

Welcome to CSCI 134!

Introduction to Computer Science

What is Computer Science?

[Hint. It is not really about computers!]

“[Computer science] is not really about computers -- and it's not about computers in the same sense that physics is not really about particle accelerators, and biology is not about microscopes and Petri dishes...” — [Hal Abelson](#)

“The topic became – primarily in the USA – prematurely known as ‘computer science’ – which, actually, is like referring to surgery as ‘knife science’ – and it was firmly implanted in people’s minds that computing science is about machines and their peripheral equipment.” — [Edsger Dijkstra](#)

What is Computer Science?

[Then what is/could it be about?]

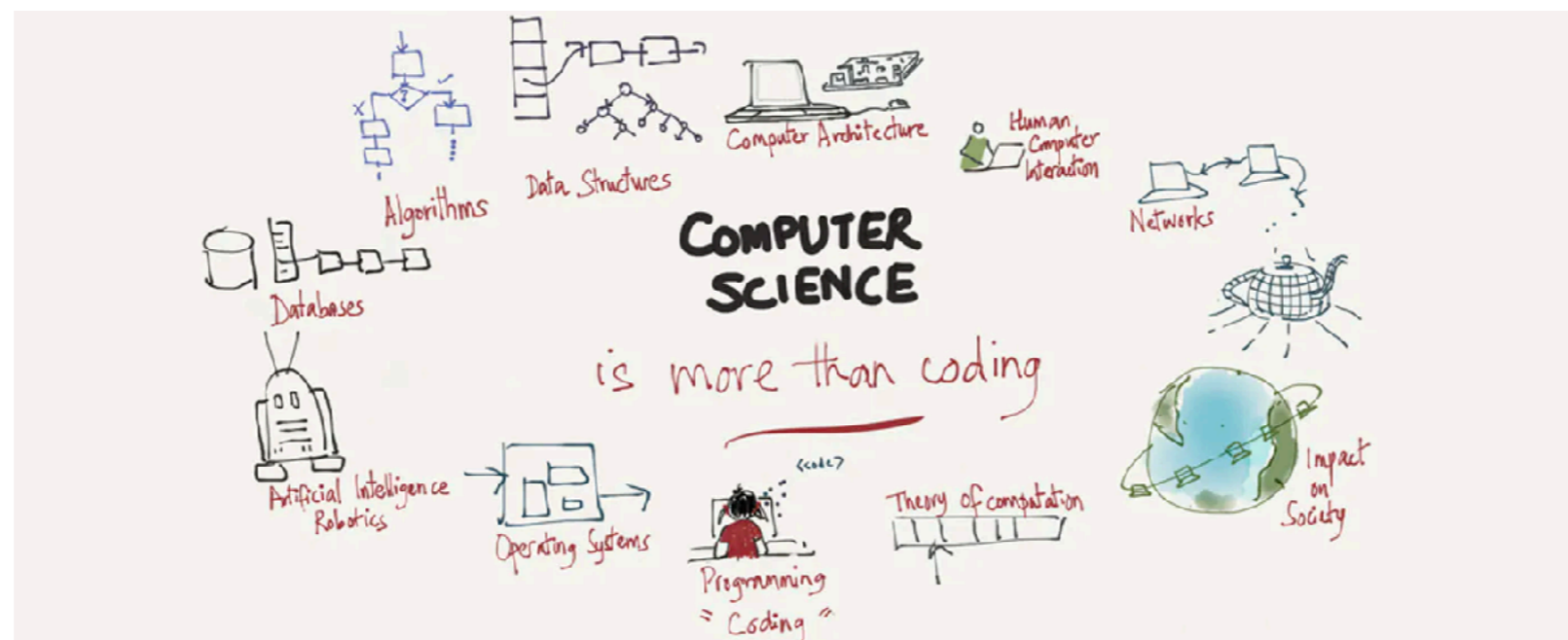
“[The Analytical Engine] might act upon other things besides *number*, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine...Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.”

