

CSC-281 - Introduction to Computer Science II

Fall, 2011

Instructor: Michael Black
Email: mblack@american.edu
(please include "CSC-281" in the subject line of your emails to me)

Recommended Textbooks:

"Big Java", Cay Horstmann
A book on Android programming:
(possibly "Android Application Development for Dummies")

Times: Mondays and Thursdays, 4:00-5:15
Anderson Computing Center, B12

Office hours: Tentatively:
Mondays & Thursdays, 1-2:30
Wednesdays, 1-4

My office is SCAN 164, and my office phone number is (202) 885-2011.

Prerequisite: You are expected to have taken CSC280 or the equivalent and understand the basics of programming in a higher level language (such as Java or Python). You should be familiar with classes and objects, methods and functions, loops, conditionals, arrays.

Course Description: This course introduces object-oriented programming, including class structure, inheritance, and polymorphism. It will introduce some elementary data structures and algorithms, event-driven GUI programming, and multithreaded programming. The Java programming language is introduced, both for the desktop and for Android mobile devices.

Projects: There will be approximately 3-6 Java projects that will make up the bulk of your grade. You will normally have between one and two weeks to complete each project. Projects will be graded on correct functionality, coding style, comments, and creativity. Project submissions that do not compile or do not achieve a level of minimum functionality will not typically be accepted.

Homework: Smaller programs and miniprojects may be assigned from time-to-time throughout the semester. These will generally be graded on correct functionality only.

Grading:

Projects:	60%
Homework / Quizzes:	10%
Midterm:	15%
Final:	15%

100-93 A 92-90 A- 89-87 B+ 86-83 B 82-80 B- 79-77 C+ 76-73 C 72-70 C- 69-60 D 59-0 F

Late Policy: Most assignments will be submitted electronically on Blackboard, and will be due at 11:59 pm on the date given in the assignment. Assignments must be submitted by the due date to receive full credit, unless I approve prior arrangements. Projects and homework received after the deadline will incur a 50% lateness penalty. No work received after the beginning of the course's scheduled final exam will be accepted for credit.

Missed exams may not be made up. If a true emergency forces you to miss an exam, permission to be excused must be sought in writing from me in advance, if possible. If granted, the final exam score will replace the score of the missed exam (or the midterm score, if the final is missed).

Attendance Policy: Class attendance is necessary in order to properly learn this material. Projects will be based on concepts covered in class, and not all material covered in class will be available in the textbooks. If you are unable to attend a lecture, you are responsible for finding out what material you missed and learning it.

Academic Integrity:

Plagiarism and academic misconduct are defined in the University's Academic Integrity Code. You should be familiar with what constitutes academic dishonesty. In particular, you should observe the following rules:

- Collaboration on projects is forbidden. Any information taken from the internet, books, or anywhere else for use on your assignments must be cited. You are permitted to discuss your work with other students at the conceptual level only.
- Your code must be entirely your own. You are not permitted to share code or use code obtained from the internet in your work.
- Unless otherwise stated, all exams will be closed-book, closed-notes.
- Instances of plagiarism may be reported and could result in disciplinary action.

Feedback:

I am eager to receive suggestions on how I can improve this course. If you have any comments, please feel free to tell me, either in person, by email, or anonymously.

Approximate Course Outline (subject to modifications)

Date	Subject	Readings (Horstman, 3rd ed.)
<i>Object-Oriented Programming</i>		
M 8/29	Crash course in Java: Hello World	17-27
Th 9/1	Crash course in Java: variables, structures, methods	34-37, 134-267
Th 9/8	Crash course in Java: arrays, classes and objects	288-292, 37-58
M 9/9	Crash course in Java: methods, instance variables	82-112
Th 9/12	Inheritance	438-462
M 9/19	Polymorphism	485-495
Th 9/22	The Java GUI (frame, components)	58-65
M 9/26	The Java GUI (listeners)	338-421
Th 9/29	Multithreaded programming	846-864
M 10/3	Locks	
Th 10/6	The Android API	
M 10/10	Android programming	
<i>Algorithms</i>		
Th 10/13	Recursion	588-617
M 10/17	Recursion (continued)	
Th 10/20	Midterm (tentatively)	
M 10/24	Running time and complexity	
Th 10/27	Running time (continued)	
M 10/31	Selection sorting	628-639
Th 11/3	Merge sorting	639-649
M 11/7	Searching	649-654
<i>Data Structures</i>		
Th 11/10	Linked lists	666-685
M 11/14	Stacks, queues	686-689
Th 11/17	Binary trees	720-731
M 11/21	Tree traversal	731-735
M 11/28	Graphs	
Th 12/1	Graphs (continued)	
<i>Misc (time permitting)</i>		
M 12/5	Socket programming	888-909
Th 12/8	Sockets (continued)	