

Introduction to Programming

MET CS201 A1 (Fall 2022)

Syllabus

BU Community COVID-19 Public Health Policies

All students attending class on campus must adhere to all Boston University COVID-19 policies and safety protocols, including vaccination, face covering policy, screening, contact tracing, and testing requirements that may be in effect and updated from time to time.

Instructor

John Keklak	
Office hours:	no fixed hours; send an email to request a time to meet
Office location:	via Zoom or other teleconferencing
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

All classes will be held in person on Wednesdays from 6:00pm-8:45pm in CAS 213 from Wednesday, September 6 to Wednesday, December 7. A final will be held in CAS 213 on Wednesday, December 21 from 6:00pm-8:00pm.

Attendance

Students are required to attend class in person. Lectures will complement and occasionally supplement the reading material. Attendance will be taken at each class, and unexcused absences will result in deductions from the final course average. Moreover, absences will significantly affect your Python skill.

Students are required to attend each class meeting with a computer capable of running Zoom and for writing programs in Python. Students may be called upon to write and explain code using screen sharing during class meetings.

Recordings of the instructor's screen and the class audio from each class will be available on Blackboard.

Late arrivals or early departures will adversely affect the final course grade.

Key Dates

The following dates are important to keep in mind:

(1) Last day for 100% refund of tuition only for on-campus classes; last day to register for or add on-campus classes; semester payment deadline for Metropolitan College on-campus students: **Monday, September 19**

(2) Last day for 60% tuition refund for on-campus classes: **Monday, September 26**

(3) No refunds granted for on-campus classes: **Wednesday, October 12**

(4) Last day to DROP on-campus classes (without a 'W' grade): **Tuesday, October 11**

(5) Last day to DROP on-campus classes (with a 'W' grade): **Monday, November 14**

(6) Last day to take a leave of absence or officially withdraw from the semester: **Tuesday, November 29**

(7) Thanksgiving Recess: **Wednesday, November 23-Sunday, November 27**

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

Homework will be assigned each week. Assignments will alternate between reading and exercises to develop fundamental skills, and projects that utilize those fundamental

skills.

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

Projects

A significant 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 intended to be a "conversation" with the instructor, largely by email. When a project is completed, the student may be requested to email their code to the instructor for feedback. The instructor may reply with recommended changes, or may simply approve the project work for submission. If changes are recommended, the student must make the changes, and send the updated code to the instructor for review. This cycle continues until the project work is suitable for submission.

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.

Assignment Grading

Assignments are graded as "done" or "not done". Late submissions will be marked "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. Quizzes and the final exam will be very difficult for students who do not complete homework on time.

Exams and Quizzes

There will be quizzes every other week during the semester, and a final exam. Students may refer to the online text, homework exercises and project work during quizzes and the final exam.

Grading

Grades will be determined by the following weighting:

Quizzes	-	25%
Projects	-	10%
Final	-	65%
Instructor's discretion, taking into account, including but not limited to, unexcused absences, incomplete homework, insufficient progress with projects, missed classes or meetings.	-	+0/-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.

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 in any way for any reason during quizzes or during the final. Such communication will result in a report of academic misconduct.

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 *VERY* rapid introduction to the Python programming language and to the software development process. It requires a significant amount of time spent on homework on an on-going basis.

Dates correspond to Wednesdays except as noted.

		Topics	Assignments
September	6	Intro to programming and Python printing messages strings and lists basic math and logic the IDLE debugger	Introduction, Chapters 0-4; due before next class
September	13	Project Set 1 briefing	Project Set 1; due before next class
September	20	More math, including random numbers functions	Chapter 5; due before next class
September	27	Project Set 2 briefing Quiz 1 (Chapters 0-4, Project Set 1)	Project Set 2; due before next class
October	5	Python and memory More on strings and lists	Chapters 6-7; due before next class
Tuesday, October 11		Last day to DROP on-campus classes (without a 'W' grade)	
October	12	Project Set 3 briefing	Project Set 3; due before next class

		Quiz 2 (thru Chapter 5, Project Set 2)	
October	19	dictionaries, file input and output JSON, CSV, TSV file formats and data exchange	Chapters 8 and 9; due before next class
October	26	Project Set 4 Quiz 3 (thru Chapter 7, Project Set 3)	Project Set 4; due before next class
November	2	Graphics, graphing, and graphical user interfaces	Chapters 10 and 11; due before next class
November	9	Project Sets 5 and 6 briefing	Project Sets 5 and 6; due before next class
Monday, November 14		Last day to DROP on-campus classes (with a 'W' grade)	
November	16	Project Sets 7 and 8 briefing Quiz 4 (thru Chapter 9 and Project Set 6)	Project Sets 7 and 8; due before next class
November	23	No class	
November	30	Project Sets 9, 10 and 11 briefing	Project Sets 9, 10 and 11; due before next class
December	7	Review for final	

December	14	No class	
December	21	Final (6pm-8pm)	