California State University, Los Angeles

Department of Computer Science

CS 5035 - Topics in Functional Programming, Spring 2020

# COURSE INFORMATION

## Instructor Information

**Instructor**: Chunhui Guo

**Office Location**: ET A310

**Email**: [cguo4@calstatela.edu](mailto:cguo4@calstatela.edu?subject=CS5035%20YourSubject)

**Office Hours**: MW 1:30-3:00 PM

## General Course Information

**Class Days/Time**: MW 3:05-4:20 PM

**Classroom**: ET A129

**Prerequisites**: CS 3035 (Programming Language Paradigms), CS 3112 (Analysis of Algorithms)

**Add/Drop Deadline**: February 4th (Tuesday)

## Course Description

This course introduces students to functional programming with OCaml language. The subjects covered in this course include evaluating expressions vs. executing statements, functions as first-class objects, higher-order functions, currying, partial evaluation, closures, mutable and immutable variables, comprehensions, referential transparency, and lazy evaluation.

# REQUIRED COURSE MATERIALS

## Slides, Lecture Notes, and Sample Codes

All slides, lecture notes, and sample codes used in lectures will be available on the [CSNS course website](https://csns.calstatela.edu/site/s20/cs5035-1).

## Textbook

The textbook has a [free online version](https://dev.realworldocaml.org/) provided by the authors. The students are not required to buy the hardcopy.

**Title:** Real World OCaml - Functional Programming for the Masses

**Author:** Yaron Minsky, Anil Madhavapeddy, Jason Hickey

**Edition:** 2nd edition

**ISBN:** 9781449323912

**Link:** <https://dev.realworldocaml.org/>

## Software Requirements

The students are free to choose their favorite operating system and software if they could successfully configure an OCaml development environment. The OCaml community also provides an [online development environment](https://try.ocamlpro.com/), which is a good tool to get familiar with OCaml basics. I strongly suggest the students configure a local environment, because using the online development environment is very difficult to finish assignments and projects that require multiple source code files.

# COURSE COMMUNICATION

The Instructor will make every effort to communicate frequently with students through announcements/postings on the [CSNS course website](https://csns.calstatela.edu/site/s20/cs5035-1) and emails. We will use the [CSNS Class Forum](https://csns.calstatela.edu/department/cs/forum/view?id=4933091) as our discussion platform. I encourage you to post any nonpersonal questions in the forum. Questions of a more personal nature can be sent to the instructor via email [cguo4@calstatela.edu](mailto:cguo4@calstatela.edu?subject=CS5035%20YourSubject). In order for me to attend to your email, the email subject should start with “CS5035”.

# ASSIGNMENTS AND GRADING POLICY

There will be about 20 in-class activities, 5 assignments, and 2 projects throughout the semester. The course does not have exams. Actually, the Midterm and Final exams will be replaced with the projects, respectively.

## In-Class Activities:

* Goal: summarize the content of corresponding lectures
* Time: about 5-10 minutes
* Type: 1 or 2 quick questions, individual task
* Submission: digital-copy, CSNS
  + submit all activities via the same [“Activities” assignment link](https://csns.calstatela.edu/submission/list?assignmentId=7534369) in CSNS
  + you may submit txt, pdf, or docx files
  + name your submission with “Activity #”
* Grades: 1 point each activity, not graded, go over in class
* Due: 11:59pm on corresponding class day
* Late Policy: NO late submissions are accepted

## Assignments:

* Goal: get up to speed with OCaml
* Time: two weeks
* Type: programming, individual task
* Submission: CSNS, digital-copy (see each assignment’s submission instruction)
* Grades: 100 points each assignment
* Due: 11:59pm on the date specified
* Late Policy: The assignments have a 2-hour grace period. Late assignments are assessed a 20% penalty per day or partial day: 0-2 hours late (grace period - no penalty), 2-24 hours late (20%), 24-48 hours late (40%), and so forth. Note that the grace period applies to the first day only! All exceptions and extensions must be approved by the instructor.

## Projects:

* Goal: build real-world projects in OCaml
* Time: four weeks
* Type: programming + short report + demo, group with maximal 2 members
* Submission: CSNS, digital-copy (see each assignment’s submission instruction)
* Grades: 100 points each project (80 points for code, 20 points for report)
* Due: The first project is due at 11:59pm on the date specified. The second project is due at 4:30pm on May 11 (Monday), which meets the university’s final schedule.
* Late Policy: NO late submissions are accepted

## Rebuttal Policy

Once the grades of assignments and projects are released in CSNS, the students have a one-week period to rebuttal their grades. After the rebuttal period, the grades become final. In order to meet the university’s grade due date, the second project’s rebuttal period will be announced separately through email.

