Basic Course Information

You are not logged in.

If you are a current student, please Log In for full access to the web site.
Note that this link will take you to an external site (`https://shimmer.mit.edu`) to authenticate, and then you will be redirected back to this page.

Table of Contents

- 1) Overview
  - 1.1) Learning Objectives
  - 1.2) Description, Prerequisites, and Major Requirements
- 2) Required Materials
- 3) Structure and Schedule

1) Overview

Welcome to 6.009! We're really looking forward to working with all of you, and we hope that you're excited as well. Together, we're about to embark on the next leg of the journey of learning computer programming, one of the most interesting, fun, and practically useful skills that anyone can possess in the modern world.

We typically have students with a variety of different goals, and we think that 6.009 has something to offer everyone. Some of you may be interested in potential future careers in software engineering. Others might be interested in computation not for its own sake but for its usefulness as a general tool for solving problems in other domains. Still others might not plan on using programming in any professional capacity but rather just find it fun and interesting as a hobby. We expect that we have students from all of these categories (and more), and we have designed 6.009 with all of you in mind, focusing on skills and competencies that are broadly useful to all who use computation.

6.009 is the second programming subject in a series, and in some sense, our focus is on adjusting our programming practices as we move from relatively small programs, run in an artificial environment; to more complex programs designed to be run on your computer, working with real data and producing meaningful results. As our programs grow in terms of scope and scale, we'll encounter new concerns that weren't as important when working with smaller programs (or maybe didn't even exist at that scale), and so we'll need to adjust our approach. And we'll focus on three separate but related parts of the programming process:

- **programming:** analyzing and decomposing problems, developing plans
- **coding:** translating those plans into correct, efficient, idiomatic, understandable Python code
- **debugging:** developing test cases, verifying correctness, and finding and fixing the errors that inevitably happen

In order to facilitate growth in these areas, we'll spend time on a number of different topics in 6.009, including:

- high-level design strategies,
- ways of managing complexity,
- details and advanced features and "goodies" of Python,
- a mental model of Python's operation,
- testing and debugging strategies,
- and more!

Of course, just talking about these things is not enough. Deliberate practice is key to developing programming, as it is with any skill (think learning a musical instrument, a sport, or a second language). To improve as a programmer, it helps to:

- watch how experienced programmers approach problems,

- write programs of your own, and
- receive feedback on your work from more experienced programmers.

And the good news is that, by design, 6.009 will provide you with a *lot* of opportunities to do all of these things.

It's worth mentioning, too, that while learning a new skill is generally very rewarding, not every part of the process will be particularly enjoyable. Learning a new skill involves a lot of time, dedication, patience, hard work, and struggle. And so it is with programming. It is natural to encounter difficulties along the way; and sometimes those difficulties can cause us to feel unsure, vulnerable, intimidated, insecure, and not very bright. Those feelings are perfectly natural (even for very experienced programmers), and they do not mean that you cannot ultimately accomplish your goals (whatever they may be) as they relate to programming; they just mean that there is more work to be done. So along the way, if and when those feelings creep in, take a deep breath and remember that those kinds of obstacles are normal, that all experienced programmers have been there, and there are lots of people here to support you along the way.

We're excited to work alongside you as you take the next steps of this journey, and we hope that you're excited to get underway as well!

1.1) Learning Objectives

At some level, 6.009 is about creating computer programs and using computation to solve interesting problems. And throughout 6.009, you will improve as a programmer. But we also hope to explore deeper conceptual questions, so that you'll not only be able to make the computer work for you, but you will be able to reason about how to structure your programs so that they are correct, efficient, and understandable.

We have several goals for your learning in 6.009:

- **Programming Objectives**

  After taking 6.009, you should be able to...

  - ...design and implement small- and medium-sized Python programs in idiomatic style, including some advanced features of the Python language.
  - ...implement, test, and debug Python programs in your preferred programming environment.
  - ...use programming as a tool to solve problems in other domains.
  - ...use the command line to interact with your computer.

- **Conceptual Objectives**

  After taking 6.009, you should be able to...

  - ...deconstruct a large problem into smaller pieces that can be reasoned about independently.
  - ...reason about the execution of small- and medium-sized Python programs without using a computer, through the use of environment diagrams and other tools.
  - ...understand the tradeoffs associated with various algorithms and data structures, in terms of efficiency and correctness.
  - ...predict some edge cases and failure modes when designing a program, before writing any code.

