1. Basics

Course details:	CS 251 (Programming Languages); Spring 2018, Carleton College
Meeting times:	3a (MW 11:10–12:20, F 12:00–13:00); CMC 301
Instructor:	Jed Yang, CMC 324, x4473, `jyang@carleton.edu`
Office hours:	Mon. 16:20–17:20 (in CMC 306), Thr. 14:35–15:35, Fri. 13:00–14:00; or by appointment
Webpage:	`http://cs.carleton.edu/faculty/jyang/cs251.18s/`

2. Course information

Official course description. What makes a programming language like "Python" or like "Java"? This course will look past superficial properties (like indentation) and into the soul of programming languages. We will explore a variety of topics in programming language construction and design: syntax and semantics, mechanisms for parameter passing, typing, scoping, and control structures. Students will expand their programming experience to include other programming paradigms, including functional languages like Scheme and ML.

Prerequisites. *Officially:* CS 201 (Data Structures). *Unofficially:* Comfort with programming in Python and Java, particularly recursion and simple data structures; willingness and ability to work in a team; and patience. If you're concerned about your background, please come talk to me in the first week.

Course goals. The elevator-pitch version of describing this course is as follows: It is about the linguistics of programming languages. We will look at different programming languages and think about how they differ, why they might have the features that they do, and what the implications of those differences are.

The intent of this course is twofold. First, I hope to entice you to think more deeply about why your favourite programming language "looks the way it does"—and also what it means for a programming language to look like anything. Second, you will get a chance to explore some other classical programming paradigms (other than object-oriented programming). These other paradigms—we'll focus on functional languages like Scheme and imperative languages like C—are both historically and conceptually central to modern CS.

By the end of the term, you will be comfortable programming in two new programming languages (Scheme and C) in two non-object-oriented programming paradigms. You will also have completed what will likely be the largest programming project that you've ever written, and gain experience both with thinking about conceptually bigger projects and with working in a team. You will also understand some of the key features and tradeoffs that undergird programming languages that you've seen in the past. And, if you're like me, by the end of the term, you will have at least one bizarre recursive dream, possibly about recursion.

In terms of major milestones/tasks, this is a three-step course: you will (i) learn Scheme, (ii) learn C, and (iii) implement a Scheme interpreter in C. Throughout, we will also spend time on various aspects of programming language design and structure: variable scope, parameters, types, memory management, *etc.* The structure of the topics is flexible. We will discuss topics as they arise naturally or when there is a lull in the project that gives us a chance for a diversion.

The term project is team based. We will talk about how to work as a team, but the single most crucial point is that your team must commit to each other, period. You will learn together, laugh together, struggle together, succeed together.

Textbook. There is no required text for the course. There are three optional books, which are available on reserve in the library:

- Brian Kernighan and Dennis Ritchie, *The C programming language* (aka "K&R").
- R. Kent Dybvig, *The Scheme programming language.*
- Michael Scott, *Programming Language Pragmatics.*

K&R is a classic reference; I recommend you consider getting (and keeping) it. Dybvig will be central to the course; I would have required it were it not freely available electronically. Scott is an advanced textbook for a year-long course and, as such, it contains a wealth of information, mostly beyond the scope of this course. We will fill in other readings (*e.g.*, book chapters and Wikipedia) as appropriate.

3. Universal Learning

I am committed to the principle of universal learning. This means that our classroom, our virtual spaces, our practices, and our interactions will be as inclusive as possible. Mutual respect, civility, and the ability to listen and observe others carefully are crucial to universal learning.

Carleton College is committed to providing equitable access to learning opportunities for all students. The Disability Services office (Henry House) is the campus office that collaborates with students who have disabilities to provide and/or arrange reasonable accommodations. If you have, or think you may have, a disability (*e.g.*, mental health, attentional, learning, autism spectrum disorders, chronic health, traumatic brain injury and concussions, sensory, or physical), please contact Chris Dallager, Director of Disability Services, by calling 507-222-5250 or sending an email to `cdallager@carleton.edu` to arrange a confidential discussion regarding equitable access and reasonable accommodations.

4. Course requirements

Attendance and participation. 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. If you prefer to use laptops to take notes, please kindly sit in the back, as the screen may distract others.[1] There is research that suggests taking notes by hand is better for long-term retention.[2]

Homework and term project. There will be stand-alone assignments, particularly at the beginning of the term. There will be mileposts for the term project due regularly throughout the term. Typically, assignments are due at 22:00.

