

CSC/CPE 357: Systems Programming

Instructional Information

Professor: Aaron Keen
E-mail: akeen@calpoly.edu
Office: 14-230
Office hours: M: 3–4, T: 2–4, W: 3–4, F: 3–4

Professor: George Taylor
E-mail: getaylor@calpoly.edu
Office: 14-240
Office hours: M: 11–12, T: 4:30–5:30, W: 11–12, R: 4:30–5:30, F: 11-12

Course Webpage: <http://www.csc.calpoly.edu/~akeen/courses/csc357>

Lecture Time and Location

- Section 03 (Taylor)
Lecture: TR 9:40am – 11:00am, 14-232B Lab: TR 12:10pm – 1:30pm, 14-303
- Section 09 (Taylor)
Lecture: TR 1:40pm – 3:00pm, 21-133 Lab: TR 3:10pm – 4:30pm, 14-303
- Section 11 (Keen)
Lecture: MWF 12:10pm – 1:00pm, 186-C200 Lab: MWF 2:10pm – 3:00pm, 14-301

Course Objectives

- Transition to upper-division coursework where programming is a tool used in the exploration of course-specific content.
- Transition from following stepwise, prescribed decompositions to independent problem solving.
- Gain experience with low-level programming in the UNIX environment.
- Learn to read and write programs of moderate complexity in the C programming language.
- Learn about the architecture of the UNIX operating system from a system-programmer's perspective and be able to write programs that use operating system services (system calls) directly.
- Learn about basic resource management.

Prerequisites:

The prerequisites for this course are CSC/CPE 103 (C- or better), and CSC 225, CPE 229, or CPE 233.

Texts

The course textbook is *The C Programming Language* by Kernighan and Ritchie and *Advanced Programming in the UNIX Environment* by Stevens and Rago.

Webpage

Clarifications, changes, etc. regarding the class and assignments will be posted to the course webpage (<http://www.csc.calpoly.edu/~akeen/courses/csc357>). Read it regularly. You are responsible for any announcements posted on the course website.

Activities

Class Participation

The lectures are for your benefit. You should ask questions when you have them. If you are confused, then others may be as well, so ask questions and listen carefully to those asked by others.

Reading

A reading schedule will be provided. This schedule outlines the order in which topics will be covered in lecture and the associated chapters and sections in the textbooks that you should read. The lectures may not cover all of the material in the assigned reading, but such material may appear in homework assignments or on exams.

Assignments

There will be 6 programming assignments. Since this is a course on systems programming, there will be some additional resource constraints on some of these assignments. A program that simply “works” will not necessarily be sufficient for a perfect score.

Programming assignments are to be completed by each student without collaboration with other students.

Note: A program that does not compile will not be graded.

Labs

There are 10 lab exercises planned. One exercise will be assigned each week and you will need to demonstrate (during lab) the required portion by the end of the week (exceptions will be noted and/or announced as needed). Some labs also contain ungraded, optional portions.

The lab exercises are, in many cases, meant to introduce, in a smaller context, material that will be used in a larger assignment. You are strongly encouraged to do both the required and optional portions of the lab exercises.

You can work together on lab exercises. You are encouraged to work together with your peers and the instructor on the lab exercises. Be certain, however, that you understand the material for each lab exercise.

Note: Allowing collaborative work on lab exercises does not mean that you are allowed to work together on programming assignments just because you are physically in the lab. Programming assignments are still separate, individual efforts.

Exams

There will be two “midterm” exams and a final exam. The exams will emphasize insight and problem solving with the intent to measure understanding.

The exams will be closed book and closed note.

Prototypes for functions used in the course will be provided with the exam. While you are not required to remember the prototype itself, you will certainly need to remember 1) what the functions are used for, 2) what the functions do, and 3) what the parameters are (a prototype lists only types and names, not purpose).

Grading

The percentage breakdown for the course grade is as follows.

Activity	% per	% total
Programming Assignments		45
Assignment 1	5	
Assignment 2 – 6	8	
Lab Exercises	1	10
Midterm Exams	10	20
Final Exam	25	25
Total		100

Performance on the exams, especially the final, will weigh more heavily in assigning course grades in borderline cases.

Simplicity, presentation, and neatness of your solutions are considered in the grading of assignments and exams.

What you turn in is exactly what will be graded. Be sure to turn in what you intend us to grade, e.g., all required parts and the correct version.

Due Dates and Lateness

Programming assignments are to be turned in electronically. Assignments must be turned in ON TIME to receive credit. Except in the most extreme situations, **late assignments will not be accepted**. If you cannot complete an assignment by the due date, submit whatever you have done in order to receive partial credit. **Receiving partial credit, however, should not be your goal.**

At a minimum, programming assignments **must compile** to be considered for grading. Those that do not meet this minimum criterion will be returned with a score of zero.

Collaboration and Cheating

Policy on Collaboration

Each student is to do his or her own work on the assignments and exams. It is fine to talk with others about general approaches used to solve the assignments, *but* each student is to develop his/her own solution; collaborative efforts are **not** allowed. Students are not to view any other student's code or exchange code in any form (hardcopy or electronically). **Sharing pseudo-code is not allowed.**

Using solutions from any other source is forbidden; in particular, using solutions (either instructors' or other students') from previous offerings of this course is not allowed. Using solutions found on the Internet is not allowed.

Collaboration that goes beyond a high-level discussion of general approaches will be considered cheating. If you are unsure about what constitutes proper or improper collaboration, consult the instructor for guidance.

To summarize: all programming assignments and exams are to be *individual* and *original* efforts.

The Last Page

This page is so I can gather a little information about you at the beginning of the class. Please fill it out, tear it off and leave it with me on the way out.

Who are you?

Name: _____
Major: _____
Enrollment: Enrolled
 Enrolled, thinking about dropping
 Trying to enroll
 Thinking about enrolling

Class Expectations?

Please take a minute to write out what your goals and expectations are for CSC/CPE 357. What do you want to learn? What do you expect to learn? Are these the same thing?

Programming Comfort Level?

How comfortable are you with programming? Do you feel like programming is a personal strength? If not, what can I do to help you?