Ada Lovelace (1843)

Notes On Sketch Of The Analytical Engine Invented By Charles Babbage.

What is Computer Science?

- Computer science \neq computer programming!
- Computer science is the study of what computers [can] do; programming is the practice of making computers do useful things
- Programming is a big part of computer science, but there is much more to CS than just writing programs!
- Another part of CS is **computational thinking**



Computational Thinking

- Computational thinking allows us to take a complex problem, understand what the problem is and develop possible solutions. We can then present these solutions in a way that a computer, a human, or both, can understand.
- Four pillars of CT:
 - **Decomposition** - break down a complex problem or system into smaller, more manageable parts
 - **Pattern recognition** – look for similarities among and within problems
 - **Abstraction** – focus on important information only, ignore irrelevant details
 - **Algorithms** - develop a step-by-step solution to the problem, or the rules to follow to solve the problem
- A computer can perform billion of operations per second, but computers only do exactly what you tell them to do!
- In this course we will learn how to 1) use CT to develop algorithms for solving problems, and 2) implement our algorithms through computer programs

Course Logistics

CS 134 Team



Rohit Bhattacharya
rohitb@cs.williams.edu
He/Him/His
TBL 309B



Jeannie Albrecht
jeannie@cs.williams.edu
She/Her/Hers
TCL 305



Steve Freund
freund@cs.williams.edu
He/Him/His
TPL 302



Lida Doret
lida@cs.williams.edu
She/Her/Hers
TCL 205

Staff email: cs134_staff@cs.williams.edu

CS 134 TA Team

Sarah Fida

Nathan Thimothe

Aaron Schroen

Molly Arts

Isabel Grondin

Mira Sneirson

Sammy Sasaki

Yufeng Wu

Sabiha Imran

Milo Chang

Justin Byun

Gaurnett Flowers

David Wang

Enrollments and Waitlists

- If you are not actually enrolled, we cannot guarantee that you'll get a slot (sorry!)
- All lectures and labs are over-enrolled
- If you are enrolled but not seriously intending to take the class, please be kind, drop the course now to make room for others

CS 134: Course Website

- <https://www.cs.williams.edu/~cs134/>
- One stop shop for: syllabus, Office & TA Hours, book chapters, lecture slides, labs, etc!



CSCI 134: Introduction to Computer Science

Search this site...

Calendar

Syllabus

Office and TA Hours

Resources

Computer Science Department

Williams College

CSCI 134: Introduction to Computer Science

We are surrounded by information. This course introduces fundamental computational concepts for representing and manipulating data. Using the programming language Python, this course explores effective ways to organize and transform information in order to solve problems. Students will learn to design algorithms to search, sort, and manipulate data in application areas like text and image processing, scientific computing, and databases. Programming topics covered include procedural, object-oriented, and functional programming, control structures, structural self-reference, arrays, lists, streams, dictionaries, and data abstraction. This course is appropriate for all students who want to create software and learn computational techniques for manipulating and analyzing data.

Spring 2022 Enrollments

Unfortunately our course is often over-enrolled for Spring 2022. If you were dropped, or if you did not pre-register but are interested in taking CSCI 134 in the future, [please add your name to the waitlist](#). We will randomly select students from the waitlist if slots become available. Students on the waitlist will also be given priority enrollment in future semesters.

Remote Attendance

For the first week of the semester, please attend lectures and labs via this [zoom link](#). You must be on campus or logged into the college's [VPN](#) to access that link.

