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 illustration, 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

808 Commonwealth Avenue Room 258

Boston, MA 02215

email: epinsky@bu.edu

Course Materials:

(1) Required Textbook: Introduction to Programming Using Python by Y. Daniel Liang (Pearson Publishing),
(2) Students should purchase access card to MyProgrammingLab for this book. Most programming exercises will be from this book and need to be submitted electronically for grading to MyProgrammingLab
(3) Course notes (from the course website) – presentation slides

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

Course Syllabus

The purpose of the course is to learn the fundamentals of Python programming. This is accomplished by presenting a unified object-oriented framework of the language. We provide a detailed discussion with illustrated examples to help understand major language components. This is supplemented by programming assignments, active discussions and quizzes/exams. Upon completion of the course, the students will have a solid foundation in Python programming and will have the background to pursue more advanced courses in Python, Data Science and Big Data.

Homework, Grading and Exams:

70% programming and homework assignments. This is a programming class and it is essential that students have practice. Solutions to even numbered exercises are available from MyProgrammingLab

10% quizzes (2 quizzes, 30 minutes each). These are closed book and will consist of typical Python questions that one can expect at a job interview

10% project – this is open ended and the topics can be chosen by students. Students can work in groups of 2 if desired. The project has to illustrate the usage of different programming concepts covered in class. Students will present their projects on the last day of the course after the final exam.

10% final exam (one hour, in class, closed book). Same style as quizzes – typical Python interview questions

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.

Tentative Course Outline:

The course consists of 7 modules. Each module is 2 weeks for regular class and 1 week for the on-line course

Module	**topics**	**readings**
1	introduction to computing and problem solving, Python programming environment, Python IDEs, iPython Notebook environment, modules, input/output, running Python	Chapter 1
2	core data types, types, variables, immutability, expressions, operators and Boolean expressions, operator precedence	Chapter 2, 3
3	mathematical functions, strings and text manipulation, selections, control flow (if, break, continue, for, while) and iterations, files and file manipulation	Chapter 4, 5, 8, 13
4	collections, set membership and	Chapter 10, 11, 14, 17

	comprehension, lists, tuples, sets, dictionaries, searching and sorting	
5	advanced data structures, functions, exception handling, parameter passing, recursive functions	Chapter 6, 15
6	objects and classes, attributes, methods, data encapsulation, abstract classes, inheritance and polymorphism	Chapter 7, 12
7	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.