Introduction to Programming
MET CS201 A1 (Summer 2 2020)

Due to Coronavirus restrictions, this class will be held online only.

Syllabus

Course Description

MET CS201 provides an introduction to computer programming. While the skills taught in this course apply to any programming language, the in-class exercises, homework assignments, exams, etc., utilize the Python language.

The Python language features to be covered include, but are not limited to: strings, lists, and dictionaries; flow of control constructs such as 'if' and 'while' statements; functions; file input and output; graphics, graphing and graphical user interface programming. Additionally the course introduces the fundamentals of software development, including application analysis and program design.

Time permitting, additional Python features and programming techniques may be covered, including sets, tuples, iterators, comprehensions, classes, recursion and object-oriented design.

The course also will require students to formulate solutions for certain types of problems (for instance, to count the number of occurrences of a word in a block of text), to write clear and efficient Python code to implement these solutions, and to produce fully-tested, debugged and working programs.

Students will be required to meet weekly with the instructor to discuss their projects, to demonstrate their working programs, and to explain the thinking behind the code they write. Homework exercises will consist mostly of writing code to practice fundamentals.

By the end of the course, students will be able to:

(1) explain basic concepts of how programs work;
(2) explain how to solve certain types of problems with a program;
(3) write short programs in Python without referring to documentation;
(4) explain the nature of the Python features covered in this course, and to provide examples of their use.

Prerequisites

This course is designed for students who are interested in learning to program. No prior programming experience is required. Good algebra and logic skills are strongly recommended. This course will move quickly through a wide range of programming language features that students will be expected to master quickly. Please contact the instructor before registering if you would like to better understand the depth and pace of this course.

Laptops are required at each class.

Location and Time

Mondays and Wednesdays 6:00pm-9:30pm on Zoom from Monday, July 6 to Wednesday, August 12. **Due to Coronavirus restrictions, this class will be held online only.** A final will be held on Wednesday, August 12 from 6:00pm-9:30pm on Zoom.

Students are required to attend each class meeting via Zoom on computers suitable for writing programs in Python. Students may be called upon to write and explain code using screen sharing during class meetings.

Instructor

John Keklak	
Office hours:	no fixed hours; send an email to request a time to meet
Office location:	via teleconferencing only
Email:	jkeklak[at]bu[dot]edu

Required Texts

The text for the course is provided online in Blackboard. We will also be using the Python Software Foundation on-line documentation.

Readings and Exercises

Reading and exercises will be assigned for each week. Students must reproduce the Python examples provided in the online readings, and also must complete all exercises at the end of each chapter. This code must be submitted to Blackboard. Solutions to the exercises are provided in the online text at the end of each chapter.

It is strongly recommended that students devote some time every day to do a portion of assigned readings and exercises.

Additionally, students must submit code written for projects (see "Projects" below)..

Assignment Grading

Assignments are graded as "done" or "not done". Assignments marked as "not done" will adversely affect the final course grade.

Written responses and code in submissions must be clearly-written and well-formatted. Poorly formatted or incomplete code and written responses will be considered "not done". It is recommended that you consider the following practices when preparing your submissions:

(1) complete and well-written sentences and paragraphs for written responses

(2) a consistent coding style

(3) a consistent naming convention for variables, functions and files

(4) use of ample white space

(5) use of names that convey intended meanings

(6) hiding unnecessary details

(7) comments that explain code that you suspect will puzzle someone other than you

(8) rewriting (refactoring) your code once it works

Timely Completion of Assignments

With the very rapid pace of the course, it is extremely important that students complete homework on time. Class exercises, projects, quizzes and the final exam will be very difficult for students who do not complete the examples and exercises in the reading on time.

Projects

One portion of the work for this course consists of the homework exercises which introduce the fundamentals of Python. See "Readings and Exercises" above for details.

Another (and very substantial) portion of the work for this course consists of projects to create applications using Python. The purpose of these projects is to provide students with opportunities to practice applying the Python fundamentals they learn by doing the homework exercises, and to thereby build fluency with Python.

Project work is graded as done/not done, and is designed to allow students to work at their own pace. However, it is expected that students work up to their aptitude for programming as judged by the instructor.

Weekly Project Status Reports and Meetings with the Instructor

Before each Monday class, students must email a status report to the instructor listing the projects they completed, the projects they intend to complete in the coming week, and any issues they have with projects they are currently working on. The status report must be accompanied by the source code of the completed projects, and a log of the hours spent on projects.