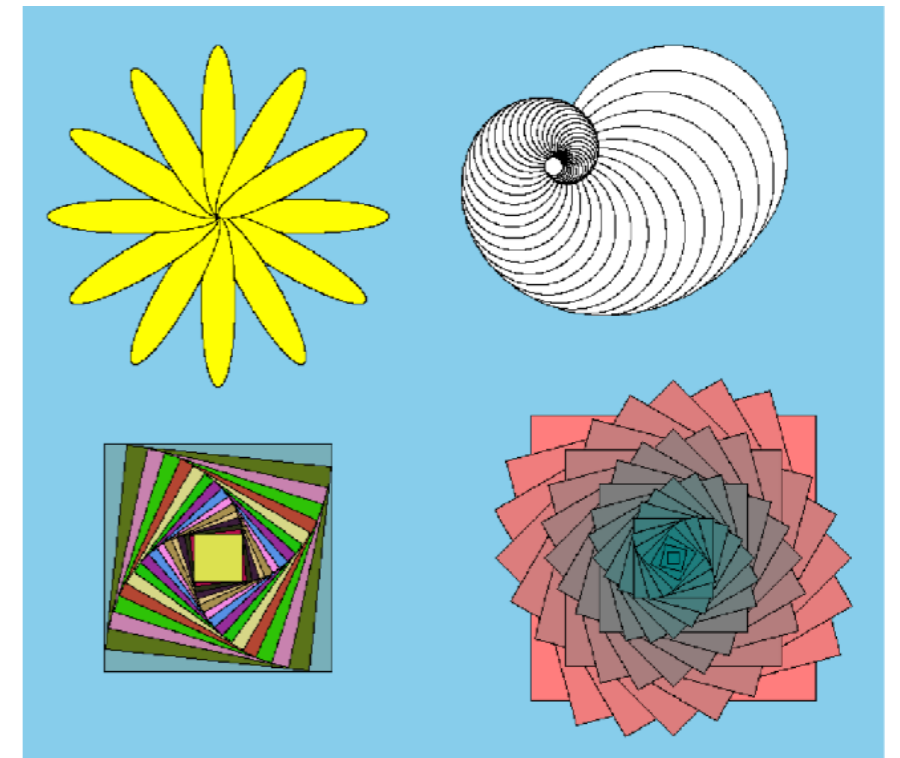
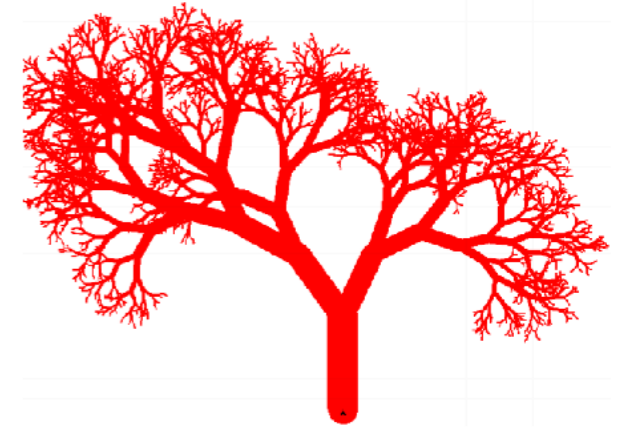
Calendar

↓ Scroll to Today

MON	TUE	WED	THU	FRI
01/31	02/01	02/02	02/03	02/04 Welcome & Logistics
02/07 Types & Expressions	02/08	02/09 Functions	02/10	02/11 Booleans & Conditionals

Topics/Concepts

- Abstraction and modularity
- Representing knowledge with data structures
- Iteration and recursion as computational tools
- Divide and conquer problem solving strategies
- Iterative and incremental programming
- Testing and debugging
- Organizing and dealing with data
- Plotting and visualizing data
- Playing with python graphics
- Object oriented programming
- Transitioning from python to Java



