

CS 125 - Introduction to Computer Science
Spring 2009
Section 1208 2:40 - 3:50 MWF

Instructor: Jonathan Pikalek
Office: Ivers 234E
E-mail: pikalek@cord.edu
Web: <http://www.cord.edu/faculty/pikalek/>
Phone: 299-4237
Office hours: Tuesday 10:00am-12pm, 1pm-3:00pm
Thursday 10:00am-12pm, 1pm-2:00pm
Additional times available by appointment

Overview:

"The first course in the (computer science) major/minor sequence. An introduction to the Java programming language, algorithm design, structured and object-oriented programming techniques. No prior programming experience is assumed. *Prerequisite: higher algebra*" [1]

Methodology

Problem solving techniques will be introduced. Students will learn to analyze problem statements, develop corresponding models, convert their models to computer programs and analyze their solutions. Programming concepts, data structures and algorithms will be discussed. Several commonly used algorithms will be introduced and mathematically analyzed. Students will consider the use of computing to address topics and problems from a variety of disciplines.

Goals:

Preparation for continuing on to higher level computer science courses.
General knowledge about computing and information technology.
Basic programming and software development skills.
Develop strong problem solving skills.

Writing:

"In computer science, writing is broadly defined to include several types of activities, and this is reflected in student's engagement in writing in a variety of ways in computer science course." "Introductory computer science courses will place special emphasis on program code and documentation with a special eye on good program coding and documentation styles." [4]

Required Texts:

Understanding Computers Today & Tomorrow, 12th Ed by Morley & Parker.
Java Concepts, 5th Ed by Horstmann.

Attendance:

"Regular attendance and participation in class is critical to a student's success at Concordia College. Because any absence, excused or unexcused, detracts from the learning experience, students are expected to attend all classes." [2] If you are absent, you are responsible for learning materials covered during class. If you have a scheduled absence due to extracurricular activities, it is your responsibility to provide prior notice.

Grading:

Hardcopy assignments are due at the beginning of class on the date indicated by the schedule. Digital (Moodle) assignments are due as posted online. Late work will be subject to a 10% per day penalty. Programming assignments will be graded based on program correctness, clarity and documentation. Assignments will account for 45% of your overall grade.

Short quizzes will be periodically given as preparation for exams. Quizzes will be categorized as assignments for the purposes of grading weights.

Since later materials depend on earlier materials, exams are semi-comprehensive; however the emphasis will be on materials covered since the last exam. The fourth exam will be held during finals week along with a comprehensive final. Exams will account for 55% of your overall grade, meaning each exam and the final will contribute 11% to your final grade.

Makeup exams will only be allowed for situations arranged with the instructor **prior** to the exam (and only for a very compelling reason) or for an illness or injury verified by a doctor's statement. Generally speaking, makeup exams must be scheduled within one week of the mixed exam.

The tentative grade cut off scale is as follows:

94% A	90% A-	
87% B+	84% B	80% B-
77% C+	74% C	70% C-
67% D+	64% D	60% D-

Academic Honesty:

You are expected to read and understand Concordia's policies regarding academic integrity [3]. While students are both allowed and encouraged to collaborate on assignments (especially programming assignments), it is imperative that as an individual you each understand the material and concepts covered by an assignment. Since academic dishonesty is expected to be rare to non-existent, infractions will be handled in a case by case manner.

Special Needs:

If you require special accommodations in this course, please contact the Counseling Center in Academy Hall 106 (299-3514).

General Advice:

The best way to learn programming is to write programs. You are encouraged to work on additional exercises and personal projects. The textbook has self check exercises throughout the chapters – I recommend taking advantage of them to check your understanding before continuing on to harder material.

The problem solving skills learned here are useful in many disciplines beyond computer science. When working on problems don't focus solely on reaching the solution, actively think about your problem solving process as well.

Good software depends on more than code. Clear documentation and design is equally important; both will be factored into the grading of programming assignments.

Plan ahead: even experienced programmers are prone to underestimating the amount of time required to complete a project. Set aside ample time to code, test and debug your programs.

Later topics are highly dependent on early material. Falling behind early on will make learning subsequent material significantly more difficult. Look for help from myself, the tutor or your peers as early as possible.

Collaborate with your peers. There is rarely a single best way to write a program. Different perspectives lead to greater insight into the process of problem solving and programming.

Collaboration means more than copying. You must take responsibility for understanding the materials. After all, exams will be a zero collaboration environment and count for a significant portion of your final grade.

References:

[1] 2007-08 Academic Catalog:

http://www.cord.edu/academic/catalog/departments/c_scdept.course.html

[2] College Handbook: General and Academic Policies - class attendance:

http://www.cord.edu/student/handbook/class_attendance.php

[3] College Handbook: General and Academic Policies – Academic Integrity:

http://www.cord.edu/student/handbook/academic_integrity.php

[4] Concordia College Department of Mathematics and Computer Science – Writing in Computer Science.