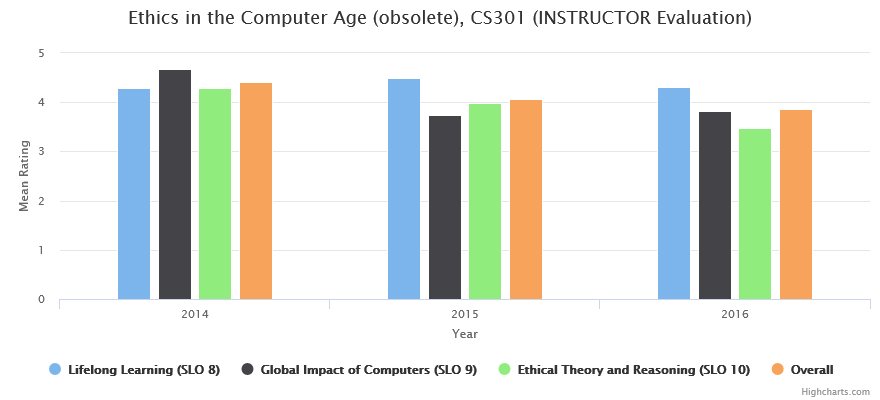
CS 3801 covers various topics on Societal and Ethical issues in Computing including the following.

* The effects of social media on politics
* Cyberbullying
* The effects of automation on employment
* The effects of globally available information
* The effects of automation on consumers, standards of living
* The use and effects of software in law enforcement
* The “Dark Web”
* The effects of hacktivist groups like Anonymous
* Warfare in the cyber age

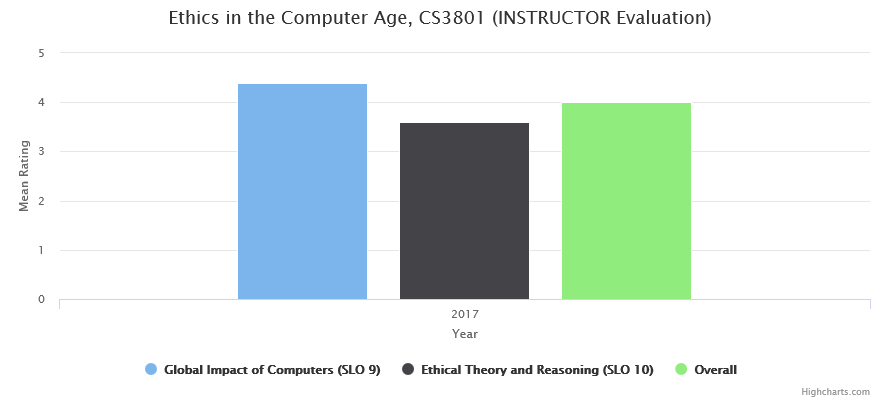
Students are evaluated on the basis of written reports, presentations, and exams. These evaluations are normalized to a 5-point scale and recorded on a rubric with one indicator: Global Impact of Computers.

In addition, students are asked to consider these and related social and ethical issues as they work on their Senior Design projects (CS 4961-CS 4962).

***Summary of results***



**Figure 4.29: CS301 Rubric Evaluation**



**Figure 4.30: CS3801 Rubric Evaluation**

**Figure 4.31: Survey Results**

* During semester conversion CS3801 was expanded from the earlier CS301. The unit coverage was tripled to allow for the increased time needed to cover a broad range of topics. The expanded time allows for more student presentations and deeper discussions throughout the course. We are satisfied that students are achieving this outcome sufficiently.
* CS3801 rubric evaluation by faculty has been satisfactory (See Figure 4.30).
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.31).

**Student Learning Outcome #10**

*Students will have a fundamental understanding of social, professional, ethical, legal, and security issues in computing.*

**Courses in which this SLO is Introduced, Reinforced and Emphasized**

| **Introduce** | **Reinforce** | **Emphasize** |
| --- | --- | --- |
| CS1010 | CS3801 | CS4961, CS4962 |

CS 3801 covers various topics in Societal and Ethical issues in Computing.

* Professional and ethical issues
  + Ethical thinking in general
  + The ACM and IEEE Codes of Ethics
  + The software developer’s professional responsibility regarding cybersecurity
  + Ethical outcomes regarding the effects of computers on people.
  + The possibility of strong AI and questions such whether machines may eventually have rights
  + Privacy and data sharing
  + Autonomous vehicles
* Legal issues
  + Intellectual property law
  + Lawsuits and intellectual property disputes in the tech industry
* Social issues:
  + Cyber age
  + Social media
  + Hacktivism
  + Free/open source software
  + Automation and employment

In CS3801, students are evaluated based on written reports, presentations and exams. These evaluations are normalized on a 5-point scale and denoted on a rubric with one indicator (Ethical Theory and Reasoning).

In addition, students are asked to consider these and related social and ethical issues as they work on their Senior Design projects (CS 4961-CS 4962).

***Summary of results***

**Figure 4.32: Survey Results**

* During semester conversion CS3801 was expanded from the earlier CS301. The unit coverage was tripled to allow for the increased time needed to cover a broad range of topics. The expanded time allows for more student presentations and deeper discussions throughout the course. We are satisfied that students are achieving this outcome sufficiently.
* CS3801 rubric evaluation by faculty has been satisfactory (See Figure 4.30).
* Students, Alumni, Faculty, Employer and IAB surveys have all been satisfactory (See Figure 4.32).

***B: Continuous Improvement***

Continuous improvement process combines an evaluation and analysis of program assessment results with student feedback, faculty analysis, university approval, and Advisory Board review. This information is then integrated in changes to be made to the program.

* *Assessment Committee:* The Assessment Committee ensures that all student outcomes are evaluated at the appropriate points as mentioned in the previous section. The committee then conducts annual evaluation of all student outcomes which are compiled into an Assessment Report. Typically, the Assessment Reports are published on the web as a two year report. All reports are accessible to ABET evaluators by logging on to the CSNS server at http://csns.calstatela.edu/wiki/content/assessment/documents/.
  + Assessment Report, 2016-2018
  + Assessment Report, 2014-2016
  + Assessment Report, 2012-2014
  + Assessment Report, 2010-2012
* *Faculty Retreat:* This is an annual event as described at <https://csns.calstatela.edu/wiki/content/department/cs/assessment/retreat_presentations/>.
  + The assessment coordinator and the department chair make a presentation detailing the SLO evaluations.
  + The faculty retreats prove a forum for faculty interactions and consider the results of the assessment process. In addition, the Quarter to Semester conversion provided a one-time excellent opportunity for the faculty to re-examine, strengthen and repackage our courses.
  + Any curricular changes that have been proposed were approved by the faculty.
* *Industry* Advisory *Board (IAB) meeting:* This is an annual event as described at <https://csns.calstatela.edu/wiki/content/department/cs/assessment/iab/> .
  + The assessment coordinator and the department chair make a presentation during the annual Industry Advisory Board (IAB) meeting detailing the curriculum and SLO evaluations.
  + Some of the IAB meetings are followed by Senior Design Project presentations which give a better perspective of our student skills.
  + Input is solicited from IAB membership.

After analyzing the data collected and as a result of the annual/biannual reviews, we made a number of curricular changes that include course modification, general education modification, course teaching pedagogies, and assessment processes. Significant changes are described below:

**2012-2014**

* ***PHYSICS sequence***

The program requirements had an option of choosing either the PHYS101-103 (shorter non-calculus based sequence) or PHYS 211-213 (a calculus based sequence used by all Engineering majors). To get a thorough science background, we removed the PHYS101-103 option. Our Industry Advisory Board and the department faculty have advocated this change.

* ***CS301 course modification***

CS301 was designed to just cover the ethical issues in computing. It did not provide enough content to cover the broader societal, legal and security issues in computing. The course was modified and increased by 1-unit to provide for the additional coverage. This change has significantly added to the importance of SLO #10.