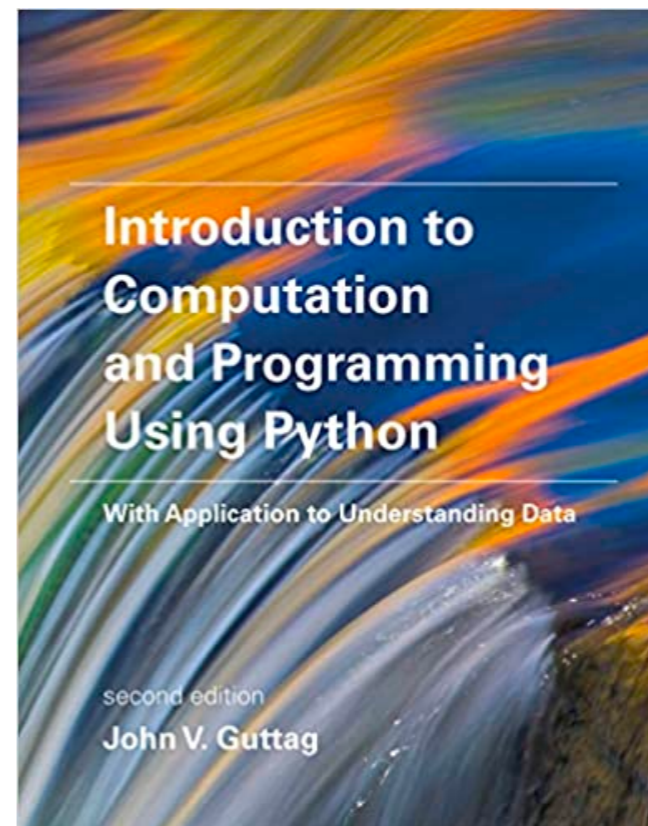
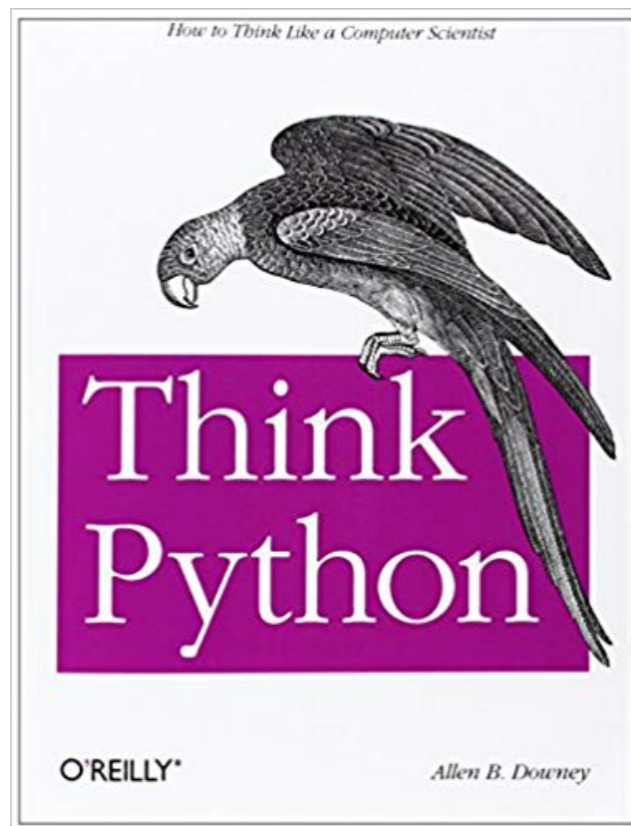
Schedule

- Full schedule available on the course webpage

MON	TUE	WED	THU	FRI
01/31	02/01	02/02	02/03	02/04 Welcome & Logistics
02/07 Types & Expressions	02/08	02/09 Functions	02/10	02/11 Booleans & Conditionals
02/14 Conditionals & Modules	02/15	02/16 Sequences & Loops	02/17 Winter Carnival	02/18 Winter Carnival
02/21 Lists & Loops	02/22	02/23 Lists & Loops (2)	02/24	02/25 Files & Comprehensions
02/28 Nested Lists & List Methods	03/01	03/02 Lists & Mutability	03/03	03/04 Aliasing & While Loops
03/07 Tuples & Sorting	03/08	03/09 Sorting, Sets & Dictionaries	03/10	03/11 Dictionaries & Plotting
03/14 Dictionaries & Sets	03/15	03/16 Recursion	03/17 Evening Midterm!	03/18 No class

Textbook(s)

