CSC 160-02 Fall 2008
Computer Programming for Non-Majors
Prof. A. Wittenstein

Contact Information

Office: Alumnae Hall 114
Phone: 877-4486
Course Web Page: http://www.adelphi.edu/~wi16133/csc160/f08
Email: Wittenstein@adelphi.edu

Office Hours

MW 5:30-6:00pm. Other days/times by appointment.

Class Meetings

MW 4:15-5:30pm, Science 227 (M 8/25 -> W 12/10)
*The final exam will be on Weds 12/17 from 3:30-5:30pm. The location will be posted on Moodle in advance of the exam date.

Course Description

There are no prerequisites, as this course is intended for students with little or no experience in computer programming. It gives students a feel for what programming is like, introduces the process of program development, and introduces the major concepts of programming --- variables, data types, functions, parameters, assignment statements, conditionals, compound data types such as structures, lists, and arrays, and repeating constructs such as loops and recursion.

Course Expectations

Although the course has no prerequisites, and is considered suitable for General Education distribution requirements, it is nonetheless hard work. Lectures will not cover everything you need in order to complete the homework assignments; you need to complete and understand the reading assignments as well. There will be homework assignments every week or two, most of which will require hours of programming, either in a computer lab or at your home computer.

Since this course meets for three credit hours per week, it is expected that on average you do at least 3-6 hours of work for this course per week outside of class time. Please budget this time to ensure that homework is completed on time, and that all reading assignments are read *before* the class session covering those sections.

The class meetings will highlight the important parts of the material, but there may not be time in class to cover all of the material in each reading assignments in detail. Therefore, having access to, *and reading*, the textbook and other assigned readings is an essential component of this course.

It is known that especially in beginning programming courses, it is of the utmost importance for students to *keep up with the readings and assignments*. Students who fall behind tend to *stay* behind, and either drop or fail. I don't want anybody to drop or fail; if you fear that you're falling behind, please talk to me as *soon* as possible.

Moodle Learning System

--To access Moodle, log on to your ecampus account then click on the Moodle tab.
--All grades (for assignments, quizzes, and exams) will be posted to Moodle.
--All lecture slides, assignments, and assignment solutions, will be posted to Moodle.
--Each assignment on Moodle will include a link to be used to submit the assignment.
--We may also use other Moodle features, such as forums.

Submitting Assignments

--All homework assignments are to be submitted on the announced due date either in class or via the Moodle Learning System by 11:55pm. Ten points will be taken off per day (or portion thereof) for late assignments. Late assignments will not be accepted after they have been reviewed in class for any reason.
--Instructions for submitting classwork assignments will be given in class.

Assignment Standards

The first line of every assignment must contain a comment indicating the name(s) of the student(s) working on it and which assignment it is. Assignments not containing this information, clearly visible, will get a zero.

Many of the homework assignments will involve programming. I realize that sometimes you hit a brick wall and cannot get a program working. If this happens, turn in the program together with a detailed description of how the program fails, what you've tried in your attempts to fix it, and how those attempts didn't succeed. You won't get full credit, but if I'm convinced that you're working on it diligently, you'll likely get partial credit.

Ethics

Most homework assignments in this course involve writing, testing, and debugging one or more programs. For each of these assignments, you are to work either alone or in teams of two students, switching teams from one assignment to the next (so you get experience working with different people). When I say "teams of two students", I don't mean that you each do half the questions. I mean working *together* on *all* of the assignment.

Students are encouraged to help one another with difficulties like ("How do I save this file?", "what's the syntax for `define-struct`?", *etc.*), regardless of whether they're on the same team, but designing, coding, testing, and debugging must be done solely by the person(s) whose names are at the top of the assignment.

It's remarkably easy for me to notice when three different teams have turned in nearly-identical programs; if that happens, I'll grade it once and divide the credit among the three, so the best any of them can hope for is 33%. I don't try to figure out who copied from whom; it is *your* responsibility to not let anyone copy your homework. Among other things, that means don't leave it on a campus computer, because anyone at Adelphi can copy it and even delete it.

Attendance

Attendance is required. After four absences, your grade will be lowered by one-third of a grade (e.g., A to A-, A- to B+, etc.). You are also responsible for whatever work is covered in class ***whether or not you are there***. Absence from quizzes, the midterm and the final exam will be excused only for a good and **well-documented** reason. The decision to allow a make-up exam will be made in accordance with the policies of Adelphi University. Please arrive to class *on time*. Also, I will count latenesses (or leaving early) as partial absences.

Texts

--***An Introduction to Computer Programming***, by Dr. Stephen Bloch, August 2008 draft. (Chapters can be found on Moodle under "Textbook".)
--Links to additional readings will be provided through Moodle or the class web page.

Grading

Assignments	40%	
Quizzes	10%	
Midterm Exam	20%	(*Tentative* date: Weds. 10/22)
Final Exam	30%	(Weds. 12/17 3:30pm-5:30pm)

***Tentative* Schedule of Topics**

I. Running and Writing Programs
Chapter 1: Drawing pictures in DrScheme
Chapter 2: Shorthand for values: Variables
Chapter 3: Building more interesting pictures
Chapter 4: Writing your own functions
Chapter 5: A recipe for defining functions
Chapter 6: Animations in DrScheme
Chapter 7: Working with numbers
Chapter 8: Working with strings
Chapter 9: Reduce, re-use, recycle

II. Making Decisions
Chapter 10: Booleans and decisions on strings
Chapter 11: Decisions involving numbers
Chapter 12: Decisions among data types
Chapter 13: Symbols, characters, and event-handling
Chapter 14: Handling Errors

III. Complex Data
Chapter 15: Animations and Posns
Chapter 16: Inventing New Structures
Additional Topics: Lists, arrays, loops, recursion, assignment statements