* ***Senior Design Projects***

Externally sponsored senior design projects in CS496 have received considerable positive feedback. However, it was difficult to get all the projects externally sponsored. We have made a concerted effort to reach out to secure more external projects including some community partners. In addition, we reached out internally within the university to define projects that improve their current process. At the program level, all the projects are sponsored by some external entity and give students a real-world experience,

* ***Assessment tools on CSNS***

Prior to 2012, we employed a summative assessment of skills that were determined by the faculty. This practice was used to assess all the SLOs. This process 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. This is still a cumbersome process without an automated evaluation process. New assessment tools have been developed on CSNS by which rubrics can be created and evaluated by the instructors.

**2016-2017**

* ***Created CS2148, a new CS course, replacing MATH248***

The MFT scores described under SLO#1 and SLO#3 had a lot of scope for potential improvement. The class average on the MFT Assessment Indicator was less than 50% for an extended period of time signifying that some changes needed to be done in this area. The MATH248 (Discrete Mathematics) offered by the MATH department did not adequately serve as a prerequisite for CS312 (Analysis of Algorithms). We determined that we teach the Discrete Mathematics in our own department. The topics in discrete mathematics are covered in a way that make them more suitable for Computer Science majors. A new CS2148 was created in Fall 2016 which will provide a stronger foundation for CS3112. This modification is intended to improve student performance in SLO #1 and SLO #3.

* ***Developed CS2011, CS3012, CS2013 as a three semester programming sequence***

Programming is considered to be the bread and butter of computer science. We are satisfied that our students’ mean percentiles on this MFT assessment (SLO #2) averages slightly below the 70th percentile. The class average on the MFT Assessment Indicator is around 56% which can be improved. As a result, we decided to convert the 3-quarter programming sequence, (CS201, CS202, CS203) into an expanded 3-semester programming sequence. We expanded it to a three semester sequence. The sequence will contribute to the success of many important skills:

1. Data Structures is one of the important areas of Computer Science as it equips students with the fundamental building blocks for the development of complex systems. Creating CS2013 as third semester Data Structures course will give students a stronger foundation for competent programming and improve student performance in SLO#2.
2. The mock interview process results under SLO#8 was a revelation to both students and faculty. Google Engineers who participated indicated that students need to have a sound knowledge of Data Structures and to be able to think of problem solving in a language independent way. In our newly introduced 3-semester programming sequence, our CS2011 and CS2012 are taught in Java; CS2013 is language independent. This should improve student performance in SLO #8.
3. This change has mutual benefits to improve the student performance in SLO#2 and SLO#6.

* ***Increased the number of units in the Senior Design sequence from 3 quarter courses of 2 units each (a total of 6 quarter units, which is equivalent to 4 semester units) to 2 semester courses of 3 units each (a total of 6 semester units)***

The Senior Design program sequence is a signature event for the students where knowledge gained in the classroom is applied to a major design project. Senior design projects have contributed to the student success in SLO#5, SLO#6, SLO#7 and SLO#8. These projects received considerable positive feedback from students, faculty, and participating customers, of whom are members of our Industry Advisory Board (IAB). The number of sponsored projects and the variety of projects have increased over the years.

Faculty and Students have long held the belief that 2-unit courses does not justify the time commitment required towards these senior projects. During the semester conversion, the senior design sequence has been expanded to 3 units each semester. This is a 50% increase in terms of units. The CS4961 & CS4862 sequence have now added many new assignments and activities that will contribute to the success of many important skills:

1. Enhance student experience in solving real-world problems of interest to industry and help prepare them for careers in engineering. (SLO#5, SLO#6, SLO#8).
2. Enhance teamwork skills (SLO#6)
3. Enhance written, oral and presentation skills (SLO#7)

* ***Increased the number of units in the capstone recapitulation course when converting CS490 (2 quarter units) to CS4963 (3 semester units)***

We initially created CS490 as a 2-unit quarter course to ensure that students synthesize the knowledge acquired throughout the undergraduate curriculum. It was a revelation to both students and faculty that there is so much value gained by such a course. The course was divided into four parts: Theory, Programming, Algorithms, and Systems. Each of these topics covered a wide area of courses and allowed the students to appreciate the knowledge and skills gained throughout the curriculum.

During the semester conversion, we decided to expand this course to also include professional development activities. One such activity was to have every student complete a mock technical interview with a group consisting of a faculty member and industry representatives. We partnered with Google where the emphasis of the mock interview was on problem solving. Students were exposed to the technical interview process, given them the opportunity to speak ad hoc in front of people who know more than they do, and make interview less anxiety provoking.

CS4963 was thus expanded to a regular 3-unit semester course and will contribute to the success of many important skills:

1. Students will review, discuss, and present topics from computing and mathematics.
2. Students will have the opportunity to discuss their career goals (short-term and long-term) and to engage in mock interviews.
3. Students will improve both their oral communication and problem solving skills.

In particular, CS4963 aims to enhance the student performance relating to student outcomes SLO#1, SLO#2, SLO#3, SLO#4, SLO#7 and SLO#8.

* ***Added ENGL2030 as a required course***

It has been difficult for our students to transition from pure English classes to technical writing. Faculty evaluation of writing assignments indicated that students lacked the technical writing skills. We have worked with the English department to create ENGL2030 course to address this issue that is more focused on majors from our College. ENGL2030 will replace the second composition requirement in the General Education requirements starting Fall 2016. We will soon initiate a dialogue with the English department to create sections for our Computer Science majors focusing on software engineering technical writing. This modification will improve student performance in SLO #5 and SLO #7.

* ***Expanded CS3801 to be a 3-unit version of the 2-unit CS301 quarter course***

In 2005, We incorporated ethics and professional responsibilities into couple of our classes, specifically into the Software Engineering and Senior Design courses. We soon realized that the students were not paying enough attention to these topics as evidenced by the student surveys for SLO#9 and SLO#10. We did a survey across other campuses and noticed that the trend was to create a computer ethics course within the department. We then created a 2-unit quarter course. Even though the student achievement was satisfactory for SLO#9 and SLO#10, we decided to convert the 2-unit CS301 course into a 3-unit CS3801 course which is an increase of more than 50% in unit coverage. We did this for two reasons:

1. CS3801 covers an increasingly wide range of topics in Societal and Ethical issues in Computing. Students now have more time to explore these topics and to make presentations about them.
2. Faculty have felt that students will devote the necessary time if it is scheduled as a regular 3-unit course rather than an abridged 2-unit course.

This modification will improve student performance in SLO #9 and SLO #10.

* ***Expanded CS1010 to be a 3-unit semester version of the 2-unit CS101 quarter course***

CS1010 is required of all incoming freshman students that satisfies the LIFELONG UNDERSTANDING and SELF-DEVELOPMENT area in General Education. Many of these students do not have any programming or technology background. A large percentage are not ready to take Calculus I. CS1010 provides an overview of computer science and information technology and on careers in these fields. In addition, it covers some elementary programming concepts in an easy to use web interface.

CS1010 is now a prerequisite CS2011. This addresses the long-felt faculty concern that CS2011 poses unsurmountable hurdles to some students as their first technology course. With CS1010 as background CS2011 students are better prepared start learning topics in software development.

This modification will improve student retention rate and student performance in SLO #2 and SLO #8. Anecdotal feedback from CS2011 instructors indicate that CS1010 has already made a positive impact when offered in 2016.

* ***Maintained six elective courses in the program***

The core program requirements provide a solid foundation in computer science. The elective requirement gives students the opportunity to take advanced courses in a number of areas. We had retained the number of elective courses even after the conversion process. Typically, the number of elective courses must be reduced by two courses when converting from a quarter to a semester system. However, by double counting CS1010 and CS3801 as general education, we were able to maintain the required number of electives that can be chosen by the students.

Choosing a wider number of electives should enable students to customize and broaden the educational experience based on their own interests. It is becoming increasingly apparent that there are a number of advanced/specialized course topics (such as Data Science, Web Development, Mobile Development, Artificial Intelligence, Security, Graphics, Multimedia, Gaming, Programming) outside of the main required courses that are very useful for the students to complete the senior design projects and generally useful in their academic careers.

