Introduction to Programming
MET CS201 A1 (Spring 2021)

Syllabus

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

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 communicate regularly 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 of writing code to practice fundamentals, and creating small application programs to apply the 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 6:00pm-8:45pm on Zoom from Monday, January 25 to Monday, April 26, except as noted below. A final will be held on Monday, May 3 from 6:00pm-8:45pm.

Class will not be held on Presidents' Day, Monday, February 15, and will be held instead on Tuesday, February 16. Class will also not be held on Patriots' Day, Monday, April 19, and will be held instead on Wednesday, April 21.

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. Unexcused absences will result in deductions from the final course average.

COVID-19 Policies

Although class will meet only on Zoom, students must comply with the following restrictions: Prior to returning to campus, all students are required, through a digital

agreement, to commit to a set of Health Commitments and Expectations including face coverings, symptom attestation, testing, contact tracing, quarantine, and isolation. The agreement makes clear that compliance is a condition of being a member of our on-campus community.

You have a critical role to play in minimizing transmission of COVID-19 within the University community, so the University is requiring that you make your own health and safety commitments. If you will be attending any class in person, you will be asked to show your Healthway badge on your mobile device to the instructor in the classroom prior to starting the in-person class, and wear your face mask over your mouth and nose at all times. If you do not comply with these rules you will be asked to leave the classroom. If you refuse to leave the class, the instructor will inform the class that they will not proceed with instruction until you leave the room. If you still refuse to leave the room, the instructor will dismiss the class and will contact the academic Dean's office for follow up.

Boston University is committed to offering the best learning environment for you, but to succeed, we need your help. We all must be responsible and respectful. If you do not want to follow these guidelines, you must participate in class remotely, so that you do not put your classmates or others at undue risk. We are counting on all members of our community to be courteous and collegial, whether they are with classmates and colleagues on campus, in the classroom, or engaging with us remotely, as we work together this fall semester.

Required Texts

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

Assignments

Reading and exercises will be assigned each week. Students must reproduce the Python examples provided in the reading, 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 reading at the end of each chapter.

It is strongly recommended that students devote some time every day to do a portion of assigned reading and exercises. History shows that regular practice is the most effective way to learn a computer language.

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 homework 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 learned from the text examples and 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. Failure to work up to one's aptitude, as judged by the instructor, will be grounds for deductions from the final course average.

Most project work will involve frequent back-and-forth via email with the instructor. Expect to send and receive at least two email messages regarding projects each week. Typically students' emails will consist of their work on a project, and typically the instructor's emails will contain feedback about a student's work with suggestions for next steps. These email exchanges will allow students to proceed steadily with project work.

Projects will include applications such as:

* loan payment calculator

* curve fitting

* tic-tac-toe

* poker

* sudoku solver

* Boston MBTA subway solver

* image generators

Project details and instructions will be posted in BlackBoard.

Exams and Quizzes

There will be four quizzes during the semester, and a final exam. 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	-	25%
Final	-	75%
Instructor's discretion, taking into account, including but not limited to, unexcused absences, incomplete homework, insufficient progress with projects, missing, late or improper status reports, missed weekly meetings.	-	+/-10%

Incompletes will not be given.

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 complement and occasionally supplement the reading material. Attendance is taken automatically by Zoom, indicating when you log in and when you log out. Absences will significantly affect both your Python skill and your grade.

Be sure to start Zoom five or ten minutes before class to allow time to resolve any technical issues, if any. Late arrivals or early departures will adversely affect the final course grade.

Additionally, if needed, you or the instructor may request a private meeting via Zoom at a time convenient to both.

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 six 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)**
January	25	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 February 1)*
February	1	Lecture topic TBA	
	8	More math, including random numbers functions Python and memory More on strings and lists	Chapters 5-7 *(example and exercise code due before class on February 16*

	16	Lecture topic TBA Quiz 1 (covers all material through Chapter 4)	**Note that class is on Tuesday, February 16, not Monday, February 15**
	22	dictionaries file input and output JSON, CSV, TSV file formats and data exchange	Chapters 8-9 *(example and exercise code due before class on March 1)*
March	1	Lecture topic TBA Quiz 2 (covers all material through Chapter 7)	
	8	Graphics, graphing, and graphical user interfaces	Chapters 10-11 *(example and exercise code due before class on March 15)*
	15	Lecture topic TBA	
	22	Lecture topic TBA Quiz 3 (covers all material through Chapter 9)	
	29	Lecture topic TBA	
April	5	Lecture topic TBA Quiz 4 (covers all material through Chapter 11)	
	12	Lecture topic TBA	
	21	Lecture topic TBA	**Note that class is on Wednesday, April 21, not Monday, April 19**

	26	Lecture topic TBA	
May	3	Final on Wednesday, May 3 from 6:00pm to 8:45pm	