Syllabus and Course Information

BU MET CS-521: Information Structures with Python (Fall 2018)

Welcome to CS-521!

I am excited to teach this course. It will present an effective approach to help you learn Python and Object Oriented Design (OOD) that is applicable and common for software languages like C++, Java, C#, etc. 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 not just in Python: you will be able to effectively apply in real-job situations and future courses.

Lecturer: Anatoliy Aleksandrov, Ph.D.
Computer Science Department,
Metropolitan College, Boston University
1010 Commonwealth Avenue Boston, MA 02215

SPRG19 MET CS521 C2, starts on Wednesday, January 23, 2019

CAS 315 Wed 6:00pm 8:45pm

- Welcome!

email: aaleksan@bu.edu

Course Times: Wednesday, 6:00 – 8:45 PM

Place: CAS 315

Office Hours: N/A

Teaching Assistant and Grader: Mahboobeh Vaeznia (vaeznia@bu.edu)

Course Materials:
(1) Required Textbook: Introduction to Programming Using Python by Y. Daniel Liang (Pearson Publishing), ISBN 9780132747189
(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.

Course Id: BOSUNI-4181-0
(3) Course notes (from the course website) – presentation slides.
(4) Python Programming Environment – we recommend to use 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/ However, you can use another IDE (like PyCharm or any kind of online IDE).

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 (excluded the read/write examples with local disk files on your computers).
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 (exception the last one).

Homework, Grading and Exams:

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

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

This is a programming class and it is essential that students have practice. Homework assignments will consist of programming problems from the textbook (and maybe more).

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

The Project is open ended and the topics can be chosen by students. 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 (logical) constructs. Students will present their projects on the two last weeks of the course.

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 Outline:

The course consists of 7 modules. Each module is typically 1-2 weeks. All exercises are from the textbook (plus maybe one from not there). They will be posted as we progress in the course. Due dates for the homework will be indicated explicitly. No late homework will be accepted except by personal requests. As a rule, re-submission is not allowed. If for any reason, ask your professor (don't forget to add the class TA/Grader to the thread).

Please check for updates and new materials as they will be added throughout the course.

Module 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

Module 2

Topics: variables, immutability, expressions, operators and Boolean expressions, operator precedence
Reading: Chapters 2, 3
Course Materials:
types_and_variables.pdf

Module 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:
control_flow.pdf, files.pdf, strings_indexing_and_slicing.pdf
strings_methods.pdf

Module 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

Module 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

Module 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

Module 7

Project presentations and review. Everybody has to present the course Project in the class.

About the course Lecturer:

Anatoliy Aleksandrov received his M.S. and Ph.D. in Civil Engineering in Moscow, Russia. He also took Java developer courses at Harvard University, has more than 22+ year experience as a software engineer (Java, Python, C/C++) and has been teaching at BU MET Python and Java courses in campus and online.

About the course author and coordinator:

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.