This modification will strengthen all the student learning outcomes SLO#1 thru SLO#10.

* ***Modified most lower division and some upper division courses to fit the Project-based learning model***

Under this project-based learning model 3-unit courses have built-in 1-unit (3-hour) labs. This concept was adopted for several courses (CS1010, CS2011, CS2012, CS2013, CS3035, CS3112, CS3220, CS3337, CS3801, CS4961, CS4962, and CS4963). The labs provide additional face-to-face contact with students and incorporate project work into the courses. Project work helps students become better collaborators, critical thinkers, public speakers, and communicators— vital to success at CSULA and attainment of skills outlined by all the student outcomes

This project-based learning model provides 5 contact hours per week instead of the traditional 3 contact hours per week.

This modification will strengthen all the student learning outcomes SLO#1 thru SLO#10.

* ***Integrate Upper Division Natural Science, Humanities and Social Science General Education outcomes in Computer Science curriculum***

The program Student Learning Outcomes are closely tied to the General Education Learning Outcomes in Natural Science, Humanities and Social Science; in particular, students must (a) establish their self-identity in their profession, (b) be able to analyze the interrelationship between the self and the social / political / cultural environments, and (c) be able to make ethical decisions when designing systems. To address the challenge of unit reduction, we proposed to integrate upper division General Education outcomes in various required Computer Science courses. The goal is to provide a significant integrated learning experience that will (a) achieve both GE and major learning outcomes and (b) help develop our students into professionals whose humanistic, social, and technical knowledge, skills, and sensibilities will equip them to function in a rapidly changing world.

* ***Built CSNS into an Assessment Management System that works alongside a Learning Management System***

This integration of assessment and learning management simplifies and streamlines the assessment process. CSNS now provides the following tools:

1. Course-level assessment tools such as rubrics and course journals.
2. General-purposed tools that can also be used for program assessment, e.g. surveys, a file manager, a wiki, and mailing lists.
3. The ability to Import data from such as Major Field Tests (MFT) results and then integrate that data with data collected in-house.
4. Tools for data visualization.

Building assessment into a system that is used on a daily basis encourages and facilitates a continuous and sustainable assessment process.

***C. Additional Information***

**C-1 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/Faculty Comments:***

….

***Alumni Comments:***

….

# 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 (CS1000 & CS2000 level) required courses, the curriculum provides students with the basic mathematical and science framework. Through its upper division (CS3000 & CS4000 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.

**A-1 Course requirements**

The requirements for the major are described below:

**Requirements for the Major (93 units)**

A grade of "C" or better is required for all prerequisite courses in the major.

**Lower Division Required Courses (39 units)**

CS 1222 Introduction to Relational Databases(3)

CS 2011 Introduction to Programming I (3)

CS 2012 Introduction to Programming II (3)

CS 2013 Programming with Data Structures (3)

CS 2148 Discrete Structures (3)

ENGL 2030 Introduction to Technical Writing (3)

MATH 2110 Calculus I (4)

MATH 2120 Calculus II (4)

MATH 2550 Introduction to Linear Algebra (3)

PHYS 2100 General Physics I (5)

PHYS 2200 General Physics II (5)

**Upper Division Required Courses (33 units)**

CS 3035 Programming Language Paradigms (3)

CS 3112 Analysis of Algorithms (3)

CS 3186 Introduction to Automata Theory (3)

CS 3220 Web and Internet Programming (3)

CS 3337 Software Engineering (3)

CS 3801 Societal and Ethical issues in Computing (3)

EE 3445 Computer Organization (3)

CS 4440 Introduction to Operating Systems (3)

CS 4961 Software Design Laboratory I (3)

CS 4962 Software Design Laboratory II (3)

CS 4963 Computer Science Recapitulation (3)

**Electives (21 units)**

**Mathematics Electives (3 units)**

Select 3 units of lower division or upper division course(s) in the Mathematics area with prior approval of the Computer Science undergraduate adviser.

**Computer Science Electives (18 units)**

Select 15 units of upper division Computer Science (CS3xxx/CS4xxx) courses.

Table 5-1 describes the suggested plan of study (by year and semester term) for students in the computer science program. Many of the above courses have a built in laboratory component. All courses are delivered using a formula where one unit means one hour of lecture or three hours of laboratory per week.

Program Name: Computer Science

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course  (Department, Number, Title)  List all courses in the program by term starting with first term of the first year and ending with the last term of the final year. | | | Indicate Whether Course is Required, Elective or a Selected Elective by an R, an E or an SE.1 | *Subject Area (Credit Hours)* | | | | | | Last Two Terms the Course was Offered:  Year and,  Semester, or  Quarter | Average Section Enrollment  for the Last Two Terms the Course was Offered2 |
| Math & Sciences | Computing Topics  Mark with an F or A for Fundamental or Advanced | | General Education | | Other |
| **1st Semester** | | |  |  |  | |  |  | |  |  |
| MATH 2100 Calculus I | | | R | 4 |  | |  |  | |  |  |
| ENGL 1010 Written Communication | | | R |  |  | | 3 |  | |  |  |
| COMM1500 Oral Communication | | | R |  |  | | 3 |  | |  |  |
| GE B2/B3 (Biological or Interdisciplinary Physical/Biological Science Elective) | | | R | 3 |  | |  |  | |  |  |
| CS 1010 Introduction to Higher Education for Computer Science Majors | | | R |  |  | | 3 |  | |  |  |
| **2nd Semester** | | |  |  |  | |  |  | |  |  |
| MATH 2200 Calculus II | | | R | 4 |  | |  |  | |  |  |
| HIST 2020 US History | | | R |  |  | | 3 |  | |  |  |
| POLS 1500 Gov. & Amer. Soc. | | | R |  |  | | 3 |  | |  |  |
| CS 1222 Introduction to Relational Databases | | | R |  | 3(F) | |  |  | |  |  |
| CS 2011 Introduction to Programming I | | | R |  | 3(F) | |  |  | |  |  |
| **3rd Semester** | | |  |  |  | |  |  | |  |  |
| MATH Elective | | | SE | 3 |  | |  |  | |  |  |
| MATH 2550 Introduction to Linear Algebra | | | R | 3 |  | |  |  | |  |  |
| PHYS 2100 General Physics I | | | R | 5 |  | |  |  | |  |  |
| CS 2012 Introduction to Programming II | | | R |  | 3(F) | |  |  | |  |  |
| GE C1 (Humanities Elective) | | | R |  |  | | 3 |  | |  |  |
| **4th Semester** | | |  |  |  | |  |  | |  |  |
| GE D1 (Social Science Elective) | | | R |  |  | | 3 |  | |  |  |
| ENGL 2030 Introduction to Technical Writing | | | R |  |  | | 3 |  | |  |  |
| PHYS 2200 General Physics II | | | R | 5 |  | |  |  | |  |  |
| CS 2013 Programming with Data Structures | | | R |  | 3(F) | |  |  | |  |  |
| CS 2148 Discrete Structures | | | R | 3 |  | |  |  | |  |  |
| **5th Semester** | | |  |  |  | |  |  | |  |  |
| CS 3035 Programming Language Paradigms | | | R |  | 3(F) | |  |  | |  |  |
| CS 3112 Analysis of Algorithms | | | R |  | 3(F) | |  |  | |  |  |
| CS 3220 Web and Internet Programming | | | R |  | 3(A) | |  |  | |  |  |
| CS 3337 Software Engineering | | | R |  | 3(F) | |  |  | |  |  |
| EE 3445 Computer Organization | | | R |  | 3(A) | |  |  | |  |  |
| **6th Semester** | | |  |  |  | |  |  | |  |  |
| CS 3186 Introduction to Automata Theory | | | R |  | 3(A) | |  |  | |  |  |
| GE D2 (Social Science Elective) | | | R |  |  | | 3 |  | |  |  |
| CS 3801 Societal and Ethical issues in Computing | | | R |  | 3(A) | |  |  | |  |  |
| CS 4440 Introduction to Operating Systems | | | R |  | 3(A) | |  |  | |  |  |
| CS Elective | | | SE |  | 3(A) | |  |  | |  |  |
| **7th Semester** | | |  |  |  | |  |  | |  |  |
| CS 4961 Software Design Laboratory I | | | R |  | 3(A) | |  |  | |  |  |
| CS Elective | | | SE |  | 3(A) | |  |  | |  |  |
| CS Elective | | | SE |  | 3(A) | |  |  | |  |  |
| CS Elective | | | SE |  | 3(A) | |  |  | |  |  |
| **8th Semester** | | |  |  |  | |  |  | |  |  |
| CS 4962 Software Design Laboratory II | | | R |  | 3(A) | |  |  | |  |  |
| CS 4963 Computer Science Recapitulation | | | R |  | 3(A) | |  |  | |  |  |
| CS Elective | | | SE |  | 3(A) | |  |  | |  |  |
| CS Elective | | | SE |  | 3(A) | |  |  | |  |  |
|  | | |  |  |  | |  |  | |  |  |
| *Add rows as needed to show all courses in the curriculum.* | | | | | | | |  | |  |  |
| TOTALS-ABET BASIC-LEVEL REQUIREMENTS | | | | 30 | 21 (F)  42 (A) | 27 | |  | |  |  |
| OVERALL TOTAL CREDIT HOURS FOR COMPLETION OF PROGRAM | 120 |  | | | | | | | | | |

