1. Basics

Course details:	CS 111 (Introduction to Computer Science); Fall 2016, Carleton College
Meeting times:	5a (MW 13:50–15:00, F 14:20–15:20); CMC 102
Instructor:	Jed Yang, CMC 324, x4473, `jyang@carleton.edu`
Office hours:	Mon. 15:00–16:00, Thr. 14:30–15:30, Fri. 13:00–14:00; or by appointment
Prefect:	Aditya Subramanian (`subramaniana@carleton.edu`)
Webpage:	`http://cs.carleton.edu/faculty/jyang/cs111.16f/`

2. Course information

Official course description. This course will introduce you to computer programming and the design of algorithms. By writing programs to solve problems in areas such as image processing, text processing, and simple games, you will learn about recursive and iterative algorithms, complexity analysis, graphics, data representation, software engineering, and object-oriented design.

Prerequisites. *Officially:* There are no prerequisites. No previous programming experience is necessary. If you have significant coding experience (*e.g.*, you took a course in high school, even if you didn't take an AP exam), please come talk to me about whether CS111 is the right course for you.

Course goals. In this course we will focus on problem solving and formulating problems in a computational way. This allows us to then develop algorithms, or step-by-step procedures, for solving these problems and then analyze how well these algorithms perform. Over the course of the term, we'll be looking at a variety of computer science applications, including manipulating images, processing text, and playing games. We'll be learning the programming language Python as a consequence of our exploration of fundamental computer science ideas. By the time you've completed the course, you will be able to:

- Break down a task into a series of steps (an algorithm) that solve the task.
- Read and understand simple programs (or algorithms) written by others.
- Design, implement and test projects in Python that can be run and understood by others.

Textbook. *Python Programming: an introduction to computer science*, 2nd edition, John N. Zelle, 2010.

3. Course requirements

Attendance. I expect you to attend class. You may not notice me taking attendance during class meetings, but I will notice if you are not in class. Occasional absences will not impact your grade because what I look for is not mere attendance, but engagement and participation.

Indeed, coming to class is not just about showing up; it is also about being fully engaged in the learning experience. If you have a question, others in the class may also be wondering the same thing. So, please speak up and ask questions anytime you need to. Not only will you be helping yourself, but also you will be helping your peers. Attending office hours is another great opportunity to ask questions.

Be mindful of others. Refrain from using mobile phones or laptops for activities unrelated to the learning process. These may distract others (and yourself).

Homework. There will be homework assignments due approximately twice a week (sometimes more, sometimes less). These assignments will usually be in the form of writing a Python program to complete a particular task. Some assignments will be **solo assignments**, meaning you can discuss the project with others, but you are responsible for writing the code on your own. Other assignments will be **pair programming assignments**. For these assignments, I may assign partners. You will switch partners several times throughout the term. I strongly encourage you to start on homeworks early. Typically, assignments are due at 23:55.

Exams. There is one in-class midterm exam (tentatively scheduled for **Wednesday, October 12, 2016**) and a take-home final exam (tentatively handed out **Monday, November 7, 2016** and due the following week). If you need accommodations for test-taking because of a disability, please contact me early and we will make appropriate arrangements.

Final Project. There will be a final project due at 17:00 on **Monday, November 21, 2016**. These projects will be completed in groups of two. You will demo your project during our final exam slot, 15:30–18:00 on **Sunday, November 20, 2016**, so you must be on campus that afternoon.

Time Outside of Class. I expect that you should be spending about 10–12 hours per week on this course outside of class. Some students need to spend a bit more than that (which is okay). If you are spending more than 15 hours per week on this course outside of class time, please come talk to me so we can find ways to help you learn the material without spending so much time.

4. How to get help

If you need help there are are multitude of resources you can use:

(a) **Yourself.** If you're stuck on a problem or struggling with a concept from class, take a break and think about something else (*e.g.*, your Greek assignment, the economics of Star Trek) for a few hours and then try a fresh start.
(b) **The book.** There is a lot of really useful information in it. Peruse it at your leisure.
(c) **Your classmates.** You are each other's best resource: talking through the course material with someone else who is also trying to master it is a great way for you both to learn. (And don't discount the learning that you will do while trying to explain to a classmate an idea covered during class that you think you understand; I can't count the number of times that I've discovered that I didn't really understand something until I tried to teach it to someone.) The homework assignments are meant to challenge you, and figuring some of them out together is a great approach.
(d) **The prefect.** Attend a prefect session to get help with homework or general concepts.
(e) **The instructor.** Come to my office hours or email to make an appointment. (Please include a list of a few times that you will be free to meet, and give me at least 24 hours of lead time.) I will consistently reserve Tuesdays for research, and I do not schedule office hours or make appointments for that day. I have this scheduled "research day" so that I can work on my research projects in an uninterrupted block of time. Without reserving a large block, I won't have time for any research.
(f) **College-wide resources.** The library, the Academic Skills Center, the Math Skills Center, the Writing Center, the CS lab assistants (in CMC 102, 304, 306), *etc.*

