

Introduction To Computer Science And Programming In Python

[Menu](#)
[More Info](#)

Syllabus

Course Meeting Times

Lectures: 2 sessions / week, 1 hour / session

Recitations: 1 sessions / week, 1 hour / session

Course Information

6.0001 *Introduction to Computer Science and Programming in Python* is intended for students with little or no programming experience. It aims to provide students with an understanding of the role computation can play in solving problems and to help students, regardless of their major, feel justifiably confident of their ability to write small programs that allow them to accomplish useful goals. The class will use the Python 3.5 programming language.

This is a half-semester course. Students who successfully complete 6.0001 may continue into 6.0002 *Introduction to Computational Thinking and Data Science*, which is taught in the second half of the semester.

Goals

- Provide an understanding of the role computation can play in solving problems.
- Help students, including those who do not plan to major in Computer Science and Electrical Engineering, feel confident of their ability to write small programs that allow them to accomplish useful goals.
- Position students so that they can compete for research projects and excel in subjects with programming components.

Textbook

The textbook is Gutttag, John. *Introduction to Computation and Programming Using Python: With Application to Understanding Data Second Edition*. MIT Press, 2016. ISBN: 9780262529624. The book and the course lectures parallel each other, though there is more detail in the book about some topics. It is available both in hard copy and as an e-book.

Lecture and Recitation Attendance

A significant portion of the material for this course will presented only in lecture, so students are expected to regularly attend lectures.

Recitations give students a chance to ask questions about the lecture material or the problem set for the given week. Sometimes, new material may be covered in recitation. Recitation attendance is encouraged but not required.

Problem Sets and Quizzes

Each problem set will involve programming in Python. There will be 6 problem sets in the course. There will be two quizzes. All quizzes will be closed-book, though you will be allowed to bring one page of notes to the first quiz and two pages of notes to the second quiz. Pages must be letter-sized, double-sided, either handwritten or typed.

Grading Policy

Grades will be roughly computed as follows:

ACTIVITIES	PERCENTAGES
Problem sets	30%
Completion of mandatory finger exercises	10%
Midterm Quiz	20%
Final Quiz	40%

Problem sets will be graded out of 10 points. Submissions that do not run will receive at most 20% of the points. Please contact your Teaching Assistant if you have a problem understanding your problem set grade.

Note: Quizzes and finger exercises are not available on OpenCourseWare.

Extension and Dropping Problem Sets Policy

We do not grant any extensions. Instead, we offer late days and the option of rolling at most 2 problem set grades into the final quiz score.

Late Days

At the beginning of the term, students are given two late days that they can use on problem sets. Starting with Problem Set 1, additional late days can be accumulated for each assignment, one late day for each day the assignment is turned in ahead of the deadline. Up to three late days may be accumulated in this fashion in this course, i.e you can only have a maximum of 3 late days at any point in time. Late days are discrete (a student cannot use half a late

day). The staff will keep track of late days and feedback for each problem set will include the number of late days the student has remaining. Any additional late work beyond these late days will not be accepted. To avoid surprises, we suggest that after you submit your problem set, you double check to make sure the submission was uploaded correctly.

Rolling Over Problem Sets

Before the final quiz, we will send out an announcement in which you can choose at most 2 problem sets that you can drop. If dropped, the percent that the problem sets are worth will be rolled into the final quiz score. We strongly urge you to see the late days and dropping the problem sets as backup in case of an emergency. Your best strategy is to do the problem sets early before work starts to pile up.

Calendar

SES #	TOPICS	ASSIGNMENTS
1	What is computation?	Pset 0 released
2	Branching and Iteration	Pset 1 released
3	String Manipulation, Guess and Check, Approximations, Bisection	Pset 0 due
4	Decomposition, Abstractions, Functions	Pset 2 released
5	Tuples, Lists, Aliasing, Mutability, Cloning	Pset 1 due
6	Recursion, Dictionaries	Pset 3 released
7	Testing, Debugging, Exceptions, Assertions	Pset 2 due; Quiz 1
8	Object Oriented Programming	
9	Python Classes and Inheritance	Pset 3 due; Pset 4 released
10	Understanding Program Efficiency, Part 1	Pset 4 due; Pset 5 released
11	Understanding Program Efficiency, Part 2	
12	Searching and Sorting	Pset 5 due; Final Quiz



Over 2,500 courses & materials

Freely sharing knowledge with learners and educators around the world. [Learn more](#)

[Accessibility](#)

[Creative Commons License](#)

[Terms and Conditions](#)

Proud member of:  Open Education GLOBAL



© 2001–2024 Massachusetts Institute of Technology