CS 1101 Computer Science I
Sections 04 and 05 Fall 2015

Instructor Muller

Syllabus

Welcome to CS1101 sections 04 and 05. This course is an introduction to the art and science of computer programming and to some of the fundamental concepts of computer science. Students will write programs in the OCaml dialect of the ML programming language. ML is a modern programming language featuring polymorphic static typing, automatic memory management and a robust module system. ML is *value-oriented* — computation is driven by an imperative to find the *value* of an expression. ML is often also called a *functional* programming language because of its emphasis on functions.

Good program design methodology will be stressed throughout the course. There will also be a study of some of the basic notions of computer science, including computer systems organization, files, and some algorithms of fundamental importance.

Course Goals

The main goal of this course is to help the student develop an understanding of *computation* and to help them master the art of designing algorithms, and developing the programs that implement them. Important parts of the latter include documenting and testing the program.

Students will learn how to decompose problems into specific subproblems, write an algorithm to solve a specific problem, and then translate that algorithm into an OCaml program.

Basic Information

CS 1101 has two 75-minute lectures each week and one one-hour lab. Lectures for section 04 are held on Tuesdays and Thursdays at 9AM in Fulton Hall 415. Lectures for section 05 are held on Tuesdays and Thursdays at 1:30PM in Fulton Hall 250. Attendance at the lectures is critical, as all new material will be presented there. **Note:** Laptops are not allowed in lecture. If you have special circumstances that require you to bring one, please see me.

Course Web Site

Please bookmark the course homepage:

`http://www.cs.bc.edu/~muller/teaching/cs1101/f15/`

We will use this site very heavily throughout the semester and most of the course materials will be distributed through this site. Some lab materials will be distributed through the linked Piazza site. Problem sets are to be submitted through the linked Canvas website. Note that we will not have a course homepage on Canvas.

Lab Times

All lab/section meetings are in Higgins 260 or 280. The labs are run by the course Teaching Assistants. They provide a great opportunity for you to work with a partner and practice with new material on some fun problems in a supervised setting. Regular attendance at labs is strongly encouraged. **Please bring your laptop if you have one.** If you do not have a laptop, please see me.

Lab times are:

1. CSCI100601 (Nick Denari) Tuesdays 4PM - 5PM, Higgins 260
2. CSCI100602 (Elias Hatem) Wednesdays 5PM - 6PM, Higgins 280
3. CSCI100603 (Jesse Mu) Tuesdays 5PM - 6PM, Higgins 260
4. CSCI100604 (Meagan Gonzalez) Wednesdays 4PM - 5PM, Higgins 280

Staff

Instructor: Robert Muller, robert.muller2@gmail.com, office: St. Mary's Hall Rm S277, hours: Mondays 10AM - 12PM, Thursdays 8:15AM - 10:15AM, 617-552-3964.

Teaching Assistant: Nick Denari, lastname AT bc DOT edu, office: Fulton 160, hours: Fridays 10AM - noon.

Teaching Assistant: Meagan Gonzalez, lastname yz AT bc DOT edu, office: Fulton 160, hours: Sundays noon - 2PM.

Teaching Assistant: Elias Hatem, lastname e AT bc DOT edu, office: Fulton 160, hours: Mondays 10AM - noon.

Head Teaching Assistant: Jesse Mu, lastname j @bc.edu, office: Fulton 160, hours: Sundays 3PM - 5PM.

Problem Sets

Each week you will be assigned a problem set. Unless otherwise specified, all problem sets are due on 5PM on the specified due date. The single best indicator of success for computer science is *starting problem sets early.*

Problem sets should be submitted for grading by uploading an appropriately named zip file through the course Canvas web site. (As linked from the course home page.) Problem sets cannot be submitted as email attachments. Attempts to submit problem sets as email attachments will not receive an email reply indicating that the attempted submission failed.

Topics

Roughly construed and subject to variation.

1. Overview, Administration, OCaml Setup and Introduction, Types, expressions, reduction, values, functions and libraries. Function definitions and calls. Libraries.

2. Function definitions and calls, the replacement and reduction model of evaluation, type notation, basic graphics, Pattern matching, branching and repetition. Using variables to name values. The `bool` type, comparing, branching.

3. More on repetition, primality, tessellation. Nested definitions, block-structure and scope.

4. More repetition, a bisection algorithm for square roots, Lists, tuples and records.

5. Working with lists, concatenation, zipping, linear search, merging, making change. **First Exam**.

6. Functions are values, map and reduce. Binary search, an introduction to algorithmic complexity/work. Binary search trees.

7. Binary search, complexity/work, binary search trees.

8. Sorting: insertion sort, mergesort and quicksort.

9. *Storage*, persistent and volatile, system and user, static and dynamic, the stack and the heap. Pointers and storage diagrams. The *storage model* of evaluation. Binary and Hex.

10. Machine architecture, assembly code. Mutation. Mutation-based repetition idioms `while`-loops and `for`-loops.

11. **Second Exam**.

12. Mazes, 2D-list representations and algorthims, Backtracking, 8-queens.

13. Working with strings, Dictionaries, Markov models.

14. Tessellation Contest, Defining new types, Java.

15. More on Java, review and wrapup.

Exams

There will be two midterm exams and a final exam. The exams are in class and are closed notes and closed book. You will have 75 minutes to complete each midterm, and 2.5 hours to complete the final. If you require extra time for documented reasons, please let us know.

Midterms

First Midterm Thursday October 1,

Second Midterm Thursday October 29,

Final Exam TBA.

Reading

There is one textbook for the course *OCaml from the Very Beginning* by John Whittington. There are two other books listed on the course homepage. We will use extensive code and lecture notes which will be posted to the course web site.

Grading

Your grade for this class will be a combination of your homework, exam, and participation work. Participation is largely based on effort (not correctness). Lab work and Piazza involvement will be incorporated into the participation score. Final grades are computed, roughly as follows:

- Ten problem sets, these account for 45% of your grade,
- Two midterm exams, each accounts for 12.5% of your grade and a final exam which accounts for 20% of your grade,
- Class, lab and piazza forum participation, together, these account for the remaining 10% of your grade.

Important! If you fail one of these components, you will fail CS 1101, even if your weighted-average scores are mathematically above the passing threshold.

Late Homework Policy

Homework is due on the day indicated at 5PM. *This is a strict deadline.* Homework submitted at 5:01PM is one day late as is homework submitted 23:59 late. Late homework is penalized 25% per 24-hour period.

In the case of medical exigencies, students may petition the the Instructor for an extension. Medical problems or family emergencies are the only conditions under which extensions will be granted.

Honor Code

All solutions and code should be produced by you alone, or by you and a partner, where appropriate. For pair-programmed assignments, each partner needs to submit the assignment and each needs to acknowledge the other partner when submitting.

You may discuss algorithms at a high level with any student in the class. You may also help any student find a small bug in their code. However, you may not copy solutions from anyone, nor should you collaborate beyond high-level discussions with anyone who is not your partner. For pair programming problems, you must follow the guidelines given above.

If you have any questions about what behavior is acceptable, it is your responsibility to come see one of the instructors before you engage in this behavior. We are more than happy to answer any questions you may have.