Additionally, students must confer with the instructor weekly to discuss their project work. Students must select a day and time during which they will meet via Zoom with the instructor each week.

Exams and Quizzes

There will be three short quizzes during the semester (Monday, July 20, Monday, July 27 and Monday, August 3), and a final exam on Wednesday, August 12. Students may refer to the online text, homework exercises and project work during quizzes and exams.

Grading

Grades will be determined by the following weighting:

Quizzes	-	30%
Final	-	70%
Instructor's discretion	-	+/-10%

Incompletes will not be given.

Quiz grades may be viewed on the course grade sheet in BlackBoard.

Grade conversions:

A	93-100
A-	90-92.999...
B+	87-89.999…
B	83-86.999...
B-	80-82.999…

C+	77-79.999…
C	73-76.999…
C-	70-72.999…
D	60-69.999…
F	59.999… and below

Grades are not curved. The quizzes and the final exam are designed such that a student who has diligently completed the homework and a reasonable portion of the projects, and has participated in all class meetings can realistically earn an A. The quizzes and final exam will not contain any "trick questions", but rather will determine a student's level of fluency with Python and programming.

Attendance/Meetings

Class meetings, and meetings with the instructor, will be crucial components of the learning process, and therefore are required. Missing meetings will adversely affect a student's final course grade. Lecture will occasionally complement and supplement the reading material. Attendance will be taken at the beginning of each class. Absences will significantly affect both your Python skill and your grade.

Be sure to join the Zoom session on time, since it is highly disruptive to have students joining Zoom after class has started. Late arrivals will adversely affect the final course grade.

Collaboration

All course participants must adhere to the Boston University Metropolitan College academic conduct code. Printed copies of the code are available from the college. All instances of academic dishonesty will be reported to the appropriate academic conduct committee.

The material you submit for assignments must be your own original work and it is an act

of plagiarism to represent the work of another as your own. You are welcome to discuss the general concepts in assignments with other students in the course, but it is not acceptable to share, to post, or to copy code or written answers to homework questions. Posting homework questions, project statements or exam questions, or solutions to any of these items anywhere, including on tutoring websites, is a violation of the academic conduct policy. If you are uncertain whether an action constitutes a violation of the academic conduct policy, I will be glad to discuss the matter with you.

In other words, you may discuss concepts with other students, and you may help (or receive help from) other students in preparing your programs and written responses. However, all the material you submit must be code and/or prose that you -- and only you -- wrote. Your code and prose may not include material written by others, or derived from the material of others, regardless of how much you edit such material. In addition to avoiding violations of the collaboration policy, writing your materials from scratch will help you gain a clearer understanding of the concepts and principles presented in this course.

Under no circumstances may you communicate with anyone during quizzes or during the final.

Email Responses

I will check my email each morning at about 9am, and will reply to your emails within 24 hours, barring some unforeseen circumstance.

Expected Course "Roadmap" (subject to change)

This course is a rapid introduction to the Python programming language and to the software development process The first three weeks, in particular, will require a very significant amount of time for homework.

		Topics	**Assignment (to be completed before next class meeting except as noted)**
July	6	Intro to programming and Python printing messages strings and lists basic math and logic the IDLE debugger	Introduction, Chapters 0-4 *(example and exercise code due before class on July 13)*
	8	More math, including random numbers functions	Chapters 5-7 *(example and exercise code due before class on July 20)*
July	13	Python and memory More on strings and lists	
	15	dictionaries file input and output JSON, CSV, TSV file formats and data exchange	Chapters 8-9 *(example and exercise code due before class on July 27)*

July	20	Quiz 1 on Monday, July 20 (covers all material through Chapter 7) lecture tbd	
	22	Graphics, graphing, and graphical user interfaces	Chapters 10-11 *(example and exercise code due before class on August 3)*
July 27		Quiz 2 on Monday, July 27 (covers all material through Chapter 9) lecture tbd	
		lecture tbd	
August	3	Quiz 3 on Monday, July 20 (covers all material in text) lecture tbd	
	5	lecture tbd	
August	10	lecture tbd	
		Final on Wednesday, August 12 from 6:00pm to 9:30pm	

Applications to be written in class include utilities, games and problem solvers such as:

* image generators

* loan payment calculator

* curve fitting

* tic-tac-toe

* poker

* sudoku solver

* Boston MBTA subway solver

Details and instructions will be posted in BlackBoard.