

## CMPS 12B & CMPS 12M

### Introduction to Data Structures

### Winter 2009

**Description:** Teaches students to implement common data structures and the algorithms associated with each data structure, through progressively difficult exercises. Topics include big “O” notation; pointers, recursion (induction), and dynamic allocation; linked lists and list processing; stacks, queues, binary trees and binary search trees; simple sorting techniques and simple search techniques. Students will gain a working knowledge of the elements of the Java and C programming languages. Prior experience with Unix is assumed. **Prerequisites:** CMPS 12A. Concurrent enrollment in CMPS 12M is required.

**Time and Place:** TTh 12:00 – 1:45 Thimann Lecture 001

**Class Webpage:** <http://www.soe.ucsc.edu/classes/cmcs012b/Winter09/>

**Class Webforum:** <http://forums.soe.ucsc.edu/>

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

**Office:** E2 257

**Office Hours:** MT 10:00 – 11:00am, WTh 2:00 – 4:00, or by appointment.

**Email:** [ptantalo@soe.ucsc.edu](mailto:ptantalo@soe.ucsc.edu)

**Phone:** 831-459-3898

#### Teaching Assistants:

Iryna Gordei <[igordei@ucsc.edu](mailto:igordei@ucsc.edu)>

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

#### MSI Tutor:

Daniel Key <[dkey@ucsc.edu](mailto:dkey@ucsc.edu)>

**Required Text:** *Data Abstraction and Problem Solving with JAVA* (second edition) Frank M. Carrano, Janet J. Prichard. Pearson 2006 (ISBN 9780321304285).

#### Supplementary Texts:

*Your Unix* (second edition) Sumitabha Das. McGraw-Hill 2006 (ISBN 978007250422).

*Java By Dissection* Ira Pohl, Charlie McDowell. Lulu.com 2006.

*C for Java Programmers: A Primer* Charlie McDowell. Lulu.com 2007.

#### Coursework and Evaluation for CMPS 12B:

We will have five **Programming Assignments**, due at roughly two week intervals. The **first Midterm Exam** will be held **Thursday, January 29**, and the **second Midterm Exam** will be held **Thursday, February 26**. The **Final Exam** will be held on **Tuesday, March 17 12:00 – 3:00 pm**. Please make arrangements now to be available at the appropriate times. Coursework for 12B will be weighted as follows:

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

### Coursework and Evaluation for CMPS 12M:

We will also have 9 or 10 **Lab Assignments** (which are really just mini programming assignments) dealing with various topics such as: makefiles, jar files and Java executables, command line arguments, file input and output, data abstraction, and information hiding in both Java and C. These assignments will be due at roughly one week intervals. Students taking 12M will also be required to sit for the 12B **Final Exam**. Coursework for 12M will be weighted as follows:

Lab Assignments	80%
Final Exam	20%

The grading scale for both 12B and 12M 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/](http://www.ucsc.edu/academics/academic_integrity/) to see the full text of the University's policy on Academic Integrity.