CSCI E-7: Introduction to Programming with Python
Spring, 2015

General Course Information

Instructor: Dr. Andrey Sivachenko
Office:
Office Hours:
Phone: 801-450-5934
E-mail: asivachenko@fas.harvard.edu,
Web site: https://canvas.harvard.edu/courses/8251
Classroom: 53 Church St, Rm 202, Cambridge
Class Times: Wednesdays, 7:40-9:40 pm

Prerequisites: none

Textbook: *Fundamentals of Python: First Programs*
Author: Kenneth Lambert
Publisher: Course Technology, Cengage Learning, 2012
ISBN-13: 978-1-111-82270-5

Course Objectives

This course introduces core programming basics—including data types, control structures, algorithm development, and program design with functions—via the Python programming language. The course discusses the fundamental principles of Object-Oriented Programming, as well as in-depth data and information processing techniques. Students will solve problems, explore real-world software development challenges, and create practical and contemporary applications.

Specific topic coverage includes:

- Algorithms and Information Processing
- Control Structures
- Boolean logic and Numeric Data Types
- Strings, Text Files, Lists, and Dictionaries
- Procedural Abstraction in Function Definitions
- Objects and Classes
- Graphics and Image Processing
- Networks and Client/Server Programming
- Graphic User Interfaces (GUI)
- Events and Event-driven Programming

Supplementary information for the course is available at **https://canvas.harvard.edu/courses/8251**. The Web site contains class announcements and notes, test dates, PowerPoint slides, the course syllabus, and additional information.

If you have any questions about the course or need assistance, please contact me in person or by telephone or by e-mail at any time.

All class assignments should be submitted via Canvas website on or before the due date. The solutions to the programming problems *must include full working code*, not just an output of your program.

Grading and Evaluation Criteria

50% of the grade is based on a midterm and a final examination (25% each).

50% of the grade is based on homeworks. Solutions for homework problems that ask for writing a program must include full working code in order to get full credit.

Course Outline

Week	**Topics**	**Chapter Readings**
1	Conceptual introduction: topics in computer science, algorithms; modern computer systems: hardware architecture, data representation in computers, software and operating system; installing Python; basic syntax, interactive shell, editing, saving, and running a script.	Chapter 1
2	The concept of data types; variables, assignments; immutable variables; numerical types; arithmetic operators and expressions; comments in the program; understanding error messages;	Chapter 2
3	Conditions, boolean logic, logical operators; ranges; Control statements: if-else, loops (for, while); short-circuit (lazy) evaluation	Chapter 3
4	Strings and text files; manipulating files and directories, os and sys modules; text files: reading/writing text and numbers from/to a file; creating and reading a formatted file (csv or tab-separated). String manipulations: subscript operator, indexing, slicing a string; strings and number system: converting strings to numbers and vice versa. Binary, octal, hexadecimal numbers	Chapter 4
5	Lists, tuples, and dictionaries; basic list operators, replacing, inserting, removing an element; searching and sorting lists; dictionary literals, adding and removing keys, accessing and replacing values; traversing dictionaries.	Chapter 5
6	Design with functions: hiding redundancy, complexity; arguments and return values; formal vs actual arguments, named arguments. Program structure and design. Recursive functions.	Chapter 6
7	**Midterm Exam**	

8	Simple Graphics and Image Processing: “turtle” module; simple 2d drawing - colors, shapes; digital images, image file formats, image processing Simple image manipulations with 'image' module (convert to bw, greyscale, blur, etc).	Chapter 7
9	Classes and OOP: classes, objects, attributes and methods; defining classes; design with classes, data modeling; persistent storage of objects	Chapter 8
10	OOP, continued: inheritance, polymorphism, operator overloading (_eq_, _str_, etc); abstract classes; exception handling, try block	Chapter 8
11	Graphical user interfaces; event-driven programming paradigm; tkinter module, creating simple GUI; buttons, labels, entry fields, dialogs; widget attributes - sizes, fonts, colors layouts, nested frames	Chapter 9
12	Multithreading, Networks, and Client/Server Programming; introduction to HTML, interacting with remote HTML server, running html-based queries, downloading pages; CGI programming, programming a simple CGI form.	Chapter 10
13	Searching, Sorting, and Complexity Analysis	Chapter 11
14	**Final Exam**	