1. **Required** courses are required of all students in the program, **elective** courses (often referred to as open or free electives) are optional for students, and **selected elective** courses are those for which students must take one or more courses from a specified group.
2. For courses that include multiple elements (lecture, laboratory, recitation, etc.), indicate the maximum enrollment in each element. For selected elective courses, indicate the maximum enrollment for each option.

Instructional materials and student work verifying compliance with ABET criteria for the categories indicated above will be required during the campus visit.

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

The requirements indicated above translate to a road map as indicated in Table 5.2. This ideal roadmap provides a pathway for students to complete their undergraduate requirements in four years.

|  |  |  |  |
| --- | --- | --- | --- |
| **year**  **1** | **summer** | **fall** | **spring** |
|  | MATH 2110 | MATH 2120 |
|  | ENGL 1010 | HIST 2020 |
|  | COMM 1100 | POLS 1000 |
|  | GE B2/B3 | CS 1222 |
|  | CS 1010 | CS 2011 |

|  |  |  |  |
| --- | --- | --- | --- |
| **year**  **2** | **summer** | **fall** | **spring** |
|  | MATH Elective | GE D1 |
|  | MATH 2550 | ENGL 2030 |
|  | PHYS 2100 | PHYS 2200 |
|  | CS 2012 | CS 2013 |
|  | GE C1 | CS 2148 |

|  |  |  |  |
| --- | --- | --- | --- |
| **year**  **3** | **summer** | **fall** | **spring** |
| WPE | CS 3035 | CS 3186 |
|  | CS 3112 | CS 3801 |
|  | CS 3220 | CS 4440 |
|  | CS 3337 | CS Elective |
|  | EE 3445 | GE D2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **year**  **4** | **summer** | **fall** | **spring** |
|  | CS 4961 | CS 4962 |
|  | CS Elective | CS 4963 |
|  | CS Elective | CS Elective |
|  | CS Elective | CS Elective |
|  |  |  |

**Table 5.2: Road map (Ideal)**

For a variety of reasons many students formulate their own schedules with varying loads in any given term. Depending on the summer schedules, students can elect to spread the load over the Summer term as well. 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 (CS3000/CS4000) courses as electives. Therefore, the roadmaps described in Table 5.2 should be used as an ideal guide.

It is essential that every student should see a faculty academic advisor and complete an individualized roadmap as indicated in Student Handbook posted on the department website at <http://www.calstatela.edu/ecst/cs/undergraduate>. We encourage the students to keep their roadmap updated if any situation changes down the road.

**A-2 Curriculum alignment with the Program Educational Objectives**

Our undergraduate curriculum aligns with the three Program Educational Objectives as described in Table 5.3.

|  | **PEO #1** | **PEO #2** | **PEO #3** |
| --- | --- | --- | --- |
| **CS 1010** | **X** | **X** | **X** |
| **CS 1222** | **X** | **X** |  |
| **CS 2011** | **X** | **X** |  |
| **CS 2012** | **X** | **X** |  |
| **CS 2013** | **X** | **X** |  |
| **ENGL 2030** | **X** | **X** |  |
| **CS 2148** | **X** | **X** |  |
| **CS 3035** | **X** | **X** |  |
| **CS 3112** | **X** | **X** |  |
| **CS 3186** | **X** | **X** |  |
| **CS 3220** | **X** | **X** |  |
| **CS 3337** | **X** | **X** |  |
| **EE 3445** | **X** | **X** |  |
| **CS 3801** | **X** | **X** | **X** |
| **CS 4440** | **X** | **X** |  |
| **CS 4961** | **X** | **X** | **X** |
| **CS 4962** | **X** | **X** | **X** |
| **CS 4963** | **X** | **X** | **X** |

**Table 5.3: Core classes that support the Program Educational Objectives**

**Program Educational objective #1**: *Students who had entered 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.*

These courses are largely concerned with the development of the theoretical understanding and skills that are necessary to succeed as software professional. Thus, students that successfully complete these courses should also find success in the workforce.

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

These courses are largely concerned with the development of the theoretical understanding and skills that are necessary to succeed as a graduate student.

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

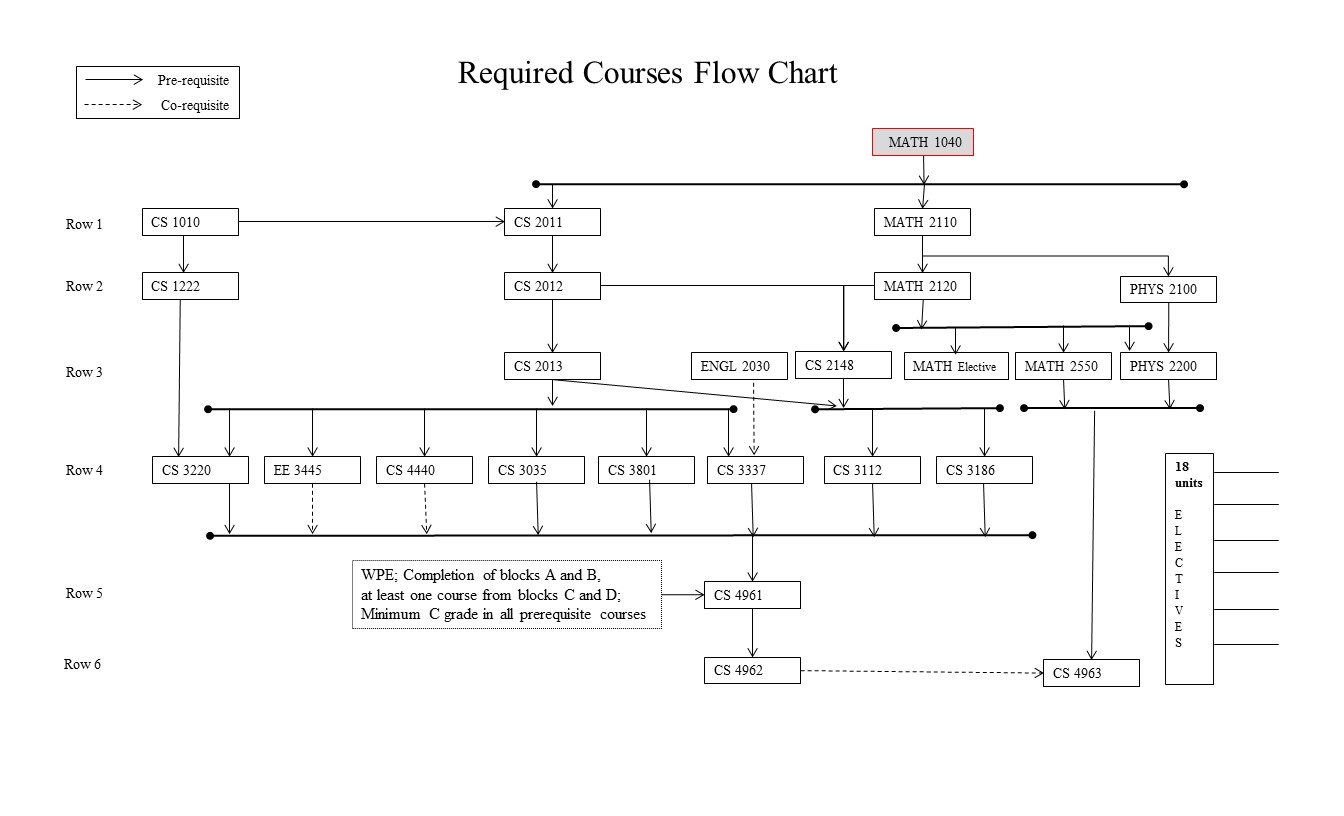
These courses are largely concerned with the abilities to be successful throughout their career. Successful completion of these courses provides our graduates with the necessary views of the world that will allow them to adapt to diverse environments.

