6.00.1x Syllabus

Welcome to 6.00.1x! In this course you'll be learning the basics of computer programming in Python and the fundamentals of computation, as well as getting the opportunity to implement your own Python functions.

This course is offered online and we understand that there are many opportunities available to cheat. We caution you to not do so. You will learn less and only harm yourself by cheating. We ask that you review our collaboration and forum guidelines, available on the course handouts page, to understand how we expect our students to conduct themselves in this course. Additionally, all students are expected to follow the edX Honor Code, available at https://www.edx.org/honor.

If you have a disability-related request regarding accessing an assignment or the final exam, please contact the edX Help Center https://courses.edx.org/support/contact_us as early in the course as possible or at least 2 weeks prior to exam start date to allow time to respond in advance of course deadlines. Requests are reviewed via an interactive process to meet accessibility requirements for learners with disabilities and uphold the academic integrity for MITx.

Grading Policy

In this course there will be many types of assignments. Your final grade will be a weighted average of the following:

- Finger exercises (available within each lecture video sequence) – 10%
- Problem sets – 40%
- Midterm – 25%
- Final exam – 25%

In order to earn a certificate for 6.00.1x, students must pass the course with a grade of C or better. The following grading breakdown will apply:

- >= 80%: A
- >= 65%: B
- >= 55%: C

Exercises and Exams

All course material will be released at 14:00 UTC. Finger exercises have no due date, but we encourage students to complete them as they view the lectures. See the Calendar tab for Problem Set due dates. Regrettably, **extensions are unavailable** for any assignment but your **lowest problem set score is dropped**.

All problem sets will be due at **23:30** or **11:30 pm UTC**. This is the Coordinated Universal Time, also known as the Greenwich Mean Time. Convert to your local time zone using an online converter such as this one:
http://www.timeanddate.com/worldclock/converter.html

Exams are scheduled in advance. The exams will take place online, on the course website. **Exams are timed** – once you begin during the exam period, you will have 8 hours to complete it.

- The **Midterm** will take place from July 1 (14:00 UTC) to July 5 (23:30 UTC).
- The **Final Exam** will take place from July 29 (14:00 UTC) to Aug 2 (23:30 UTC)

During the exam period, the forums will be shut down. You will still be able to read posts but you will not be able to post any questions. The honor code prohibits students from communicating with one another during the exam period in any way whatsoever – so please don't discuss the exam on any other forum, website or in person with anyone else.

List of Lecture Topics

Lecture 1 – **Introduction to Python**:

- Knowledge
- Machines
- Languages
- Types
- Variables
- Operators and Branching

Lecture 2 – **Core elements of programs**:

- Bindings
- Strings
- Input/Output
- IDEs
- Control Flow
- Iteration
- Guess and Check

Lecture 3 – **Simple Programs**:

- Approximate Solutions
- Bisection Search
- Floats and Fractions
- Newton-Raphson

Lecture 4 – **Functions**:

- Decomposition and Abstraction
- Functions and Scope
- Keyword Arguments
- Specifications
- Iteration vs Recursion
- Inductive Reasoning
- Towers of Hanoi
- Fibonacci
- Recursion on non-numerics
- Files

Lecture 5 – **Tuples and Lists**:

- Tuples
- Lists
- List Operations
- Mutation, Aliasing, Cloning

Lecture 6 – **Dictionaries**:

- Functions as Objects
- Dictionaries
- Example with a Dictionary
- Fibonacci and Dictionaries
- Global Variables

Lecture 7 – **Debugging**:

- Programming Challenges
- Classes of Tests
- Bugs
- Debugging
- Debugging Examples

Lecture 8 – **Assertions and Exceptions**

- Assertions
- Exceptions
- Exception Examples

Lecture 9 – **Classes and Inheritance**:

- Object Oriented Programming
- Class Instances
- Methods
- Classes Examples
- Why OOP
- Hierarchies
- Your Own Types

Lecture 10 – **An Extended Example**:

- Building a Class
- Viualizing the Hierarchy
- Adding another Class
- Using Inherited Methods
- Gradebook Example
- Generators

Lecture 11 – **Computational Complexity**:

- Program Efficiency
- Big Oh Notation
- Complexity Classes
- Analyzing Complexity

Lecture 12 – **Searching and Sorting Algorithms**:

- Indirection
- Linear Search
- Bisection Search
- Bogo and Bubble Sort
- Selection Sort
- Merge Sort

Lecture 13 – **Visualization of Data**:

- Visualizing Results
- Overlapping Displays
- Adding More Documentation
- Changing Data Display
- An Example

Lecture 14 – **Summary**