

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 10:30am-12pm, 1pm-2:00pm
Additional times available by appointment

Overview:

A study of computer architecture and assembly language programming. Highlights include: machine language programming, assembly and linkage processes, interrupts, memory management, macros, graphics, and mouse programming. *Prerequisite: C SC 225 — Fundamental Structures*[1]

Goals:

Preparation for select high level computer science courses
Assembly programming and development skills
Working knowledge of computer architecture
Familiarity with structured programming
Understanding program optimization tradeoffs

Required Texts:

Assembly Language for Intel-Based Computers, 5th Edition, by Kip Irvine.
Write Great Code Volume I: Understanding the Machine, by Randall Hyde.

Optional Texts

The Art of Assembly Language, by Randall Hyde (freely available online)

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.

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 nonexistent, 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:

Most students will spend little to no time programming assembly outside of this class. However, familiarity with assembly and the low-level operations of computers can significantly improve your ability to design and code in high level languages. Try to keep this connection in mind in order to get the most out of this course.

The best way to learn programming is to write programs. You are encouraged to work on additional exercises and personal projects.

Strong problem solving skills are useful in many disciplines, including computer science. When working on problems don't just focus solely on 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.

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.

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