- **Think Python:** How to think like a computer scientist by Downey:
<http://greenteapress.com/wp/think-python-2e/>
- Introduction to Computation and Programming Using Python, (2nd Ed) by Guttag
- Recommended, but not required
- Chapters are assigned on the course schedule



Grading Breakdown

- **Homeworks (10%)**
 - Short answer programming & problem solving questions
 - Due on Mondays (usually on GLOW)
 - Practice using “pencil and paper” before submitting answers
- **Labs (40%)**
 - Meet Mon/Tues for 75 min
 - Monday labs → Wed @ 11pm | Tuesday labs → Thurs @ 11 pm
 - Review lab as soon as it comes out (~Fridays)
 - No herds, but work on lab for ~1 hour before lab meeting time
- **Midterm (25%)**
 - Evening exam on **Thur, Mar 17 (plan accordingly)**
- **Final (25%)**
 - Scheduled Final Exam

Homework

- Homework 1 is out; linked on the course webpage
- It is a simple google form to get some information about you
- Due Monday Feb 7 by 11pm
- Future homework assignments will be available on GLOW

Labs

- We will release lab assignments typically on Friday
- You can find the upcoming lab assignments on the webpage
- We expect you to read over the lab and start before your scheduled lab meeting on Mon or Tue
- Labs are short: only 1.25 hours! Make the best of it by coming prepared!

Accounts

- **CS accounts**

- You should have received an email from Lida about your CS account. This is a **separate** account from your campus account!
- You will use these accounts for submitting labs this semester

- **Labs** are in **TCL 217A and TCL 216** (behind the stairwell)

- This door is also always locked!
- The combination is **3-9-2-7-8-1** (think 3-9-27-81)
- Each of you have also been assigned a unique anonymous ID assigned
 - Allows us to implement anonymous grading
 - Your email from Lida contains this info
 - Do not share your ID

Weekly Workload Summary

MON	TUE	WED	THU	FRI	SAT	SUN
Lab	Lab	Next HW posted		Next Lab posted		
		Graded Lab returned			Work on lab	
HW due 11 pm		Mon Labs due 11 pm	Tues Labs due 11 pm		Work on HW	
					Review Lectures	

Honor Code

- Resources to consult when completing assignments:
 - Textbook and lecture notes
 - Resources listed on the course website, Python3 documentation
 - TAs and Instructors
- Honor Code guidelines
 - **“Any work that is not your own is considered a violation of the Honor Code.”**
 - “Help locating errors and interpreting error messages are allowed, but a student may only receive help in correcting errors of syntax; help in correcting errors of logic is strictly forbidden.”
 - “In general, if you are taking photos of someone else’s screen, looking at someone else’s screen, or telling someone else what to type, it is likely your/their work is no longer the work of an individual student.”
 - PLEASE DO NOT CHEAT!
 - If you aren’t sure if something is considered a violation, just ask (beforehand)!

Public Health Reminders

- Masks that fully cover your nose and mouth are required at all times if you are indoors...this includes classrooms and labs
- No eating or drinking in the lab
- Use hand sanitizer as you enter and exit the lab
- Do not spray the keyboards!
- If you feel sick, do not come to class or lab!
 - Contact your instructors (cs134_staff@cs.williams.edu) and we'll figure out a plan

About Class Participation

- **We like interaction in our classes!**
- Many ways to participate:
 - Ask questions! (there are no bad questions!)
 - Answer questions (there are no wrong answers!)
 - Laugh at our jokes... (no guarantees here)
 - Talk to us after class/come to office hours
- Class participation does not mean dominating classroom discussions or interrupting your peers

