

CSC-280 Introduction to Computer Science I

Credit:	Most of this syllabus comes from Dr. Michael Black (but errors are my own).
Instructor:	Dr. Serge Kruk
Email:	kruk@american.edu
Office:	SCAN 112
Office hours:	Monday 1500h-1700h Friday 1200h-1600h
Textbook:	Practical Programming: <i>An Introduction to Computer Science Using Python</i> You can get a paper copy at Amazon for less than twenty dollars. An electronic version is also available from the publisher.
Additional:	python.org/doc/
Times:	MWR 1145h-1300h Andersen B-12 MWR 1310h-1425h Ward ST01
Pre-requisite:	That you have used a computer before.

It goes against the grain of modern education to teach children to program. What fun is there in making plans, acquiring discipline in organizing thoughts, devoting attention to detail and learning to be self-critical?

– Alan Perlis, Epigrams in Programming

Classes: Programming is very much a skill learned by reading programs and writing programs. We will be spending much of the lecture time going through examples. Although we will meet in a computer lab, you are highly encouraged to bring your own laptop to class if you have one.

Attendance: Attendance, though not absolutely mandatory, is necessary for most students to do well in this course. (In addition, my lectures are great!) If you miss class it is your responsibility to learn the material covered that day on your own. This should involve finding out what example programs were covered in class that day and understanding them on your own.

Computers and Software: If you do not have a computer, computers are available during the day in the MCK11 computer lab and at other locations on campus. If you do own a computer, you must install the python interpreter from www.python.org. (As with all great software it is free.)

Projects: There will be around 10 programming projects that will make up the bulk of your grade. You will have between two hours and two weeks to complete each project; the length depends on the project complexity. You can do the projects alone

or in a group of two. Extensive academic and industrial experience shows that Team Programming is one of the most effective programming techniques. Here are the rules for team work: For each project done by a team, one student is the Scribe, the other student is the Overseer. A group of students must submit at least two projects and the roles of Scribe and Overseer must alternate on consecutive projects. The Scribe will be at the keyboard and will write *every single line of code*; the Overseer is not allowed to touch the keyboard. Other than that students share responsibilities.

The grading rubric is as follows. Note that you get 60% just for submitting something minimally satisfying the requirements but to get 100% you need to show creativity. Students in a group get the same grade.

Grading:

You submitted something, on time.	10%	Projects:	65%
There are no syntax errors.	+20%		
There are no semantic errors (bugs).	+20%	Midterm:	15%
The code includes and passes test cases.	+20%	Final:	20%
The functions are sufficiently efficient.	+20%	Total:	100%
The program is elegant.	+10%		
Total:	100%		

Conversion to letter grade:

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: Late projects will not be accepted, unless permission is sought before the project is due. Missed exams may not be made up. If a true emergency forces you to miss an exam, permission to be excused must be sought 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).

Academic Integrity: Plagiarism and academic misconduct are defined in the University Academic Integrity Code. You should be familiar with what constitutes academic dishonesty. In particular, you should observe the following rules: *Collaboration on projects is restricted.* 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 outside of your group at the conceptual level only (that means not in front of a computer and not while taking notes). Your code must be entirely your own work (or your group's work if working in a group). 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.