

CSC 110
Intro to CS through programming
Section 03
Syllabus
Fall 2022

Class Information

These sections contain information about how the class is run

Class and Office Hours Schedule

Professor: Pablo Frank Bolton (pfrank at smith)

Research interests: *Human-Robot Interaction, Robotic Perception, STEM education*

Class Schedule:

- Monday (Sec 03) 3:05 PM – 4:20 PM at Ford 342
- Wednesday, Friday (Sec 03) 2:45 PM – 4:00 PM at Ford 342

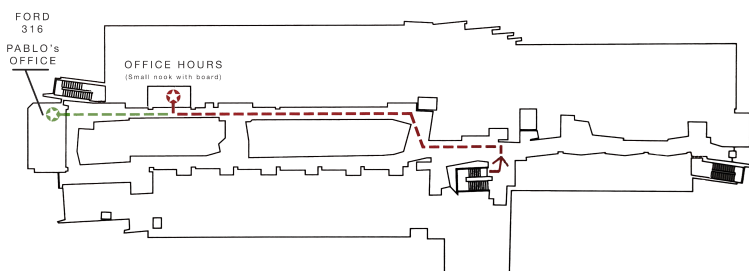
Classes will be recorded and uploaded but Attendance is mandatory.

Contact for Class stuff: Use *Slack* (fastest), office hours (most detailed), or can also email me (slowest).

Office Hours:

- Wednesdays, 4:00 PM to 5:00 PM Ford 316 or the Study Nook (both shown below)
- Extra Help: I will help students through Slack and by appointment.
- Open Door Policy: If you find my office door open and I'm inside, feel free to ask for help

FORD HALL LEVEL THREE



Office Hour Rules:

The following are requirements for showing up to OHs:

- Check the class notes / Slack before to see if this was already answered
- Have your question/program ready
- Show up on time

Objectives and Structure

CSC 110 is a gentle introduction to designing programs (recipes) for systematically solving problems. Students will learn to build programs including designing, coding, debugging, testing and documenting them. An introduction to block-structured procedural control flow including branching, iteration, and functions, using primitive and simple data types (lists). Students will understand the high level internal operation of computer systems (inputs, outputs, processing, and storage) and their application. Students will be exposed to the social and historical aspects of computing. This course is recommended for those who have no prior experience in computer science at the high school, AP, or college level.

Objectives - In completing this class, students will...

- Create and document computer programs using correct Python syntax that can be readily understood and used by other programmers.
- Propose algorithms in order to analyze problems that use basic control flow constructs (e.g., if-then statements, loops, functions, lists, simple input-output).
- Demonstrate foundational development techniques, including top-down design, program documentation, modular design, and library usage.
- Understand the high-level internal operation of a computer, including the central processing unit, simple memory management.

Course Topics: In CSC110 we will cover the following topics, but not necessarily in this order):

- Components of a computer at an abstract level (i.e., input, output, storage, computation) and physical level (i.e., hard drive, CPU, keyboard, etc.)
- Functions
- Conditionals, Math Operators, Logical Operators
- Loops and Iteration
- Variables and Data Types
- Strings
- Lists
- Dictionaries
- Basic Graphics
- Reading and Writing to Files Storage
- User inputs and print formatting
- Intro to the concept of an algorithm (e.g, as a recipe)

In addition to these core programming concepts, we will also work on some soft skills:

- Programing problem solving
- File systems (e.g., files, storage as bytes) and the cloud
- Pair programming
- Doc strings and commenting code
- Testing code (esp. thinking of edge cases)
- Debugging and tracing code
- Designing a program (e.g., software development methodologies)
- Impacts and applications of Computing (e.g., social, ethical, historical)
- Pseudocode 101

Prerequisites and Student Responsibilities

This course is recommended for those who have no prior experience in computer science at the high school, AP, or college level. Not open to students who have taken CSC 111. May not be taken concurrently with CSC 120.

Prerequisites:

This course does not have any prerequisites. In this class, we will not assume that you have prior computer science or programming experience. If initially you find the pace of class too slow and homework too easy, then you are invited to help your peers in the lab, until everyone has caught up to your prior experience. To this end, everyone is expected to respect and honor the unique perspectives each participant brings to this course and work to help one another.