**A-3 Course prerequisite(s) requirements**

All major requirements have a designated set of prerequisites. Many courses have an additional designation of a grade of "C" or better in all prerequisite coursework. Prerequisites are strictly enforced on the GET system at the time of registration.

The prerequisite flowchart for the B.S. in Computer Science curriculum is depicted in Figure 5.1 below. The courses are structured with the lower division requirements shown in Rows 1 – 3. The advanced required CS courses are shown in Rows 4 – 6. Elective courses should be scheduled between Rows 4 – 6.

As a general rule, students are advised to finish the courses in a particular row before progressing to the next row and follow the graduation roadmaps described in the next section.



**Figure 5.1: Prerequisite Flow Chart**

**A-4 Curriculum attainment of Student Learning Outcomes**

Table 4.1 indicated all the required Computer Science courses that contribute to the achievement of the ten SLOs. Courses in which SLOs are either introduced, reinforced or emphasized are marked by the letter “X”. In some of the courses marked by “X”, assessment data is evaluated for the achievement of SLOs as described in Section 4.A.6.

Without including any of the technical electives, general education classes, science classes, and math classes, all of the outcomes are covered in the required CS and EE courses. Here are the number of times each outcome is covered in just the CS and EE courses.

SLO #1 : 4 courses

SLO #2 : 8 courses

SLO #3 : 6 courses

SLO #4 : 5 courses

SLO #5 : 6 courses

SLO #6 : 6 courses

SLO #7 : 5 courses

SLO #8 : 5 courses

SLO #9 : 4 courses

SLO #10: 4 courses

Based on the required classes, we are covering all of the outcomes in multiple classes that students are required to take during the program. If we consider technical electives and supplementary classes, such as general education, science, and math classes, we cover all of the outcomes even more. We feel that we are sufficiently covering all of the student outcomes, and this coverage gives us the ability to assess the student outcomes in multiple classes.

**A-5 Course coverage of general criteria**

The ABET CAC section on curriculum states, *“The program’s requirements must be consistent with its program educational objectives and designed in such a way that each of the student outcomes can be attained. The curriculum must combine technical and professional requirements with general education requirements and electives to prepare students for a professional career and further study in the computing discipline associated with the program, and for functioning in modern society”.*

To show that our program meets all of the criteria described, we have taken each sentence and explained how our program attains the requirements below.

* *“The program’s requirements must be consistent with its program educational objectives and designed in such a way that each of the student outcomes can be attained.”* 
  + Section 5.A.2 above explains how the curriculum is consistent with the program educational objectives. Section 5.A.3 above explains how the student outcomes can be attained.
* *“The curriculum must combine technical and professional requirements with general education requirements and electives to prepare students for a professional career and further study in the computing discipline associated with the program, and for functioning in modern society.”* 
  + The computer science curriculum has technical requirements, including math, computer science, and electrical engineering classes. There are also professional requirements including writing, general education, and capstone courses. Our senior design capstone courses (CS4961-CS4962) have a professional component of working with a stakeholder and learning professional etiquette. CS 4963 encourages students to recognize the importance of Lifelong Learning and the ability to engage in continuing professional development.
  + We offer a wider number of electives that should enable students to customize and broaden the educational experience based on their own interests. It is becoming increasingly apparent that there are a number of advanced/specialized course topics (such as Data Science, Web Development, Mobile Development, Artificial Intelligence, Security, Graphics, Multimedia, Gaming, Programming) outside of the main required courses that are very useful for the students to complete the senior design projects and generally useful in their academic careers.
* *“The technical and professional requirements must include at least one year of up-to-date coverage of fundamental and advanced topics in the computing discipline associated with the program.”* 
  + As described in Table 5.1, our curriculum requires 24 units of fundamental computer science courses and 42 units of advanced computer science courses.
* *“The program must include mathematics appropriate to the discipline beyond the pre-calculus level.”* 
  + Beyond the pre-calculus level, we require the following courses.
    - MATH 2110 Calculus I (4)
    - MATH 2120 Calculus II (4)
    - MATH 2550 Introduction to Linear Algebra (3)
    - Lower division or upper division elective course in the Mathematics area (3)
    - CS2148 Discrete Structures (4)
* *“For each course in the major required of all students, its content, expected performance criteria, and place in the overall program of study must be published.”* 
  + The Computer Science curriculum, including all of the required classes, is published in the university catalog (http://ecatalog.calstatela.edu/preview\_program.php?catoid=22&poid=9093) and also on the Computer Science web site (http://www.calstatela.edu/ecst/cs/undergraduate)

**A-6 Course coverage of program criteria**

The ABET CAC section on curriculum states, *Students must have the following amounts of course work or equivalent educational experience:*

*a. Computer science: One and one-third years that must include:*

* + 1. *Coverage of the fundamentals of algorithms, data structures, software design, concepts of programming languages and computer organization and architecture.*
    2. *An exposure to a variety of programming languages and systems.*
    3. *Proficiency in at least one higher-level language.*
    4. *Advanced course work that builds on the fundamental course work to provide depth.*

*b. One year of science and mathematics:*

* + 1. *Mathematics: At least one half year that must include discrete mathematics. The additional mathematics might consist of courses in areas such as calculus, linear algebra, numerical methods, probability, statistics, number theory, geometry, or symbolic logic.*
    2. *Science: A science component that develops an understanding of the scientific method and provides students with an opportunity to experience this mode of inquiry in courses for science or engineering majors that provide some exposure to laboratory work.*

To show that our program meets all of the criteria described, we have taken each sentence and explained how our program attains the requirements below.