You are allowed up to three (3) late days throughout the term for homework and two (2) late days for project mileposts (but not the last project submission). A *day* starts when the assignment is due and ends 24 hours later. Each late day is indivisible: turning in work a few minutes late will consume one late pass. A late team assignment costs each student a late pass. 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 (in person in class is fine) that you are submitting an assignment late. After the freebies, work handed in late will receive zero credit. You may not use more than one late day for a single assignment unless you talk to me first. To be fair to everyone in the class, I will generally not grant additional extensions without the intervention of a doctor or class dean. But if a genuine emergency situation arises, please talk to me.

Exams. There are three in-class midterm exams, tentatively scheduled for Friday of Week 3, Friday of Week 6, and Wednesday of Week 10. The exams are cumulative, but will focus on material not yet tested. (There is no final exam, but I reserve the right to replace the third midterm with a final exam.)

Time outside of class. Like other Carleton courses, 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.

---

[1] See `http://tinyurl.com/laptops-degrade-nbrs-grades`.

[2] P. A. Mueller and D. M. Oppenheimer, The pen is mightier than the keyboard: advantages of longhand over laptop note taking, *Psychological Science* **25** (2014), 1159–1168.

Illness. You should make every effort to attend class when you are healthy. If you become ill, for your well-being and the well-being of the rest of the class, you should not come to class. (Nor should you show up in my office with your germs!) Yes, this sounds like common sense, but it is tempting to try and power through as normal so as not to fall behind, particularly at a place like Carleton. If you become ill, or know that you will need to miss class for some reason, please contact me as soon as you are able, and we will work together to plan how you will keep up and/or make up any missed work.

5. Grading

Your grade will be determined by a weighted arithmetic mean of various components with weights listed in the table on the right. In more detail: (a) I record a numerical score for each component (possibly adjusted so that different components are on comparable scales). (b) For each student, I calculate the weighted arithmetic mean. (c) I sort these overall scores in descending order. (d) Except in very unusual circumstances (*e.g.*, medical or other personal-life issues that affected a particular score) I assign letter grades strictly based on this sorted order.

component	weight
Term Project	21%
Contribution	10%
Homework	15%
Exam 1	18%
Exam 2	18%
Exam 3	18%

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 therefore *you should not care how difficult the exams (or homework assignments) are.* Indeed, the Platonic `A` student earns fewer points on a more difficult exam than she does on an easier exam. In fact, in many courses I intentionally make one exam harder than the others, which gives me information (in a mathematical sense) in separating an `A` performance from an `A-` performance.

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.* If you all do `A` work, you will each get an `A`. 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.

Contribution. This component is based on your contribution to the learning of your classmates in general (classroom participation, Moodle involvement, *etc.*) and to the success of your partners. *Peer Evaluations:* Because you will be working in teams, I will ask you to give honest and thorough feedback about the contributions of you and your teammates. Providing an honest appraisal of your peers is difficult, but it's also important. Your contribution to your team, as measured both by peer evaluations and by my observations, will be part of your contribution grade.

6. How to get help

If you need help there 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) **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.

(c) **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. Thursdays 11:30–16:00 are usually good times for me.

(d) **The Moodle forum.** Personal questions (*e.g.*, regarding grades) should be elsewhere, but all other questions (*e.g.*, course content, homework, logistics) should go here rather than sending me email. I will periodically respond on the forum, but also please help answer each other's questions! While I try to answer queries about homework within 24–48 hours, I cannot always make this timeline, and you should not rely on faster responses than this. Among other things, this means that you would be well advised to ask any questions of clarification earlier than the day before an assignment is due.

(e) **College-wide resources.** The library, the Academic Skills Center, the Math Skills Center, the CS lab assistants, especially the ones in CMC 306, *etc.*

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, both educationally and morally) and copying an answer/code or letting someone else copy your answer/code (which is bad, educationally and morally, and has punitive consequences). Below are a few specific examples of unacceptable behaviour in this course:

- Modifying someone else's code and putting your name on it.
- Asking a homework question on a forum and then turning in the answer as your own.
- Looking at code samples (except the textbook) that are longer than a few lines.
- 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 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.