

# 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>

## 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%
- Quiz – 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:

- $\geq 90\%$ : A
- $\geq 75\%$ : B
- $\geq 65\%$ : C

## Exercises and Exams

This course is self-paced. All course materials are available right away.

The exams will take place online, on the course website, and will be timed.

- 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. **You are not allowed to post anything regarding exams. Any such posts will be removed.**
- Before each exam, you will sign an honor pledge.
- After you sign the pledge, you will have **8 hours to take the exam**. There will be no re-takes.

## 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-numeric
- 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
- Visualizing 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**