*a. Computer science: One and one-third years that must include:*

* + 1. *Coverage of the fundamentals of algorithms, data structures, software design, concepts of programming languages and computer organization and architecture.*

CS 1222 Introduction to Relational Databases(3)

CS 2011 Introduction to Programming I (3)

CS 2012 Introduction to Programming II (3)

CS 2013 Programming with Data Structures (3)

CS 2148 Discrete Structures (3)

ENGL 2030 Introduction to Technical Writing (3)

CS 3035 Programming Language Paradigms (3)

CS 3112 Analysis of Algorithms (3)

CS 3186 Introduction to Automata Theory (3)

CS 3220 Web and Internet Programming (3)

CS 3337 Software Engineering (3)

EE 3445 Computer Organization (3)

CS 4440 Introduction to Operating Systems (3)

* + 1. *An exposure to a variety of programming languages and systems.*

CS 1222 Introduction to Relational Databases(3)

CS 2011 Introduction to Programming I (3)

CS 2012 Introduction to Programming II (3)

CS 2013 Programming with Data Structures (3)

CS 3035 Programming Language Paradigms (3)

CS 3220 Web and Internet Programming (3)

CS 3337 Software Engineering (3)

CS 3801 Societal and Ethical issues in Computing (3)

EE 3445 Computer Organization (3)

* + 1. *Proficiency in at least one higher-level language.*

CS 1222 Introduction to Relational Databases(3)

CS 2011 Introduction to Programming I (3)

CS 2012 Introduction to Programming II (3)

CS 2013 Programming with Data Structures (3)

CS 3035 Programming Language Paradigms (3)

CS 3220 Web and Internet Programming (3)

* + 1. *Advanced course work that builds on the fundamental course work to provide depth.*

CS 3035 Programming Language Paradigms (3)

CS 3112 Analysis of Algorithms (3)

CS 3186 Introduction to Automata Theory (3)

CS 3220 Web and Internet Programming (3)

CS 3337 Software Engineering (3)

CS 3801 Societal and Ethical issues in Computing (3)

EE 3445 Computer Organization (3)

CS 4440 Introduction to Operating Systems (3)

CS 4961 Software Design Laboratory I (3)

CS 4962 Software Design Laboratory II (3)

CS 4963 Computer Science Recapitulation (3)

CS3000/4000 upper division computer science electives - (18)

*b. One year of science and mathematics:*

* + 1. *Mathematics: At least one half year that must include discrete mathematics. The additional mathematics might consist of courses in areas such as calculus, linear algebra, numerical methods, probability, statistics, number theory, geometry, or symbolic logic.*

MATH 2110 Calculus I (4)

MATH 2120 Calculus II (4)

MATH 2550 Introduction to Linear Algebra (3)

CS 2148 Discrete Structures (3)

Mathematics Electives (3)

* + - Select 3 units of lower division or upper division course(s) in the Mathematics area with prior approval of the adviser.
    1. *Science: A science component that develops an understanding of the scientific method and provides students with an opportunity to experience this mode of inquiry in courses for science or engineering majors that provide some exposure to laboratory work.*

PHYS 2100 General Physics I (5)

PHYS 2200 General Physics II (5)

General Education Science requirement (3)

* + - Select one course from Biological science OR Interdisciplinary Physical-Biological science

### 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**

Ten of the eleven 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 either tenured or on tenure-track and has many years of teaching experience.

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

Faculty expertise cover all of the major areas of computer science with a number of faculty who publish original research and teach courses.

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 two semester teaching load with twelve weighted teaching units (WTU) per quarter. This typically involves teaching three separate courses every quarter. Faculty who receive 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, Principal Advisor, Senior Design Project Advisor, and Graduate Thesis Advisor) have a reduced teaching load.

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

In spite of the heavy teaching load, faculty show excellent scholarship. 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 eleven full-time faculty and ten regular part-time faculty. We have hired three tenure-track faculty in the last four years. We had an unsuccessful search in 2017 and are in the process of recruiting one more in 2018. The total number of faculty are adequate to teach the courses we offer and to perform other tasks related to program assessment and continuous improvement. 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 yearlong 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. In addition, the College also provide a 1:2 matching of released time for any awarded federal grants, to allow the PI with more time to focus on the research work. 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 creative leaves, summer research, professional travel, and general research. A mini creative leave generally provides 3 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 creative leaves and sabbatical leaves.

All new faculty are supported with a competitive amount of startup funding, upwards of $50,000, which provides for professional activities over a three year period. The university provides faculty travel support of $1,500 per year. This generally provides each faculty .member funds to attend one conference per year. In addition, the Department and the College generate additional funds through Open University offerings that is also used to support faculty professional development.

Another notable factor is that many of our faculty are involved in student capstone projects both at the undergraduate (senior design team projects) and graduate (individual thesis/projects) levels. Most of the projects are externally sponsored. In addition to providing a minimal release time, these projects are a significant scholarly work.

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 University evaluates every program on a 5 to 6 year cycle. Because of the semester conversion, this formal process has been delayed to 2018. The report from the University’s most recent (2010) review of our program included the following.

*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.*

Since CSULA is primarily a teaching institution and since full-time faculty are responsible for twelve weighted teaching units each semester. Despite this heavy teaching load, 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 | 31 | - | M | M (now)  H (prev) | L (now)  H (prev) |
| Vladimir Akis | Ph.D.- 1982  Mathematics | P | T | FT | 0 | 40 | 36 | - | L | M | L |
| Huiping Guo | Ph.D.- 2003  Computer Science | P | T | FT | 4 | 13 | 13 | - | L | M | M |
| Jiang Guo | Ph.D.- 1996  Computer Science | P | T | FT | 3 | 17 | 16 | - | M | M | L |
| Eun-Young “Elaine” Kang | Ph.D. – 2003  Computer Science | P | T | FT | 1 | 15 | 14 | - | M | M | L |
| Raj S Pamula | Ph.D. – 1987  Computer Science | P | T | FT | 1 | 31 | 31 | - | L | L | L |
| Behzad Parviz | Ph.D. – 1986  Computer Science | P | T | FT | 10 | 32 | 32 | - | L | L | L (now)  H (prev) |
| Mohammad Pourhomayoun | Ph.D.- 2013  Computer Science | AST | TT | FT | 2 | 2 | 2 | - | H | H | M |
| Chengyu Sun | Ph.D.- 2004  Computer Science | P | T | FT | 1 | 14 | 14 | - | L | M | L |
| Zilong Ye | Ph.D.- 2015  Computer Science | AST | TT | FT | 0 | 3 | 3 | - | H | H | L |
| Yuquing Zhu | Ph.D.- 2014  Computer Science | AST | TT | FT | 0 | 4 | 4 | - | H | H | L |
| Albert Cervantes | MS – 2006  Electrical Engg. | A | NTT | PT | 7 | 5 | 5 | - | L | M | H |
| Richard Cross | MS – xxxx  Computer Science | A | NTT | PT |  |  |  |  |  |  |  |
| Edmund Gean | MS – 1991  Computer Science | A | NTT | PT | 20 | 9 | 9 | MCPE  CCNA | M | H | H |
| John Hurley | MS-2012  Computer Science | A | NTT | PT | 20 | 5 | 1 | - | L | L | L |
| Keenan Knaur | MS – xxxx  Computer Science | A | NTT | PT |  |  |  |  |  |  |  |
| Jung Soo Lim | PH.D – xxxx  Computer Science | A | NTT | PT |  |  |  |  |  |  |  |
| Jose M. Macias | Ph.D – 1998  Mathematics | A | NTT | PT | 31 | 30 | 30 | - | M | H | H |
| Mark Sargent | MS – xxxxCS  Ph.D – xxxx Phil. | A | NTT | PT |  |  |  |  |  |  |  |
| Jithika Thomas | PH.D – xxxx  Computer Science | A | NTT | PT |  |  |  |  |  |  |  |
| John Tran | MS – 1998  Computer Science | A | NTT | PT | 14 | 14 | 5 | - | H | H | H |
| Eric Liao |  |  |  |  |  |  |  |  |  |  |  |
| Randall Moss |  |  |  |  |  |  |  |  |  |  |  |

