Syllabus and Course Information

BU MET CS-521: Information Structures with Python

Welcome to CS-521!!!

I am excited to teach this course. This course will present an effective approach to help you learn Python. With extensive use of graphical illustrations, we will build understanding of Python and its capabilities by learning through many simple examples and analogies. The class will involve active student participation, discussions, and programming exercises. This approach will help you build a strong foundation in Python that you will be able to effectively apply in real-job situations and future courses.

Instructor: Professor Eugene Pinsky

Computer Science Department,

Metropolitan College, Boston University

808 Commonwealth Avenue Room 258

Boston, MA 02215

email: epinsky@bu.edu

Course Times: Tu and Thu 6 – 8 pm

Office Hours:

Each student will be assigned a facilitator who will be available to answer questions. Both facilitators and myself will be available on-line on regular basis to assist you with course related questions.

Course Materials:

(1) Required Textbook: Introduction to Programming Using Python by Y. Daniel Liang (Pearson Publishing),
(2) Recommended: students may purchase access card to MyProgrammingLab for this book. Students are strongly encouraged to practice additional programming problems. These problems can to be submitted electronically for grading to MyProgrammingLab. We may discuss some of these exercises throughout the course.
(3) Course notes (from the course website) – presentation slides
(4) Python Programming Environment – we will be using Spyder IDE (Integrated Development Environment) and Anaconda Python Distribution. We have these installed in our virtual lab. MET Virtual Labs (VLAB) provide students with all required software. Most of the examples presented in class will be run in this environment. You can familiarize yourself with the virtual labs with the information from our website: http://www.bu.edu/metit/pc-labs/virtual-labs/

Additional Resources:

There are many on-line resources available. This is a partial list:

1. http://www.pythontutor.com/visualize.html - this website is very useful and allows to run simple Python programs and visualize the execution. Many of the illustrations in the course notes were generated using this website.
2. https://docs.python.org/2/tutorial - an official Python tutorial
3. https://www.tutorialspoint.com/python - a detailed tutorial with many simple examples
4. https://www.learnpython.org - free, interactive tutorial
5. https://www.python.org/community/sigs/current/edu-sig/ - contains links to learning resources, including two free books

Teaching Approach and Goals

I am a strong believer in learning by using many illustrated examples. These examples will help us build the fundamental understanding of Python and how to use it to solve real problems. Many simple exercises presented in the course will help you develop skills that are needed to use Python effectively in your workplace and more advanced courses.

To accomplish this goal, course materials are divided into a set of pdf files corresponding to particular topic(s). These files will typically consist of three sections:

(1) course material with many examples
(2) interview questions – these are real examples of Python job interview questions collected from various sources in the internet.
(3) sample programming problems – these can be submitted for checking to MyProgrammingLab These exercises are optional and will not be used in computing the final grade.

Please note that material in (2) and (3) is for additional practice only. The homework assignments are from the textbook.

Homework, Grading and Exams:

Final	30%
Project	20%
Homework	35%
Quizzes	15%

class contribution, discussion 5% extra credit

There are six weekly 30 minute quizzes. The final is 60 minutes. All exams are multiple choice and will be done in the blackboard.

This is a programming class and it is essential that students have practice. Solutions to even numbered exercises are available from MyProgrammingLab. Homework assignments will consist both programming and “pencil-and-paper” problems from the textbook.

Quizzes and the final are closed book and will consist of typical Python questions that one can expect at a job interview

The project is open ended and the topics can be chosen by students. Students can work in groups of 2 if desired. In this project, students have to illustrate the usage of different programming concepts covered in class. At the minimum, the project should use a class, a function, at least three container types (lists, strings, dictionaries, sets and/or tuples) and major control flow constructs. Students will present their projects on the last day of the course after the final exam.

The goal of this is to get practice in Python programming and feel comfortable with interview type environments. We focus on presenting many illustrated simple examples to understand Python capabilities. We very strongly encourage and emphasize active student participation and discussions.

Course Syllabus

Course Outline:

The course consists of 7 modules. Each module is 2 weeks for regular class and 1 week for the on-line course. All exercises are from the textbook and need to be submitted to the blackboard for grading.

Please check for updates and new materials as they will be added throughout the course. Note that the homework assigned in week n is due the following week. Due dates are indicated explicitly. No late homework will be accepted.

Week 1

Topics: introduction to computing and problem solving, Python programming environment, Python IDEs, iPython Notebook environment, modules, input/output, running Python, core data types, simple expressions

Reading: Chapters 1, 2

Course Materials:

overview.pdf

types_and_mutability.pdf

Week 2

Topics: variables, immutability, expressions, operators and Boolean expressions, operator precedence

Reading: Chapters 2, 3

Course Materials:

types_and_variables.pdf

Week 3

Topics: mathematical functions, strings and text manipulation, selections, control flow (if, break, continue, for, while) and iterations, files and file manipulation

Reading: Chapters 4, 5, 8, 13

Course Materials:

collections.pdf

control_flow.pdf

files.pdf

strings_indexing_and_slicing.pdf

strings_methods.pdf

Week 4

Topics: collections, set membership and comprehension, lists, tuples, sets, dictionaries, searching and sorting

Reading: Chapters 10, 11, 14

Course Materials:

dictionaries.pdf

lists_indexing_and_slicing.pdf

lists_methods.pdf

sets.pdf

tuples.pdf, sets.pdf

Week 5

Topics: advanced data structures, functions, exception handling, parameter passing, recursive functions

Reading: Chapters 6, 15

Course Materials:

exceptions.pdf

functional_programming.pdf

functions.pdf

Week 6

Topics: objects and classes, attributes, methods, data encapsulation, abstract classes, inheritance and polymorphism

Reading: Chapters 7, 12

Course Materials:

classes.pdf

inheritance_and_polymorphism.pdf

Week 7

Topics: final exam and project presentations

About the Instructor:

Eugene Pinsky received his B.A. in Mathematics from Harvard University and his Ph.D. in Computer Science from Columbia University. He has taught extensively both in academia and industry. His research interests are in performance analysis and computational algorithms in data science and machine learning with emphasis on computational finance and programmatic advertising.