Introduction to Computer Science, CSC 280, Fall 2014

Instructor: Prof. Bei Xiao, American University.
TA: TBA
Contact: bxiao@american.edu, Office: SCAN 110.
Lecture Time: Mon/Wed/Thu 2:30-3:50pm
Location: Anderson B-12
Office hours: Wed 4-6pm or by appointment (SCAN 110) or by email appointments.
Website: TBA

Textbooks and references:
We will use the textbook mainly as a reference, but we will not strictly follow any textbook.

1. How to think like a computer scientist, learning with Python. Allen Downey. The online version of the book is here: http://www.openbookproject.net/thinkcs/python/english3e/

2. Introduction to Computation and Programming Using Python. John Guttag. Spring 2013 edition. MIT Press. (recommended, not required).

3. The official Python tutorial. https://docs.python.org/2/tutorial/

Computer and software:
Even though computers are provided in the classroom, you are extremely encouraged to bring your own laptop.

Perquisite: This course is aimed at students with little or no programming experiences. Interests in solving simple math and high school algebra would be useful.

Course Description:
This course serves as an introduction to computer programming and art of computation to students with little background in programming. Very much like learning a language, the goal of computer programing is to transfer thinking into logics and computations. Teaching is mostly achieved in in-class examples, discussions, quizzes, and programming exercises. In the first part of the class, students will learn basic grammar and programming methodologies (such as syntax, data types, conditionals, iterations, classes, libraries) with Python. As the class advance, we will introduce simple algorithms such as sorting and memorization. Computational concepts and techniques such as exhaustive enumeration, bisection search, dynamic programming, and efficient approximation algorithms will also be introduced.

Expected Learning outcomes:

1. Having understood fundamental concepts computer programming such as control flow, data structure, and abstraction.
2. Having acquired good programming skills and understand good programming practices.
3. Being exposed to concepts of involving computation in everyday thinking.
4. Being able to solve simple problems with Python.

5. Having built solid foundation for object-oriented programming that will be helpful in learning other programming languages and advanced courses in computer science.

Grading: 50% Homework Assignments, 20% in-class Mid-term exams (two mid-term exams), 20% Final project/exam, 10% in-class participations (5% in-class quizzes and discussions, 5% showing-up).

A final letter grade will be converted from the percentage you receive through out the course.

Grading Scale listed below:

94-100	A	
90-93	A-	
86-89	B+	
81-85	B	
76-80	B-	
70-75	C+	
65-69	C	
60-64	C-	cut off for CS major in order to receive credits for major requirement.
50-59	D	
0-50	F	

Homework Policy:

1. Homework is all about programming and you will submit to blackboard. In the first class, I will pass out a handout code template of how to comment, name, and structure your code for homework assignment. Please follow the instructions strictly. If you name your code "homework question1" without your name, it will not be graded. You must test your code on your computer. We will use libraries extensively.
2. No plagiarism (see Academic integrity below): only high-level discussions are allowed (i.e., not relating to a single line of code.
3. Late penalty: the deadline of HW is the **midnight eastern time** of the due date. You can turn in **one** HW late for up to 48 hours with no penalty. The second/third late HW will automatically receive **50% reduction if received within the 48 hours after the deadline**. After the 48 hours, the homework will not be graded and will receive zero points.
4. Grading: 80% correctness, 10% efficiency, and 10% style. Make sure your code works! It must pass sample I/O. Only having correct logic but not correct output will receive reduced points.
5. Please use **Python 2.7.** as most course demos will be done in Python2.7 so is your textbook. If you use Python 3, please note that it is not backward compatible with Python 2.x.
6. Please restrict homework related questions to office hours. If you are confused of what the homework is about, ask me last least 3 days before the due date. Last-minuet emails regarding homework might not promptly answered.

Exam Policy:

Mid-term exams will be announced at least one week ahead of time. If you have special needs, you need to notify me at least 5 days before to arrange the test be performed off-class in the exam center. Missed exams cannot be made up.

Attendance policy:

Significant amount of work is demoed in class as live programming and lectures. You are expected to regularly attending classes.

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 (if you have high-level discussion with another person, please write down the name of the person). Any information taken from the Internet, books, or anywhere else for use on your assignments must be cited. Your code must be entirely your own work. All exams will be close-book, close-note, no Internet, no smart phones. Instances of plagiarism may be reported and could result in disciplinary action.