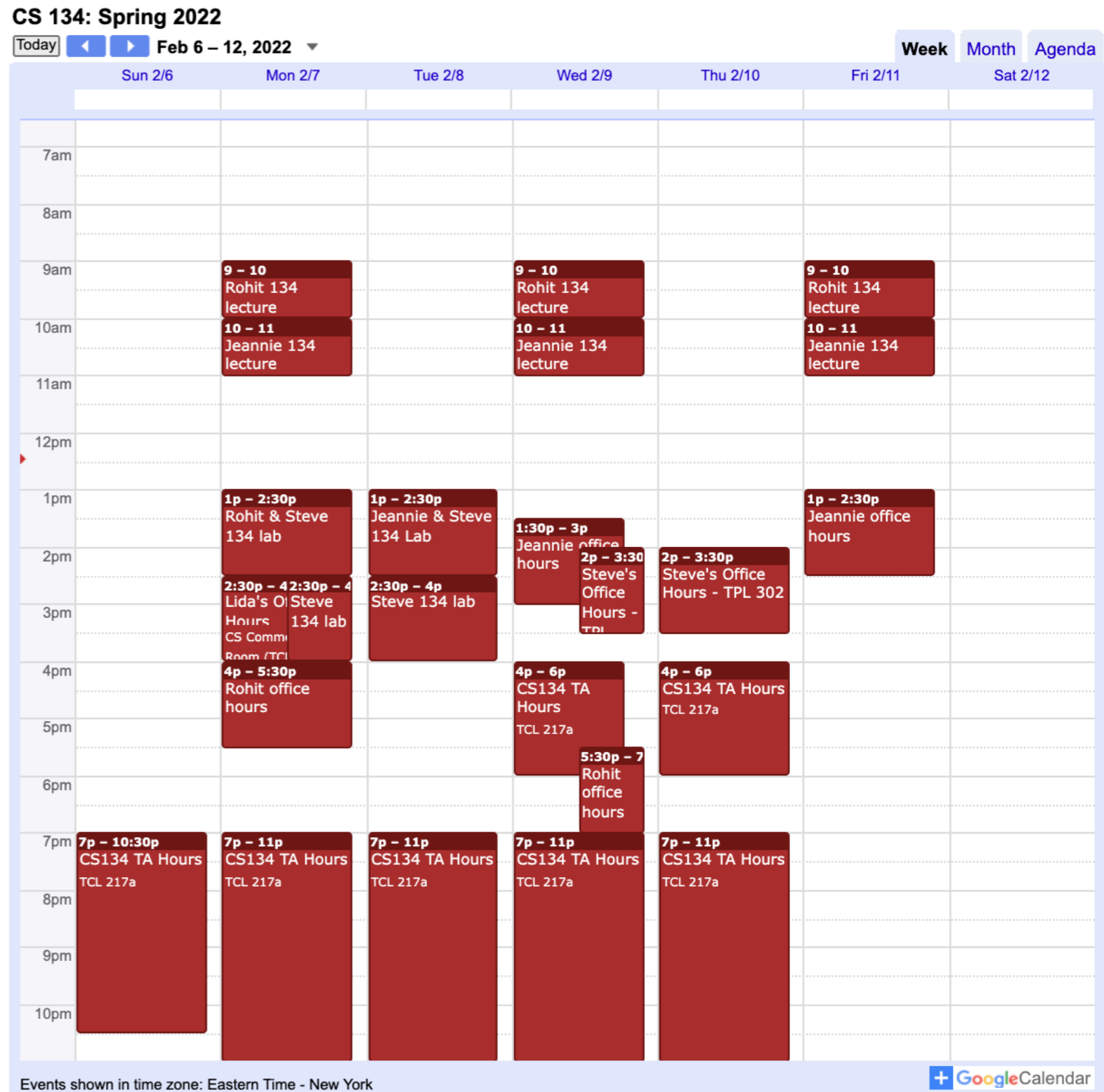
Bottom line. *Help create a vibrant, positive and inclusive classroom environment!*

Fast Paced Course

- How to succeed:
 - Read/skim recommended textbook chapters before class
 - Read and think about labs as soon as they are released
 - Seek help! Use resources! We are all here to help you!
- Learning to program is all about **PRACTICE, PRACTICE, PRACTICE!**
 - Just like learning a musical instrument, learning to ski, or building muscle, it requires repetition and dedication
 - Can't passively absorb material
 - **Don't be afraid to fail and make mistakes**—in fact you are encouraged to do so!
 - No one learns anything without making mistakes and learning why and how to fix them

