6.00x Syllabus

Welcome to the first offering of 6.00x, Fall 2012! 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) – 5%
- Problem sets – 15%
- Midterm exam 1 – 20%
- Midterm exam 2 – 20%
- Final exam – 40%

In order to earn a certificate for 6.00x, 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

Finger exercises have no due date, but we encourage students to complete them as they view the lectures. Problem sets are due weekly; see the Course At-A-Glance handout for due dates.

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. Midterm Exam 1 will take place from November 1 – November 4; Midterm Exam 2 will take place from December 6 – December 9; the final exam will take place from January 10 – January 14. The exams will take place online, on the course website.

You will be expected to sit for the exam for the duration of 3 hours during each exam period; you may choose the block of time based on your own schedule. While the exams are written to take only 90 minutes, we provide 3 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 website or in person with other students.

List of Lecture Topics

Lecture 1 – **Introduction**:

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

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**:

- Iteration
- 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
- Specifications
- Modules

Lecture 5 – **Recursion**:

- Recursion
- Inductive reasoning
- Divide and conquer

Lecture 6 – **Objects**:

- Structure 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 approaches

Lecture 8 – **Efficiency and orders of growth**:

- Complexity
- How to measure complexity
- Asymptotic notation
- Classes of algorithmic complexity

Lecture 9 – **Memory and search**:

- Memory storage
- Indirection
- Search and sort methods
- Hashing

Lecture 10 – **Classes**:

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

Lecture 11 – **Object Oriented Programming**:

- Inheritance
- Object oriented programming
- Specifications
- Iterators
- Debugging with classes

Lecture 12 – **Plotting**:

- Plotting techniques
- Methods for using plotting to understand programs

Lecture 13 – **Simulations and random walks**:

- Stochastic programs and probability
- Random walks
- Variations on random walks

Lecture 14 – **Sampling and Monte Carlo methods**:

- Monte Carlo simulations
- Inferential statistics

Lecture 15 – **Statistical thinking**:

- Statistics
- Histograms
- Statistical measures

Lecture 16 – **Using randomness to solve non-random problems**:

- Distributions
- Examples of Monte Carlo problems

Lecture 17 – **Curve fitting**:

- Understanding data
- Data fitting methods
- Coefficient of determination

Lecture 18 – **Optimization**:

- Optimization problems
- Knapsack problems
- Solution methods for knapsack problems

Lecture 19 – **Graphs**:

- Graph search
- Breadth first search
- Depth first search
- Shortest path problems

Lecture 20 – **Graphs**:

- Additional graph problems
- Cliques
- Min-cut methods

Lecture 21 – **Dynamic programming**:

- Dynamic programming
- Example problems

Lecture 22 – **Statistical fallacies**:

- Statistical fallacies
- Careful use of statistical methods

Lecture 23 – **Simulation example**:

- Large scale simulation example

Lecture 24 – **Summary**