University of Washington
Computer Science & Engineering 341: Programming Languages
Course Syllabus, Autumn 2010

Instructor

Marty Stepp (stepp@cs). Office: CSE 636. Phone**:** (206) 685-2181. See course web site for office hours.

Course Overview

This course is about basic concepts in programming languages, including abstraction mechanisms, types, and scoping. We will perform a detailed study of several different programming paradigms, such as functional, object-oriented, and logic programming. The prerequisite for this course is CSE 143, and we assume that you come to the course with significant experience in either Java or C++. We will build on that experience by examining three languages that provide an interesting contrast to what you already know. The goal is to give you a much broader view of programming and to teach you the basic terminology that we use to distinguish different programming languages. In this course, you will:

- Internalize an accurate understanding of what functional and object-oriented programs mean
- Develop the skills necessary to learn new programming languages quickly
- Master specific language concepts such that you can recognize them in strange guises
- Learn to evaluate the power and elegance of programming languages and their constructs
- Attain reasonable proficiency in ML, Scheme, and JavaScript
- As a by-product, become more proficient in languages you already know

Lecture Time

MWF 12:30 PM - 1:20 PM, More 220

Discussion Sections

You will be expected to participate in a weekly discussion section, held on various times on Thursdays (see course web site for details). The TA who runs your section will grade your homework assignments. In section we will answer questions, go over common errors in homework solutions, and discuss sample problems in more detail than we can in lecture.

Though attendance in section and lecture are strongly encouraged, no part of your grade comes from attendance, in-class participation, or in-class assigned work. In other words, attendance at all lectures and sections is optional, and you will never receive a direct deduction or penalty to your grade from choosing not to attend a particular lecture or section (except for exam days!).

Course Web Site and Email

- http://www.cs.washington.edu/341/

You are expected to check the course web site and your @cs.washington email daily for any important announcements.

Textbook

- Ullman, J. *Elements of ML Programming (2nd Edition)*. ISBN 0137903871. Required.
- ~~Thomas, Fowler, Hunt. *Programming Ruby 1.9: Pragmatic Programmers' Guide*. ISBN 1934356085. Optional.~~

You'll want the ML book; it's good. The Ruby one you do not need and should not buy.

Grading

50%	weekly homework assignments	
20%	midterm	(**Wednesday, November 3, 2010**, *in class*)
30%	final exam	(**Thursday, December 16, 2010, 8:30 - 10:30 AM, MOR 220**)

This maps to the 4.0 scale roughly as follows. You will get at least the grade below for the percentage shown.

90%: at least 3.5, 80%: at least 2.5, 70%: at least 1.5, 60%: at least 0.7.

Historically the median (middle) grade for this course is around 3.3 - 3.4.
We will post scores periodically so that you can see where you stand with respect to your classmates.

Computer Access and Software

The CSE building's basement lab machines have been set up with the necessary software for this course, including compilers and editors for each of the programming languages we will use. The course web site contains links to download this software free of charge if you want to work at home. If you work from home, it is your responsibility to ensure that your program will run on the school's machines, since that is the environment in which your code will be tested and graded. This also means that your code must constrain itself to language features that exist in the labs.

Exams

Our exams are open-book and open-notes. You may bring any written materials, such as textbooks, printed handouts, homework assignments, or programs. No electronic devices may be used, including calculators.

Make-up exams will not be given except in case of a serious emergency. If you must miss an exam, even if you are sick or injured, you must contact the instructor *before* the exam (or arrange for someone to do so). You must show evidence that you are physically unable to take the exam, such as a clear and specific doctor's note mentioning the date, exam, and reason. No make-ups will be granted for personal reasons such as travel, personal hardship, leisure, or to ease exam week schedules. No student will be permitted to take an exam early for any reason.

Homework

Homework consists of weekly programming assignments done individually and submitted electronically from the course web site. Programs will be graded on "external correctness" (behavior) and "internal correctness" (style and design). Disputes about homework grading must be made within **2 weeks** of receiving the grade.

Lateness

Each student receives **3 "late days"** for use on homework assignments. A late day allows you to submit a program up to 24 hours late without penalty. For example, you could use 2 late days and submit a program due Tuesday 9pm on Thursday by 9pm with no penalty. Once a student has used up all the late days, each successive day that an assignment is late will result in a loss of **20% credit** on that assignment. Regardless of how many late days you have, **you may not submit a program more than 2 days after it is due** or after the last day of class. Students will not be given extensions unless they have extenuating circumstances as decided by the instructor.

Academic Integrity and Collaboration

Unless otherwise specified, programming assignments must be completed individually; all code you submit must be your own work. You may discuss general ideas of how to approach an assignment, but never specific details about the code to write. Any help you receive from or provide to classmates should be limited and should never involve details of how to code a solution. You must abide by the following rules:

- You may not work as a partner with another student on an assignment.
- You may not show another student your solution to an assignment, nor look at his/her solution, for any reason.
- You may not have another person "walk you through" an assignment, describe in detail how to solve it, or sit with you as you write it. You also may not provide such help to another student. This includes current or former students, tutors, friends, TAs, paid consultants, people on the Internet, or anyone else.
- You may not post your homework solution code online to ask others for help. This includes public message boards, forums, file sharing sites and services, or any other online system.

Under our policy, a student who gives inappropriate help is equally guilty with one who receives it. Instead of providing such help to someone who does not understand an assignment, point them to other class resources such as lecture examples, the textbook, the IPL, or emailing a TA or instructor. You must not share your solution and ideas with others. You must also ensure that your work is not copied by others, such as making sure to log out of shared computers, not leaving printouts of your code in public places, and not emailing your code to other students or posting it on the web.

If you are retaking the course, you may resubmit a previous solution unless that program was involved in an academic misconduct case. If misconduct was found, you must write a new version of that program.

We enforce this policy vigorously by running similarity detection software a few times per quarter over all submitted student programs, including programs from past quarters. Students who violate the policy are offered reduced scores and sometimes sent to a University committee. This can lead to marks on permanent academic records. Please contact the instructor if you are unsure whether a particular behavior falls within our policy.