Course Support: Office Hours

- Office hours (chance to discuss labs, homework, weather, sports, etc! We like to chat!)
- See calendar on course webpage
- Will be held in instructors' offices or CS common room



Course Support: TA Hours

- ~13 teaching assistants (TAs) assigned to this course
- Will be held in CS labs (TCL 216 and 217A)

MON	TUE	WED	THU	FRI	SAT	SUN
7-10:30 pm	7-11 pm	4-6 pm 7-11 pm	4-6 pm 7-11 pm			

Intro Programming and Beginner Skiing

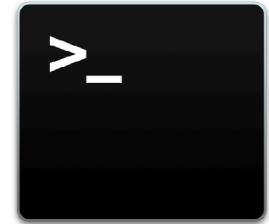
- Everyone starts at a different place with varying level of familiarities with the same or related activities
 - Maybe you have ice-skated or programmed a little Java in high school
- Different people learn at different rates
 - Some people just have really good balance!
- Do not be intimidated by classmates who seem to know more than you!
- Learning any new skills takes time and practice
- **We are here to help!**
- You'll all end up at the same place



CS | 34 Tools

- **Terminal**

- Command line or “Shell”



- **Editor**

- Atom



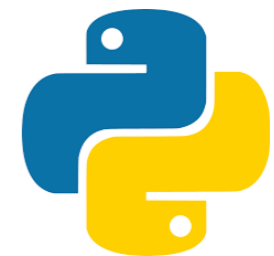
- **Git**

- “Version-control system”
- Submit work for grading (more info in lab)



- **Python**

- Programming language
- Created by Guido van Rossum in the late 1980s



- **Jupyter**

- A web-based interface for Python
- Teaching aid in class—makes teaching programming more interactive



Setting up your Computer

- In lab, you'll use our pre-configured computers (Mac)
- For Lab 1, you'll have to use your own computer (please borrow one from the library if necessary)
- Instructions on the course webpage (under Resources)
- Try following these steps on your own
- We'll help you through these steps during Lab 1



CSCI 134: Introduction to Computer Science

🔍 Search this site...

Calendar

Syllabus

Office and TA Hours

Resources

Programmer's Toolbox

[Set Up Your Mac for CSCI 134](#)

Set Up Windows for CSCI 134

How To Use Jupyter Notebooks

Think Python textbook [↗](#)

Python Style Guide

Computer Science Department [↗](#)

Williams College [↗](#)



Set Up Your Mac for CSCI 134

Throughout the semester, you should prioritize using the computers (in TC) for assignments. All the utilities you need have already been installed on them so you will not face any platform-dependent issues. However, you may also work on your own computer, if you do so during the first week.

This document gives you instructions on how to set up your personal Mac.

If you face any problems, please reach out to the [CSCI 134 staff](#) **before** you start the course.

i Note

Each person's computer is different, and while these instructions will cover most cases, there will be some unique issues and quirks that you may experience and learn about. If you have any such issues, please drop us a note.

