Introduction to Programming
MET CS201 A1 (Spring 2025)

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 a VERY RAPID 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
+ using Python for web development.

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.

Students are expected to have a cooperative attitude, and to follow assignment instructions. CS201 is not about simply creating programs that run, but rather creating programs using certain established practices that are presented in class, in the text, and in assignment statements.

Students will be required to adhere to coding standards, which specify the Python features they may use, and which Python features they may not use. In short, students must use only the Python features covered in class, in the text, and in projects, and not use additional Python syntax gleaned from online searches or other sources.

Homework exercises will consist of writing code to practice fundamentals, and creating small application programs to apply the fundamentals.

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.

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 HAR 240 (595 Commonwealth Avenue) from Thursday, January 23 to Thursday, April 24. A final will be held on Thursday, May 8 from 6:00pm-8:00pm.

Attendance

Students are required to attend all classes 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.

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 to drop without a ‘W’ grade: **Tuesday, February 25**

(2) Last day to drop with a ‘W’ grade: **Friday, April 4**

(3) Final Exam: **Thursday, May 8**

Refer to the MET On-Campus Calendar for more details.

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 for each class. Assignments will alternate between reading and exercises to develop fundamental skills, and projects that utilize those fundamental 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

Project details and instructions will be posted in BlackBoard.

Assignment Grading

Assignments -- reading, textbook exercises and project work -- 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.

Extensions will not be granted.

Exams and Quizzes

There will be four quizzes and a final exam. All quizzes and the exam will be closed book, closed notes, closed laptop.

Grading

Grades will be determined by the following weighting:

Quizzes	-	20%
Homework and Projects	-	10%
Final	-	70%
Instructor’s discretion, in addition to unexcused absences; incomplete homework; insufficient progress with projects; late assignment submissions; missed classes or meetings; a poor, uncooperative attitude; failure to follow instructions, or resistance in following instructions	-	+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.

Under no circumstances may you use any form of AI, including ChatGPT, to generate code for homework, quizzes or the final exam. Any homework assignment, quiz or the final exam, if found to contain AI-generated code, will earn a 0.

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. Note that this roadmap is subject to change.

		Topics	**Assignments**
January	23	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
January	30	Project Set 1 briefing	Project Set 1; due before next class
February	6	More math, including random numbers functions	Chapter 5; due before next class
February	13	Project Set 2 briefing Quiz 1 (Chapters 0-4, Project Set 1)	Project Set 2; due before next class
February	20	Python and memory More on strings and lists	Chapters 6-7; due before next class
Tuesday, February 25		**Last day to DROP on-campus classes without a 'W' grade**	
February	27	Project Set 3 briefing	Project Set 3; due before next class

		Quiz 2 (thru Chapter 5 & Project Set 2)	
March	6	dictionaries, file input and output JSON, CSV, TSV file formats and data exchange	Chapters 8 and 9; due before next class
March	13	No class - Spring recess	
March	20	Project Set 4 briefing	Project Set 4 projects; draft code due Monday 6pm, final versions due before next class
March	27	Quiz 3 (thru Chapter 7 & Project Set 3) Graphics, graphing, and graphical user interfaces	Chapters 10 and 11; due before next class
April	3	Project Sets 5 and 6 briefing	Project draft code due Monday 6pm, final versions due before next class
Friday, April 4		**Last day to DROP on-campus classes with a 'W' grade**	
April	10	Project Sets 7, 8 and 9 briefing	Project draft code due Monday 6pm, final versions due before next class
April	17	Quiz 4 (thru Chapter 9 & Project Set 6) Project Sets 10 and 11 briefing	Project draft code due Monday 6pm, final versions due before next class
April	24	Class exercises	

May	**8**	**Final (6pm-8pm)**	