5. Grading

Your grade will be determined by a weighted arithmetic mean of various components, with weights listed in the table on the right. There is no standard percentage that I associate with a particular letter grade (`A`, `B`, `C`, etc.). Instead, I decide on letter grade cutoffs by comparing each student's overall score to my understanding of the Platonic ideal of an x student (for $x \in \{\texttt{A}, \texttt{A-}, \texttt{B+}, \ldots\}$). To make the grades robust to small noise, I also look for large numerical gaps in the sorted list of scores when setting grade cutoffs. Note that you should therefore not care how difficult the exams (or problem sets) are: the Platonic `A` student earns fewer points on a more difficult exam than she does on an easier exam. There is also no preset curve of how many of each letter grade will be given. As such, you are encouraged to help each other in the pursuit of perfection. Feel free to talk to me if you are concerned about your standing in the class, with the understanding that given the nature of the aforementioned grading process, it is impossible to accurately predict your course grade before the final exam.

component	weight
Homework	40%
Midterm	20%
Final Exam	25%
Final Project	10%
Participation	5%

6. Homework policy

Collaboration. Many of the assignments in this class will be collaborative. You can learn a lot from working directly with your peers. However, make sure that you are familiar with the collaboration policy outlined at the end of this syllabus. Working with a partner does not mean just dividing up the work. You are responsible for knowing and being able to explain any part of an assignment that you turn in. For solo assignments, you can certainly discuss the project with your classmates, but the code you turn in must be done by you alone.

Partners. I may assign partners for pair programming assignments. In paired programming, you DO NOT simply divide up the work and work separately. Instead, you and your partner are always working together, side by side, with one of you typing ("driving") and one of you looking on and providing feedback ("navigating"). You will switch between these two roles often. Paired programming generally leads to better designed code with fewer errors. Please look at the course website for more details about paired programming.

Grading. Each assignment will include some information about the number of points the assignment is worth and how those points will be allocated. If you have a question about how a homework assignment was graded, please start by talking to the grader. If you cannot resolve the question, you may request that I look at the question.

Late work policy. You are allowed up to three (3) late assignment-days throughout the term for homework. (A **day** is 24 hours, regardless of weekends and holidays.) This allotment is to cover for legitimate reasons for tardiness that may arise. No explanation for the tardiness is necessary or desired, but please do inform me that you are submitting an assignment late. After the freebies, work handed in late will receive zero credit. If you use a late day for a partnered assignment, you both use a late day. If you wish to use more than one late day for a single assignment, please discuss with me first. If a genuine emergency situation arises, please talk to me.

7. Academic honesty and collaboration policy

Collaborative work is an integral part of many successful ventures. As such, I expect that you should collaborate with your classmates a lot during your time in this course. However, it is important to understand that there is a big difference between thinking about and solving a problem as part of a group (which is good) and copying an answer/code or letting someone else copy your answer/code (which is bad). Below are a few specific examples of unacceptable behavior in this course:

- Modifying someone else's code and putting your name on it.
- Having a friend debug your code and then turning in the revised code.
- Asking a homework question on a forum and then turning in the answer as your own.
- Seeking out resources from past versions of this course or similar courses offered elsewhere.

In short, ***I trust you to maintain the utmost level of academic integrity in this course.*** Please do not break this trust; if you do, there will be repercussions. The formal policy below lays this out explicitly, and supplements the College's academic integrity policy and the Dean of the College's detailed guide to academic integrity.

Collaboration policy: You may collaborate on the homework assignments to the extent of formulating ideas as a group, but you may not collaborate in the actual writing of solutions/code (unless explicitly allowed in the instructions). In particular, you may not work from notes taken during collaborative sessions. You *must* cite all sources, including websites and classmates from whom you obtained ideas. You may not consult any materials from any previous offerings of this course or from any other similar course offered elsewhere.

You are required to completely understand any solution that you submit and, in case of any doubt, you must be prepared to orally explain your solution to me. If you have submitted a solution that you cannot verbally explain to me, then you have violated this policy.

Of course, there is to be no collaboration whatsoever on any exams. Policies for what constitutes acceptable reference material, if any, will be specified in detail when the exam is distributed.