Step 0. Verify macOS Version

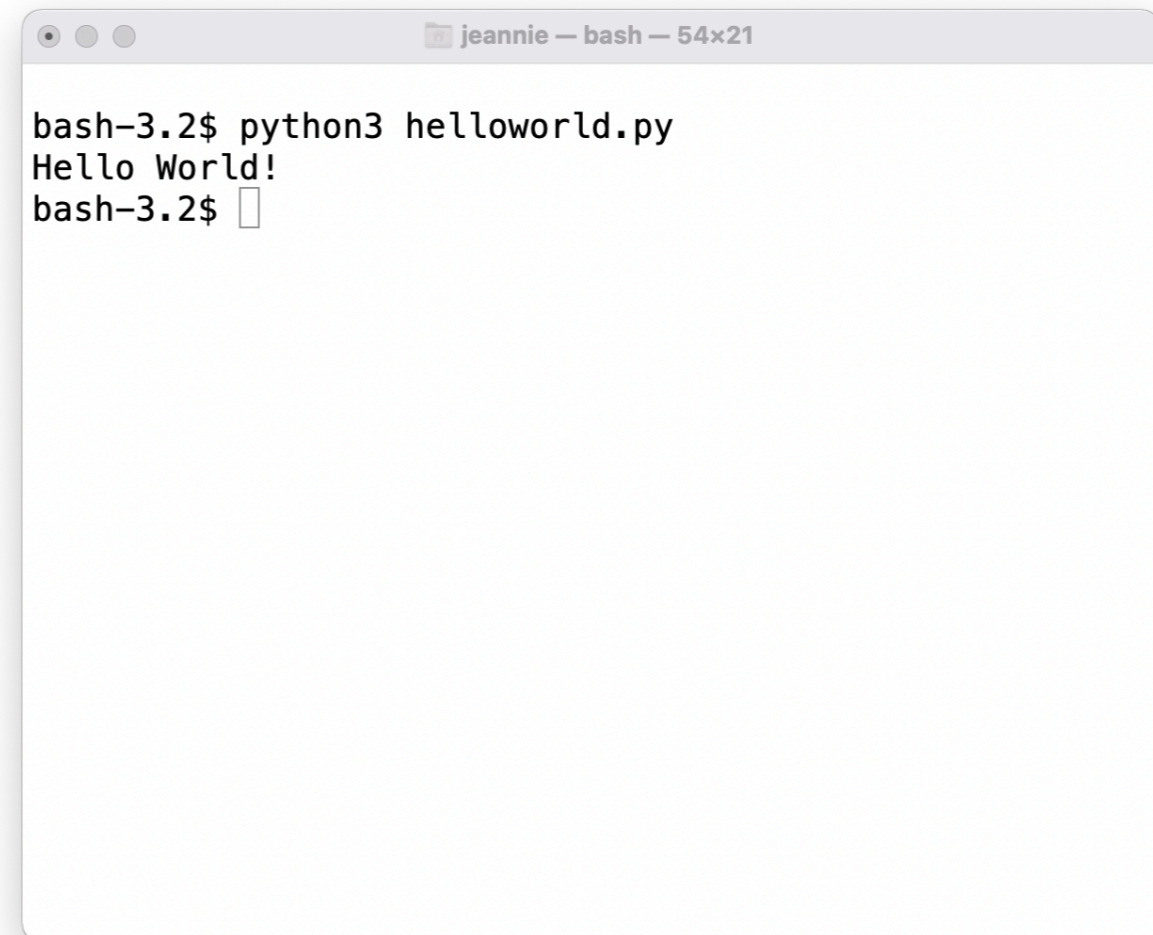
- You must have macOS 10.15 or 11 installed. If you have an older version, you will need to upgrade. To find your macOS version, click on the "Apple" icon on the top left of your screen.
- If at any point in the setup process below you are prompted to install additional tools, these tools are essential to make this setup work.

Hello World!

- Our first program:
 - Create a file called `helloworld.py` in Atom
- Execute a python3 program from Terminal
 - Type `python3 helloworld.py` and enter



```
helloworld.py — ~/cs134-s22/staff/lectures/01-welcome-and-logistics  
helloworld.py  
1 print ('Hello World!')  
2  
  
helloworld.py 2:1 1 LF UTF-8 Python main Fetch GitHub
```



```
jeannie — bash — 54x21  
bash-3.2$ python3 helloworld.py  
Hello World!  
bash-3.2$
```

CS Colloquium Today

- Almost Every Friday
- Time: **2:35pm**
- Normal Location: **TCL123** (Wege Auditorium)
- Today: Faculty research overview (on Zoom today)
 - Meet the CS faculty
 - Learn more about our research