

CMPS 12A/L Introduction to Programming Winter 2008

Description: An introductory programming course for computer science and engineering majors where students learn programming and documentation skills, as well as algorithmic problem solving and programming methodologies. Introduces students to computers, compilers, and editors, and they are expected to write medium-sized programs. Topics include, but are not limited to, procedures and functions, conditionals and loop control structures, static and dynamic memory manipulations, and text processing. Prior experience with Unix helpful, and some prior programming experience strongly recommended (e.g., course 10). This course is required for computer engineering, computer science, electrical engineering, and information systems management majors. **Prerequisites:** eligibility to enroll in Mathematics 19A (Mathematics 2B or 3 or 40 or higher on mathematics placement exam) or Mathematics 19A or 11A or Economics 11A or Applied Math and Statistics 11A. Concurrent enrollment in course 12L required.

Time and Place: MWF 11:00 – 12:10 JBE 152

Class Webpage: <http://www.soe.ucsc.edu/classes/cmeps012a/Winter08/>

Class Webforum: <http://forums.soe.ucsc.edu/viewforum.php?f=39>

Instructor: Patrick Tantalo (<http://www.cse.ucsc.edu/~ptantalo/>)

Office: E2 257

Office Hours: MWF 1:30-3:30, or by appointment

Email: ptantalo@soe.ucsc.edu

Phone: 831-459-3898

Teaching Assistants:

Neeraj Kumar <nkumar@soe.ucsc.edu>

Radhakrishna Vuppala <vrk@soe.ucsc.edu>

Lab Sections: Provide a time and place for students to complete both the programming assignments for CMPS 12A and the lab assignments for CMPS 12L, and to prepare for exams. Attendance is optional. A current schedule of lab times will be posted on the class webpage.

MSI Tutor:

Chung Lai Lam <cllam@ucsc.edu>

Required Text: *Java by Dissection* (2nd edition) Ira Pohl, Charlie McDowell. Lulu 2006. You can buy this online at <http://www.lulu.com/JavaByDissection>, or pick it up at the bookstore.

Recommended Texts:

Learning Java (3rd edition) Patrick Niemeyer, Jonathan Knudsen. O'Reilly 2005.

C for Java Programmers Tomasz Muldner. Addison Wesley 2000.

A Practical Guide to the UNIX System (3rd edition) Mark G. Sobell. Addison Wesley 1995.

Coursework and Evaluation for CMPS 12A:

We will have five or six **Programming Assignments**, due at roughly two week intervals. The **first Midterm Exam** will be held **Wednesday, February 6**, and the **second Midterm Exam** will be **Wednesday, February 27**. The **Final Exam** will be held on **Thursday, March 20, 8:00 – 11:00 am**. Please make arrangements now to be available at the appropriate times. Coursework for 12A will be weighted as follows:

Programming Assignments	60%
Midterm Exam 1	10%
Midterm Exam 2	10%
Final Exam	20%

Coursework and Evaluation for CMPS 12L:

We will also have nine or ten **Lab Assignments** dealing with various topics such as: the javac compiler, the submit command, the unix operating system, command line arguments, file input and output, the C programming language, the jdb debugger, program testing, Jar files, and graphical user interfaces. These assignments will be due at roughly one week intervals. Students taking 12L will also be required to sit for the 12A **Final Exam** (see above for time and place). Coursework for 12L will be weighted as follows:

Lab Assignments	80%
Final Exam	20%

The grading scale for both 12A and 12L will be approximately: A+::97%-100%, A::93%-96%, A-::90%-92%, B+::87%-89%, B::83%-86%, B-::80%-82%, C+::76%-79%, C::70%-75%, D::60%-69%, F::0%-59%. Letter grade boundaries may be lowered at my discretion in order to eliminate some borderline cases.

Academic Honesty:

The Baskin School of Engineering has a zero tolerance policy towards any incident of academic dishonesty. If cheating occurs, consequences within the context of the course may range from getting zero on a particular assignment, to failing the course. In addition to these sanctions, every case of academic dishonesty is referred to the students' college Provost, who sets in motion an official disciplinary process. Cheating in any part of the course may lead to failing the course and/or suspension or dismissal from the university.

What is cheating? In short, it is presenting someone else's work as your own. Examples include (but are not limited to) copying another student's program, allowing your own work to be copied, or in any way facilitating the cheating of others. Although you may discuss problems with fellow students, your collaboration must be at the level of *ideas* only. Legitimate collaboration ends when you "lend", "borrow", or "trade" *source code*, or in any way share in the act of *writing* your programs. You may freely give and receive help with the computer facilities, editors, the UNIX operating system, and the proper use and syntax of the C and Java programming languages; but you may not copy, paste, email, or in any way share *source code*. If you do collaborate (legitimately) or receive any form of help from anyone, you must credit them by placing their name(s) in your README file.

Please go to http://www.ucsc.edu/academics/academic_integrity/ to see the full text of the University's policy on Academic Integrity.