Responsibilities:

- Attendance: You should attend all classes unless you have a valid excuse. I will use Moodle Quizzes and Google forms to take attendance at least once a week. Unexcused missed classes mean a loss of points in “Attendance and Participation”.

Course Materials

- **Moodle:** Course full name “CSC110-03: Introduction to Computer Science”
- **Repl.it:** Repl.it is a cloud based platform that allows the creation, assigning, completion, testing, and grading (manual or automatic) of programming assignments. You need to register for it (you’ll receive an invitation from the instructor) so you can complete readings, exercises, and assignments.
- **Slack:** You’ll receive an invitation to the workspace: *CSC-110-03-202301*

Books:

There is no required textbook for the course, and you do not need to buy any textbook to be successful.

Below are textbooks that you may want to review as a resource and some students have found helpful in the past: % few recommended books that some students may find helpful:

1. Allen B. Downey. Think Python; 2nd edition (2015). ISBN: 978-1491939369 [Note: eBook available for **FREE**](<http://greenteapress.com/thinkpython2/thinkpython2.pdf>).
2. John M. Zelle. [Python Programming: An Introduction to Computer Science](<https://www.amazon.com/Python-Programming-Introduction-Computer-Science/dp/1590282752>). Franklin, Beedle & Associates; 3rd edition (August 8, 2016). ISBN: 978-1590282755 *Note: This is the newer edition, but the 2nd edition is also fine as a reference.

The Downey and Zelle books have been put on course reserve at the Library.

If you need help covering the cost of textbooks or other academic supplies (for this or any of your courses!) please fill out the Academic Funding Application found at socialnetwork.smith.edu/forms (you must login before the forms will appear).

Grading

The final grade for this course is S/U (Satisfactory/Unsatisfactory); However, the way we will obtain this grade will be by running a “Normal” grading scheme and converting grades the following way:

- 0% to 59.9% : Unsatisfactory
- 60% to 100% : Satisfactory

Grade Calculation:

(The following grade calculation may be modified slightly depending on how the semester progresses)

- Homework Assignments: **30%**
- In-class Exercises: **10%**
- Attendance and Participation **10%**
- Partial Tests: **30%**
- Final Project: **20%**

Notes: Completing the in-class exercises will be used as an attendance metric; Participation will be actually graded based on my memory of your active involvement in Class/Office-Hours/Slack.

Late Submissions:

We will accept no late submissions without a valid excuse. The homework and projects are designed to have “checkpoints” so that you MUST show your progress at every point so starting late is actively discouraged.

Also, for Office Hours, you must show progress up to the point where you got stuck.

Accommodations:

As individuals, we learn in different ways. I try to vary the activities used during the course to suit a variety of learning patterns, and I am always open for suggestions. Please come talk to me if you have an idea that will make the course more accessible to you and/or other students. If you need special accommodation, like extended exam time, please submit requests for accommodations in writing with proof of College support from the Office of Disabilities Service within the first two weeks of class. Let me know if you need help with this process.

Academic Honesty

Team assignments require collaboration amidst each team, but no collaboration between teams is permitted. If you did not work in a team then you are not allowed to collaborate on the homework assignments. We use software to compare submissions, so please don't risk it. If you're having significant trouble with an assignment, please contact me.

Please check the [Student Handbook](#) to see the rules for Academic Integrity.

Just as you can do a google search for code online, it is trivial for us to do the same. If you feel pressured about an assignment, please come see me instead of cheating.

Academic and Mental Health Resources

The following are resources available to you that may provide assistance and support during the semester. They provide help for learning, mental health, and wellness.

Learning resources:

- Spinelli Center for Quantitative Learning
- [Jacobson Center for Writing, Teaching, and Learning](#)
- [Teaching, Learning and Research Librarians](#)

Mental Health and Wellness resources:

- [Crisis Resources](#)
- [Counseling Services](#)
- [Wellness](#)

Additional support resources:

- [Gender Identity and Expression Resources](#)
- [Report sexual misconduct and other forms of discrimination](#)

Anonymous Feedback Form

We will add a link inside Moodle to an anonymous feedback form so you can let us know if there is anything getting in the way of your learning.

Acknowledgement Some of the materials used in this course and this syllabus are derived from previous offerings of this and other courses at Smith College, as well as similar courses taught at other institutions. Appropriate references will be included on all such material.