## Grading Weighting

|  |  |
| --- | --- |
| **Category** | **Percentage** |
| In-Class Activities | 10% |
| Assignments | 50% |
| Projects | 40% (20% each) |

## Grading Scale

|  |  |
| --- | --- |
| **Letter Grade** | **Percentage** |
| A | >= 90% |
| A- | >= 85% && < 90% |
| B+ | >= 80% && < 85% |
| B | >= 75% && < 80% |
| B- | >= 70% && < 75% |
| C+ | >= 65% && < 70% |
| C | >= 60% && < 65% |
| F | < 60% |

Grading Options for Spring 2020: <http://www.calstatela.edu/ecst/success/grading-spring-2020>

# COURSE SCHEDULE (subject to change)

|  |  |  |
| --- | --- | --- |
| **Date** | **Topic** | **Readings & Dues** |
| The Basics | | |
| 1/22 | Course Intro & OCaml Setup |  |
| 1/27 | OCaml Intro | * Lecture Notes: Functional Basics, Type Checking * Textbook (optional): Chapter 1 (OCaml as a Calculator & Functions and Type Inference) |
| 1/29 | Simple Immutable Data: Variables, Tuples, Options | * Lecture Notes: Type-directed Programming * Textbook (optional): Chapter 2 |
| 2/03 | Thinking Inductively | * Lecture Notes: Thinking Inductively * Textbook (optional): Chapter 3 |
| 2/05 | Poly-HO: Polymorphism and Higher-Order Programming | * Lecture Notes: Polymorphism and Higher-order Programming |
| 2/10 | More Data | * Textbook (optional): Chapter 6 * Due: Assignment 1 |
| Reasoning About Programs | | |
| 2/12 | Building Interpreters | * Lecture Notes: Operational Semantics |
| 2/17 | More Interpreters | * Lecture Notes: Equational Reasoning |
| 2/19 | Proving Programs Correct |  |
| 2/24 | More reasoning: Data types + Space | * Lecture Notes: Equational reasoning about data types * Online Resource (optional): Odd-Order Theorem, Four-Color Theorem * Due: Assignment 2 |
| 2/26 | Continuation-Passing Style |  |
| 3/02 | More Continuation-Passing |  |
| 3/04 | Computability |  |
| Modularity | | |
| 3/09 | Modules I | * Textbook (optional): Chapter 4, Chapter 9 * Due: Assignment 3 |
| 3/11 | Project 1 Intro |  |
| 3/16 | Modules II |  |
| 3/18 | Effective ML |  |
| 3/23 | Reasoning about Modules | * Lecture Notes (optional): Abstraction and Representation Invariants |
| 3/25 | Module Equivalence & Refs | * Textbook (optional): Chapter 8 * Due: Assignment 4 |
| 3/30 | Spring Break |  |
| 4/1 | Spring Break |  |
| 4/6 | Infinite Streams and Lazy Evaluation |  |
| 4/8 | Project 2 Intro & Project 1 Demo | * Due: Project 1 |
| 4/13 | Type Checking and Inference | * Online Resource (optional): Efficient Type Inference |
| 4/15 | Type Checking II |  |
| Parallelism and Concurrency | | |
| 4/20 | Mutable Data Structures and Imperative Interfaces |  |
| 4/22 | Parallelism Intro |  |
| 4/27 | Threads & Futures | * Lecture Notes: Using threads and futures * Due: Assignment 5 |
| 4/29 | Scheduling & Parallel Sequences | * Lecture Notes: Parallel Sequences * Paper (optional): MapReduce * Online Resource (optional): NESL |
| 5/4 | Parallel Collections |  |
| 5/6 | Parallel Complexity Models | * Lecture Notes: Complexity Analysis for Parallel Programs |
| 5/11 | Project 2 Demo (Final) | * Due: Project 2 |

# HELPFUL STUDENT RESOURCES

## Technical Resources

Information on Cal State LA technical support resources for students: [Technical Support Resources](http://www.calstatela.edu/moodlementor/technical-support-resources)

## Student Support Resources

Information on Cal State LA student support resources for students: [Student Support Resources](http://www.calstatela.edu/cetl/student-support-resources)

## Academic Support Resources

Information on Cal State LA academic support resources for students: [Academic Support Resources](http://www.calstatela.edu/moodlementor/academic-support-resources)

## Glazer Family Dreamers Resource Center

The [Erika J. Glazer Family Dreamers Resource Center](http://www.calstatela.edu/gfdrc) promotes the success of undocumented students and students from mixed-status families at Cal State LA through a variety of resources, services, and community engagement opportunities. Such programs and services are weekly immigration legal clinics, California Dream Act Application for Financial Aid Assistance, and professional and academic development workshops.

# UNIVERSITY POLICIES

## Student Conduct

Information on student rights and responsibilities, standards of conduct, etc., can be found by visiting the Cal State LA [University Catalog Appendices](http://ecatalog.calstatela.edu/content.php?catoid=26&navoid=2721).

## Dropping and Adding

Students are responsible for understanding the policies and procedures about add/drops, academic renewal, etc. Students should be aware of the current deadlines and penalties for adding and dropping classes by visiting the [GET home page](https://cmsweb.calstatela.edu/psp/CLAPRD/?cmd=login&languageCd=ENG&). (Registrar news and information)

## Americans with Disabilities Act (ADA)

Reasonable accommodation will be provided to any student who is registered with the Office of Students with Disabilities and requests needed accommodation. For more information visit the [Office for Students with Disabilities](http://www.calstatela.edu/osd) home page.

## Academic Honesty

Many incidents of plagiarism result from students’ lack of understanding about what constitutes plagiarism. However, you are expected to familiarize yourself with the [Cal State LA Academic Honesty Policy](http://ecatalog.calstatela.edu/content.php?catoid=26&navoid=2646) including [Appendix D – Academic Honesty](http://ecatalog.calstatela.edu/content.php?catoid=26&navoid=2646) and [Appendix E - Student Conduct / Student Conduct Procedures](http://ecatalog.calstatela.edu/content.php?catoid=26&navoid=2647). All work you submit must be your own scholarly and creative efforts. Cal State LA plagiarism as follows: “At Cal State LA, plagiarism is defined as the act of using ideas, words, or work of another person or persons as if they were one’s own, without giving proper credit to the original sources.”