1.2) Description, Prerequisites, and Major Requirements

- **Description**

  From the subject listing and schedule:

Introduces fundamental concepts of programming. Designed to develop skills in applying basic methods from programming languages to abstract problems. Topics include programming and Python basics, computational concepts, software engineering, algorithmic techniques, data types, and recursion. Lab component consists of software design and implementation.

- **Prerequisites**

  An understanding of the basics of programming and the basics of the Python language are necessary prerequisites to this course. 6.0001 (or 6.0001 Advanced Standing Exam), 6.0002, 6.01, or 6.145 (formerly known as 6.s080) can serve as the prerequisite.

  This prerequisite is **strictly enforced**. If you have not met the prerequisite for this course, you will not be able to submit assignments in 6.009.

- **Course 6 and Institute Requirements**

  6.009 is a prerequisite for 6.031 and a co-requisite for 6.006.

  6.009 is required for 6-3, 6-7, and 11-6 majors. It can be used as a foundation class for 6-2 majors and as an option to satisfy one of the 6-14 degree requirements and one of the requirements of the Minor in Computer Science.

  6.009 satisfies the Institute Laboratory requirement.

2) Required Materials

There is **no required textbook** for this class. Everything you need will be made available through this web site throughout the semester, through lecture and recitation materials, assignment writeups, and course notes.

If you are looking for additional materials, the following are books/resources that others have found useful in the past:

- Python:
  - The Python Tutorial
  - 6.145 (notes and exercises)
  - *Think Python* by Alan Downey
- Debugging:
  - *Debug It!*, Paul Butcher, The Pragmatic Bookshelf, 2009
  - *Debugging*, David J. Agans AMACOM, 2002
  - *Why Programs Fail* (2nd edition), Andreas Zeller, Morgan Kaufmann, 2009

3) Structure and Schedule

Each week will consist of one lecture, one recitation, one large programming assignment ("lab"), and several opportunities to get help during open lab hours.

- **Lectures**

  Lectures will be held on Mondays, 11-12:30pm in 26-100. Attendance and participation are expected, so please bring a computer with you. Most lectures will be preceded by a small set of readings and/or videos and a couple of small exercises, which you will be expected to complete before the lecture.

- **Recitations**

  On Wednesdays, we will conduct one-hour-long recitations on various topics including testing, debugging, and basic programming concepts. Attendance and participation are expected, so please bring a computer with you. We will assign sections prior to our second recitation.

We will offer the following sections:

- Section 0: 9am-10am in 32-124 with Adam
- Section 1: 10am-11am in 32-124 with Adam
- Section 2a: 11am-12pm in 26-100 with Saman
- Section 2b: 11am-12pm in 5-217 with Jonathan
- Section 3: 12pm-1pm in 26-100 with Saman
- Section 4a: 1pm-2pm in 1-390 with Charles
- Section 4b: 1pm-2pm in 4-270 with Mike
- Section 5: 2pm-3pm in 1-390 with Jonathan
- Section 6: 3pm-4pm in 4-153 with Charles
- Section 7: 4pm-5pm in 36-156 with Mike

- **Programming Assignments ("Labs")**

  A new lab will release each Friday morning, and the bulk of the lab will come due the following Friday afternoon. In some cases, portions of the lab may come due in the middle of the week, so please pay attention to those deadlines.

  Note that labs are substantial programming assignments, and we expect that in the average week, the average student will spend around 10 hours working on the lab. As such, we strongly encourage you to start early!

  In order to receive credit for a lab, you must also complete the associated "checkoff," which is a brief conversation with a staff member about your code. At the checkoff, we will ask some questions about your code and also provide some feedback on style. The checkoff can also be a good opportunity to learn new Python tricks and alternative ways of aproaching the lab!

- **Open Lab Hours**

  Course staff will be available to answer questions, help with debugging, and complete checkoff meetings during the following times:

  - Monday-Thursday, 7pm-10pm in 34-501
  - Tuesday and Thursday, 9am-noon in 34-501
  - Friday, 9am-5pm in 34-501
  - Sunday, 10am-7pm in 34-501

  During these times, please come the the given room and use the help queue to request help (or checkoffs).