**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.)**  **(Summer 2017, Fall2017, Spring2018)** | **Program Activity Distribution** | | | **% of Time Devoted**  **to the Program** |
| Teaching | Research or Scholarship | Other  (Service/  Sabbatical  leave) |
| Russell J. Abbott | FT |  | 80% | 10% | 10% | 100% |
| Vladimir Akis | FT |  | 80% | 10% | 10% | 50% |
| Huiping Guo | FT |  | 55% | 35% | 10% | 100% |
| Jiang Guo | FT |  | 80% | 10% | 10% | 100% |
| Eun-Young “Elaine” Kang | FT |  | 80% | 10% | 10% | 100% |
| Raj S Pamula | FT |  | 40% | 10% | 50% | 100% |
| Behzad Parviz | FT |  | 80% | 10% | 10% | 100% |
| Mohammad Pourhomayoun | FT |  | 80% | 10% | 10% | 100% |
| Chengyu Sun | FT |  | 80% | 10% | 10% | 100% |
| Zilong Ye | PT |  | 20% | - | - | 20% |
| Yuquing Zhu | PT |  | 20% | - | - | 20% |
| Albert Cervantes | PT |  | 20% | - | - | 20% |
| Richard Cross | PT |  | 50% | - | - | 50% |
| Edmund Gean | PT |  | 20% | - | - | 20% |
| John Hurley | PT |  | 20% | - | - | 20% |
| Keenan Knaur | PT |  | 20% | - | - | 20% |
| Jung Soo Lim |  |  |  |  |  |  |
| Jose M. Macias |  |  |  |  |  |  |
| Mark Sargent |  |  |  |  |  |  |
| Jithika Thomas |  |  |  |  |  |  |
| John Tran |  |  |  |  |  |  |
| Eric Liao |  |  |  |  |  |  |
| Randal Moss |  |  |  |  |  |  |

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

| **Code** | **Name** | **Units** | **Coordinator** |
| --- | --- | --- | --- |
| CS 1010 | [Introduction to Higher Education for Computer Science Majors](https://csns.calstatela.edu/course/view?id=4932957) | 3 | Zilong Ye |
| CS 1090 | [BASIC Programming](https://csns.calstatela.edu/course/view?id=4932959) | 2 | Raj Pamula |
| CS 1200 | [Living in a Technology-Based World](https://csns.calstatela.edu/course/view?id=4932963) | 3 | Raj Pamula |
| CS 1220 | [Introduction to Website Development](https://csns.calstatela.edu/course/view?id=4932966) | 3 | Chengyu Sun |
| CS 1222 | [Introduction to Relational Databases](https://csns.calstatela.edu/course/view?id=4932968) | 3 | Chengyu Sun |
| CS 2010 | [Computer Programming Fundamentals](https://csns.calstatela.edu/course/view?id=4932973) | 3 | Yuqing Zhu |
| CS 2011 | [Introduction to Programming I](https://csns.calstatela.edu/course/view?id=4932976) | 3 | Yuqing Zhu |
| CS 2012 | [Introduction to Programming II](https://csns.calstatela.edu/course/view?id=4932979) | 3 | Yuqing Zhu |
| CS 2013 | [Programming with Data Structures](https://csns.calstatela.edu/course/view?id=4932981) | 3 | Yuqing Zhu |
| CS 2148 | [Discrete Structures](https://csns.calstatela.edu/course/view?id=4932985) | 3 | Behzad Parviz |
| CS 2445 | [Introduction to Computer Systems](https://csns.calstatela.edu/course/view?id=4932988) | 3 | Zilong Ye |
| CS 2540 | [Special Topics in Computer Science](https://csns.calstatela.edu/course/view?id=4932990) | 3 | Raj Pamula |
| CS 2550 | [Introduction to Computer Graphics Tools](https://csns.calstatela.edu/course/view?id=5428364) | 3 | Elaine Kang |
| CS 3034 | [Widely-Used Programming Languages](https://csns.calstatela.edu/course/view?id=4932992) | 3 | Russ Abbott |
| CS 3035 | [Programming Language Paradigms](https://csns.calstatela.edu/course/view?id=4932996) | 3 | Russ Abbott |
| CS 3112 | [Analysis of Algorithms](https://csns.calstatela.edu/course/view?id=4932998) | 3 | Behzad Parviz |
| CS 3186 | [Introduction to Automata Theory](https://csns.calstatela.edu/course/view?id=4933003) | 3 | Vladimir Akis |
| CS 3220 | [Web and Internet Programming](https://csns.calstatela.edu/course/view?id=4933005) | 3 | Chengyu Sun |
| CS 3337 | [Software Engineering](https://csns.calstatela.edu/course/view?id=4933007) | 3 | Jiang Guo |
| CS 3550 | [Game Development for Graphic Communications](https://csns.calstatela.edu/course/view?id=4933010) | 3 | Elaine Kang |
| CS 3555 | [Mobile Development for Graphic Communications](https://csns.calstatela.edu/course/view?id=4933012) | 3 | Elaine Kang |
| CS 3660 | [Complex Social and Economic Systems](https://csns.calstatela.edu/course/view?id=4933014) | 3 | Mohammad Pourhomayoun |
| CS 3801 | [Societal and Ethical issues in Computing](https://csns.calstatela.edu/course/view?id=4933018) | 3 | Russ Abbott |
| CS 3890 | [Cooperative Education](https://csns.calstatela.edu/course/view?id=4933021) | 3 | Raj Pamula |
| CS 4075 | [Concurrent and Distributed Programming](https://csns.calstatela.edu/course/view?id=4933023) | 3 | Yuqing Zhu |
| CS 4112 | [Competitive Programming](https://csns.calstatela.edu/course/view?id=4933025) | 3 | Russ Abbott |
| CS 4188 | [Compilers](https://csns.calstatela.edu/course/view?id=4933027) | 3 | Raj Pamula |
| CS 4220 | [Current Trends in Web Design and Development](https://csns.calstatela.edu/course/view?id=4933029) | 3 | Chengyu Sun |
| CS 4222 | [Principles of Database Systems](https://csns.calstatela.edu/course/view?id=4933031) | 3 | Huiping Guo |
| CS 4440 | [Introduction to Operating Systems](https://csns.calstatela.edu/course/view?id=4933033) | 3 | Jiang Guo |
| CS 4470 | [Computer Networking Protocols](https://csns.calstatela.edu/course/view?id=4933035) | 3 | Zilong Ye |
| CS 4471 | [Computer Networks Configuration and Management](https://csns.calstatela.edu/course/view?id=4933038) | 3 | Zilong Ye |
| CS 4540 | [Special Topics in Computer Science](https://csns.calstatela.edu/course/view?id=4933042) | 3 | Raj Pamula |
| CS 4550 | [Computer Graphics](https://csns.calstatela.edu/course/view?id=4933046) | 3 | Elaine Kang |
| CS 4551 | [Multimedia Software Systems](https://csns.calstatela.edu/course/view?id=4933049) | 3 | Elaine Kang |
| CS 4555 | [Introduction to 3D Computer Game Programming](https://csns.calstatela.edu/course/view?id=4933051) | 3 | Elaine Kang |
| CS 4556 | [Multiplayer Online Game Design and Development](https://csns.calstatela.edu/course/view?id=4933053) | 3 | Elaine Kang |
| CS 4635 | [Modeling and Simulation](https://csns.calstatela.edu/course/view?id=4933055) | 3 | Russ Abbott |
| CS 4660 | [Artificial Intelligence](https://csns.calstatela.edu/course/view?id=4933058) | 3 | Mohammad Pourhomayoun |
| CS 4661 | [Introduction to Data Science](https://csns.calstatela.edu/course/view?id=4933060) | 3 | Mohammad Pourhomayoun |
| CS 4780 | [Cryptography and Information Security](https://csns.calstatela.edu/course/view?id=4933062) | 3 | Huiping Guo |
| CS 4961 | [Software Design Laboratory I](https://csns.calstatela.edu/course/view?id=4933064) | 3 | Raj Pamula |
| CS 4962 | [Software Design Laboratory II](https://csns.calstatela.edu/course/view?id=4933067) | 3 | Raj Pamula |
| CS 4963 | [Computer Science Recapitulation](https://csns.calstatela.edu/course/view?id=4933069) | 3 | Zilong Ye |
| CS 4990 | [Undergraduate Directed Study](https://csns.calstatela.edu/course/view?id=4933072) | 3 | Raj Pamula |

**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. 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.

Clerical staff affiliated with the department have an office. Other technical staff are housed in the College.

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 many instances, Computer Science courses are taught in Technology based 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 CS1xxx/CS2xxx and a few CS3xxx 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. 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.

