MET CS 521: INFORMATION STRUCTURES IN PYTHON
SC2 — SUMMER 2020

Instructor Information

Name: Alan Burstein
Email: alanbur@bu.edu

Class Information

Dates: MAY 20 – AUG 12
Time: Wednesdays, 6 – 9:30 PM
NO CLASS 7/1

Course Description

Only recommended for students with a programming background. Covers the concepts of object-oriented approach to software design and development using the Python programming language. Includes a detailed discussion of programming concepts starting with the fundamentals of data types, control structures methods, classes, arrays and strings, and proceeding to advanced topics such as inheritance and polymorphism, creating user interfaces, exceptions and streams. Upon completion of this course students will be capable of applying software engineering principles to design and implement Python applications that can be used in conjunction with analytics and big data. *Prerequisites: MET CS 300*.

Course Objectives

After this course, you will . . .

- Learn advanced programming techniques
- Gain experience using technology widely used in industry
- Learn to tackle many programming interview questions
- Work on real-world projects
- Develop problem-solving strategies for programming
- Explore real-world applications of the skills and techniques from this course

Teaching Techniques

- We will leverage widely used technology to both facilitate learning and gain valuable experience useful in industry.

- We will focus on the reasoning behind the techniques we study by examining the problems that programmers try to solve.
- We will ensure tangible outcomes throughout the class useful in future classes, industry, and academia.

Resources & Software

There are many resources online to help you study python syntax. In this class, I will be using How to Think Like a Computer Scientist: Interactive Edition,
a free, online, interactive textbook as a supplement to the class lecture and notes.

Required IDEs:
Jupyter Notebook and PyCharm
These are both free to install for students and visit the **Installation Guide** on Blackboard (in the Lecture Slides tab) on a detailed step-by-step on how to install them.

Class Attendance

It is essential to your success in this course that you attend each lecture as the topics covered in a three-hour block can be hefty and difficult to follow on your own.

However, due to the nature of COVID-19, I understand that there may be bigger concerns in your life that take priority. Therefore, I will not make attendance mandatory.

My attendance policy is that for each lecture you attend, you will obtain $\frac{1}{6}\%$ of extra credit added to your final grade. If you attend all lectures, you can receive a maximum of 2% extra credit.

The lectures will be recorded and posted on Blackboard, but attending lectures is vital to learning these topics - especially as we get into more complex fundamentals. I hope this policy will help encourage you all to attend and not disadvantage those in difficult situations.

Homework & Labs

Homework will consist of theoretical concepts and interview-like programming questions.
Labs will be longer and somewhat-guided programming challenges.

To give you flexibility during the semester, you will have 3 "penalty-free" days to use however you wish on homeworks or labs.
After these days are used up, there is a 10% penalty per day that an assignment is late.

Quizzes & Final

The quizzes are open-note, multiple-choice, 60 minutes, and designed to evaluate and reinforce a conceptual understanding of the material.

The final exam is TBD. It will be closed book and will either follow a similar format to the quizzes and be auto-proctored, or will be a take-home exam. More information will be provided in the

future.

Submission Policy

Each submitted assignment MUST contain the following:

- your name,
- the name of any classmates you discussed the assignment with, or the words "no collaborators",
- a list of sources you used (textbooks, wikipedia, research papers, etc.) to solve the assignment, or the words "no sources",
- number of late days used for the current assignment, and total number of late days used up thus far in the semester (include on the current assignment)

Grading

The course grade is determined by the following components:

Homework	20%
Labs	20%
Project	20%
Quizzes	10%
Final Exam	30%

MET Technical Resources

Boston University technical support is available via email (ithelp@bu.edu), the support form, and phone (617-353-4357). Please note that the IT Help Center has multiple locations. All locations can be reached through the previously mentioned methods. For IT Help Center hours of operation, please visit their contact page. For other times, you may still a submit a support request via email, phone, or the support form, but your questions won't receive a response until the following day. If you aren't calling, it is highly recommended that you submit your support request via the technical-support form, as this provides the IS&T Help Center with the best information in order to resolve your issues as quickly as possible.

Academic Integrity

Please visit Metropolitan College's website for the full text of the department's Academic Conduct Code.
For this class, there will be collaborative elements and you are encouraged to work in groups to better understand any material, but all submitted work must be your own, unless explicitly stated.

Communication Policy

If you have any general questions about what was covered in lecture or any theoretical questions

about python, please use **Blackboard Discussions** to ask and I will answer there. Any question you may have might clarify the topic better for other students and save a lot of confusion. This also includes questions about homework assignments, labs, and projects.

If you have a grade concern on an assignment, you should submit a re-grade request through **Gradescope**.

Email should only be used for personal queries or any other concern that cannot be satisfied with the two options above.

Tentative Schedule

The following is a *tentative* schedule for the course. Each module will take 1-2 weeks to complete and each module will have some combination of lab, homework assignment, and quiz associated with the topics discussed.

- Module 1
    - Introduction to Computing
    - Getting Started with Python
    - Setting up IDEs
    - Data Types
    - Input and Output
- Module 2
    - Variables
    - Operators
    - Expressions
    - If statements
- Module 3
    - Lists, Tuples, and Dictionaries
    - Loops
    - Functions
    - Parameters
- Module 4
    - Basics of Classes
    - List Comprehension
    - Errors Handling
    - Reading and writing files
- Module 5
    - Sets and Linked Lists

    - Searching
    - Sorting
- Module 6
    - Recursion
    - Files
    - Intro to Functional Programming
- Module 7: Object-Oriented Thinking
    - Objects and Classes
    - Abstract Classes and Interfaces
    - Data Encapsulation
    - Inheritance
    - Polymorphism
- Module 8: Applications of Information Structures
    - Big Data
    - Visualizing Results
- Module 9: A Touch of Cryptography
    - Hashing
    - Hash tables
    - Bloom Filters
    - Bitcoin

About the Instructor

Alan Burstein graduated with a B.A/M.S. in Computer Science from Boston University. He has a diverse range of industry experience. He worked as a machine learning engineer at Bose Corp and as a full-stack software engineer at Facebook Inc. He now works as a quantitative trader at Susquehanna International Group.