

# 6.00.1x Syllabus

Welcome to the first offering of 6.00.1x, Fall 2013! 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:

- $\geq 80\%$ : A
- $\geq 65\%$ : B
- $\geq 55\%$ : C

## Exercises and Exams

Finger exercises have no due date, but we encourage students to complete them as they view the lectures. See the Calendar tab for due Problem Set due dates. **Regrettably, extensions are unavailable for any assignment.**

All problem sets will be due at **10 PM EST** – this is the time zone for Boston, MA, USA. 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 **Quiz** will take place from November 9 – November 11. The final exam will take place from December 12 – December 15. The exams will take place online, on the course website.

You will be expected to sit for the exam for one continuous 12-hour block during each exam period; you may choose the block of time based on your own schedule. While the quiz is written to take only 90 minutes, and the final exam is written to take about 2 – 3 hours, we provide 12 hours of time with the understanding that some students may have slow internet connections.

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:**

- What a computer does
- Computational thinking
- Aspects of programming languages
- Basic machine architecture

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

- Kinds of languages
- Objects, expressions, operators
- Abstraction by naming
- Strings
- Scripts
- Straightline programs
- Branching programs

#### **Lecture 3 – Simple algorithms:**

- Simple programs
- Iteration
- Input
- Exhaustive enumeration
- Guess and check
- For and while loops
- Approximate solutions
- Bisection search
- Newton-Raphson

#### Lecture 4 – **Functions:**

- Function syntax
- Abstraction by specification
- Functions and scoping
- Floating point numbers
- Specifications
- Modules

#### Lecture 5 – **Recursion:**

- Recursion
- Inductive reasoning
- Divide and conquer

#### Lecture 6 – **Objects:**

- Structured types and mutability
- Tuples
- Lists and mutability
- Functions as objects
- Dictionaries

#### Lecture 7 – **Debugging:**

- Testing and debugging
- Black box testing
- Glass box testing
- Integration testing and unit testing
- Debugging with binary search

#### Lecture 8 – **Assertions and Exceptions**

- Assertions
- Exceptions

#### Lecture 9 – **Efficiency and orders of growth:**

- Complexity
- How to measure complexity
- Asymptotic notation

#### Lecture 10 – **Memory and search:**

- Memory storage
- Indirection
- Searching and sorting methods
- Hashing

Lecture 11 – **Classes:**

- Definition of classes
- Classes versus instances
- Methods
- Bindings of values
- Exceptions

Lecture 12 – **Object Oriented Programming and Inheritance:**

- Inheritance
- Object oriented programming

Lecture 13 – **Trees:**

- Decision trees
- Depth-first search
- Breadth-first search

Lecture 14 – **Summary**

Lecture 15 – **Research videos**