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. Ye and Dr. Zhu) 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 eleven faculty in 2015-16. We had to abort our search in 2016-17 as all our top candidates withdrew in the last minute. We are in the process of hiring one more in the 2017-18 cycle.

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 over the last 10 years.

### E. Support of Faculty Professional Development

All faculty are required to teach 12 weighted teaching units per semester. 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 $1500 per year for conference travel. 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 semester release time. Competition across the University for this awards is intense.
  + All new faculty are supported with a competitive amount of startup funding, upwards of $50,000, which provides for professional activities over a three year period
  + The Department and the College generate additional funds through Open University offerings that is also used to support faculty professional development.

All faculty members have kept up their scholarly activities. (See Appendix B)

# PROGRAM CRITERIA

The Computer ***Science*** program criteria states the following:

***Student Outcomes*** *The program must enable students to attain, by the time of graduation:*

*(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. [CS]*

*(k) An ability to apply design and development principles in the construction of software systems of varying complexity. [CS]*

***Curriculum*** *Students must have the following amounts of course work or equivalent educational experience:*

*a. Computer science: One and one-third years that must include:*

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

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

*3. Proficiency in at least one higher-level language.*

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

*b. One year of science and mathematics:*

*1. Mathematics: At least one half year that must include discrete mathematics. The additional mathematics might consist of courses in areas such as calculus, linear algebra, numerical methods, probability, statistics, number theory, geometry, or symbolic logic.*

*2. Science: A science component that develops an understanding of the scientific method and provides students with an opportunity to experience this mode of inquiry in courses for science or engineering majors that provide some exposure to laboratory work.*

***Faculty*** *Some full time faculty members must have a Ph.D. in computer science.*

***Student Outcomes***

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.

***Curriculum***

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 A6 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* (36 units)

*An exposure to a variety of programming languages and systems*. (27 units)

*Proficiency in at least one higher-level language and familiarity with at least three more* (18 units)

*Advanced course work that builds on the fundamental course work to provide depth*. (33 core units + 18 elective units)

***Mathematics*** (17 units)

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

***Science*** (13 units)

General Physics, Natural Science

**Faculty**

As described under Criterion 6, all faculty are qualified with 10 of the 11 faculty having a Ph.D in Computer Science. 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)].

| **Code** | **Name** | **Units** |
| --- | --- | --- |
| CS 1010 | [Introduction to Higher Education for Computer Science Majors](https://csns.calstatela.edu/course/view?id=4932957) | 3 |
| CS 1090 | [BASIC Programming](https://csns.calstatela.edu/course/view?id=4932959) | 2 |
| CS 1200 | [Living in a Technology-Based World](https://csns.calstatela.edu/course/view?id=4932963) | 3 |
| CS 1220 | [Introduction to Website Development](https://csns.calstatela.edu/course/view?id=4932966) | 3 |
| CS 1222 | [Introduction to Relational Databases](https://csns.calstatela.edu/course/view?id=4932968) | 3 |
| CS 2010 | [Computer Programming Fundamentals](https://csns.calstatela.edu/course/view?id=4932973) | 3 |
| CS 2011 | [Introduction to Programming I](https://csns.calstatela.edu/course/view?id=4932976) | 3 |
| CS 2012 | [Introduction to Programming II](https://csns.calstatela.edu/course/view?id=4932979) | 3 |
| CS 2013 | [Programming with Data Structures](https://csns.calstatela.edu/course/view?id=4932981) | 3 |
| CS 2148 | [Discrete Structures](https://csns.calstatela.edu/course/view?id=4932985) | 3 |
| CS 2445 | [Introduction to Computer Systems](https://csns.calstatela.edu/course/view?id=4932988) | 3 |
| CS 2540 | [Special Topics in Computer Science](https://csns.calstatela.edu/course/view?id=4932990) | 3 |
| CS 2550 | [Introduction to Computer Graphics Tools](https://csns.calstatela.edu/course/view?id=5428364) | 3 |
| CS 3034 | [Widely-Used Programming Languages](https://csns.calstatela.edu/course/view?id=4932992) | 3 |
| CS 3035 | [Programming Language Paradigms](https://csns.calstatela.edu/course/view?id=4932996) | 3 |
| CS 3112 | [Analysis of Algorithms](https://csns.calstatela.edu/course/view?id=4932998) | 3 |
| CS 3186 | [Introduction to Automata Theory](https://csns.calstatela.edu/course/view?id=4933003) | 3 |
| CS 3220 | [Web and Internet Programming](https://csns.calstatela.edu/course/view?id=4933005) | 3 |
| CS 3337 | [Software Engineering](https://csns.calstatela.edu/course/view?id=4933007) | 3 |
| CS 3550 | [Game Development for Graphic Communications](https://csns.calstatela.edu/course/view?id=4933010) | 3 |
| CS 3555 | [Mobile Development for Graphic Communications](https://csns.calstatela.edu/course/view?id=4933012) | 3 |
| CS 3660 | [Complex Social and Economic Systems](https://csns.calstatela.edu/course/view?id=4933014) | 3 |
| CS 3801 | [Societal and Ethical issues in Computing](https://csns.calstatela.edu/course/view?id=4933018) | 3 |
| CS 3890 | [Cooperative Education](https://csns.calstatela.edu/course/view?id=4933021) | 3 |
| CS 4075 | [Concurrent and Distributed Programming](https://csns.calstatela.edu/course/view?id=4933023) | 3 |
| CS 4112 | [Competitive Programming](https://csns.calstatela.edu/course/view?id=4933025) | 3 |
| CS 4188 | [Compilers](https://csns.calstatela.edu/course/view?id=4933027) | 3 |
| CS 4220 | [Current Trends in Web Design and Development](https://csns.calstatela.edu/course/view?id=4933029) | 3 |
| CS 4222 | [Principles of Database Systems](https://csns.calstatela.edu/course/view?id=4933031) | 3 |
| CS 4440 | [Introduction to Operating Systems](https://csns.calstatela.edu/course/view?id=4933033) | 3 |
| CS 4470 | [Computer Networking Protocols](https://csns.calstatela.edu/course/view?id=4933035) | 3 |
| CS 4471 | [Computer Networks Configuration and Management](https://csns.calstatela.edu/course/view?id=4933038) | 3 |
| CS 4540 | [Special Topics in Computer Science](https://csns.calstatela.edu/course/view?id=4933042) | 3 |
| CS 4550 | [Computer Graphics](https://csns.calstatela.edu/course/view?id=4933046) | 3 |
| CS 4551 | [Multimedia Software Systems](https://csns.calstatela.edu/course/view?id=4933049) | 3 |
| CS 4555 | [Introduction to 3D Computer Game Programming](https://csns.calstatela.edu/course/view?id=4933051) | 3 |
| CS 4556 | [Multiplayer Online Game Design and Development](https://csns.calstatela.edu/course/view?id=4933053) | 3 |
| CS 4635 | [Modeling and Simulation](https://csns.calstatela.edu/course/view?id=4933055) | 3 |
| CS 4660 | [Artificial Intelligence](https://csns.calstatela.edu/course/view?id=4933058) | 3 |
| CS 4661 | [Introduction to Data Science](https://csns.calstatela.edu/course/view?id=4933060) | 3 |
| CS 4780 | [Cryptography and Information Security](https://csns.calstatela.edu/course/view?id=4933062) | 3 |
| CS 4961 | [Software Design Laboratory I](https://csns.calstatela.edu/course/view?id=4933064) | 3 |
| CS 4962 | [Software Design Laboratory II](https://csns.calstatela.edu/course/view?id=4933067) | 3 |
| CS 4963 | [Computer Science Recapitulation](https://csns.calstatela.edu/course/view?id=4933069) | 3 |
| CS 4990 | [Undergraduate Directed Study](https://csns.calstatela.edu/course/view?id=4933072) | 3 |

**Table A.1: List of Courses**

Top of Form

Sun. All rights reserved.

# 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.*

.

Dean, College of Engineering Computer Science, and Technology

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature Date**