INSTRUCTOR: Mahdi Makki, mmakki@hunter.cuny.edu. Office hours by email (you must include your section number in your emails) and by appointment.

WELCOME: This course is **VERY CHALLENGING** and will require **A LOT OF HARD WORK STARTING IMMEDIATELY AND THROUGHOUT THE SEMESTER**, but even with greatest effort, **VERY FEW** of you will get an A, and many of you will struggle to pass the course. However, this course is also very **REWARDING**: you will spend a semester learning to program. This is not something you can do by reading about it or listening to lectures. The only way to make progress is to write programs and that is how you will be spending your time: learning by doing.

WHAT YOU NEED FOR THE COURSE: You will not need to buy a textbook for this course. All instructional materials are provided in this online BlackBoard module free of charge. For more in-depth documentation, you can also refer to the official Python documentation set, available here: https://docs.python.org/3/. In particular, you will find the Library Reference link helpful.

COURSE ORGANIZATION: The material for the semester is divided into units. Each one consists of two parts called "Learn something new" and "Apply what you know."
The first part teaches you some new elements of the Python programming language. As you read through this part, it is essential that you actually create and run all of the example programs.
The second part asks you to write a series of additional programs using what you have learned. If you get stuck, there is a video tutorial showing how to write each of these programs.
Your job is to work through each unit — studying the material, watching the video tutorials, and writing working programs — until you can produce all of the programs in both "Learn something new" and "Apply what you know" sections from scratch.

GETTING A GOOD GRADE: Learning to program is like learning to play guitar or soccer. Every hour you spend practicing makes you a little better and only a lot of hours will make you any good. Read the following line twice — it's your key to getting a good grade in this course:
You must spend at least 9 (nine) hours programming each week. A good plan is to schedule a 90-minute session every day. It is easy to fall into the trap of thinking that one hour a day is "pretty close" It is not. It is one third less. If you get one third less done over the course of the semester, your grade will be significantly worse. If you think 1 hour is pretty close to 1.5 hours, try spending 2 hours. It is just as close to 1.5 and will improve your performance considerably. If you miss hours one week, plan to make them up the next week.

WORKING AT HOME: This should be easy to arrange:
All course materials are available from anywhere through this BlackBoard module. All Video Tutorials are also collected here: https://ww3.hunter.cuny.edu/screencasts/programming-for-everyone.

To write and run Python programs, you will need to download and install the Python interpreter. It is available free here: https://www.python.org/ftp/python/3.5.2/python-3.5.2-macosx10.6.pkg for Mac, here:https://www.python.org/ftp/python/3.5.2/python-3.5.2.exe for PC. You will need to install Python by double-clicking on the file after download. You will always be using the IDLE environment that comes with Python - as is explained in the first unit.
Beginning in Unit 2, many of your programs will use a file called pap.txt. Download it from the Course Materials section of this module. You will need to copy this file to the **same** folder containing your program file.
I am happy to help as far as possible to get you up and running, but there is a limit to how much anyone will be able to assist though — **ultimately your computer is your responsibility.**

ATTENDANCE: This is an interactive online course, in which the work of learning may be performed on your own at a time convenient to you.

TESTING: As you complete each unit, you apply your knowledge to completing an at-home-test ---- one per each of the 10 units.
Answering a test question located in this module's "**TESTS**" area.
Test questions are based on those you have already seen in each unit. You will answer a test question by writing a program. **If you can do so to my satisfaction**, I'll note that you've completed the unit and you can move on to the next one (see note on reasonable programs below). If I'm not satisfied with your program, there is no penalty, but you will have to take the test from the same unit next time. Whether you pass or not, you may make only two test attempts per week: the first one on **Monday or Tuesday** and a second one on **Wednesday or Thursday**. If you pass, the next unit test will become available to you after I grade your work "1". However, if I grade your test "0", you must repeat the test for the same unit (once or more — until you pass it) — **only then the following unit test unlocks for you**. While working on a test question you may not use anything other than the IDLE Python environment — no notes, previous programs, scrap paper, e-mail, smartphones, tablets, etc. A reference sheet is always provided. See the note below on cheating. If you close the test program accidentally before you have had a chance to submit your work, even if it is by accident, **YOU WILL NOT** get credit and you will need to wait until the next week to take another test. You also will not be able to submit any test **Friday - Sunday** (DO NOT EMAIL ME REQUESTING IT AS THIS IS PART OF THE COURSE ORGANIZATION). Tests taken on **Tuesdays** or **Thursdays** must be submitted prior to 8:00 PM (EST)

Testing ends on the last day of classes. There is **NO WAY** to make up missed test attempts: if you register late, or miss the orientation, or start taking tests several weeks into the semester, or are unable to take a test because of an emergency, the test attempts that you missed **CANNOT** be made up for any reason. There are no exceptions to these rules — if your circumstances do not allow for this, you must drop this course.

NOTE: You will be given a password in the announcement section for the tests that you will be taking, which you would need to enter to work on the test question/s.
Once the test unlocks you will find a "Unit**x**.html" link. You must click on the link, which will then open the question that must be answered in a separate window.
Sometimes additional files(links) can be found in addition to the "Unit**x**.html" link. These files must be downloaded (right-click on the file -> save as/download) within the same folder you are working on the solutions file in and used within your program to make it run properly as requested, but do not have to be resubmitted with your python file solution, given that I already have a copy of them on my end.
Once you complete writing the test solution, you must attach and submit the python solutions file (ONLY ONE SUBMISSION WILL BE ALLOWED), otherwise you will automatically receive a "0" and will have to retake the test on the next allowed attempt.

TEST FEEDBACK: You must go to My Grades section after a test attempt is graded. There you can click on each individual test grade and test attempt review whether you got a "1" (meaning you passed and you can proceed to the next unit) or a "0" (meaning you failed the test and you must retake it).
NOTE: Given that the **SAME TEST QUESTION** for the unit can be repeated multiple times until passed, I **CANNOT** provide any direct feedback on the solution program submitted (expect either a "0" or "1").
You are however given a sample output within the "Unit**x**.html" test question, which is a direct reference when writing the solution to the question. (EVEN if the solution output submitted matches the output example provided in the test question, **but** the program is **NOT** written properly as noted in the "REASONABLE PROGRAMS" section below, **YOU WILL** receive a "0" and will have to repeat the question on the next possible attempt).

GRADING: There are 10 units in all. Your semester grade depends on how many unit tests you pass. The following table shows how many units you need for each grade. You need to pass [ ] units to get a grade of [ ]: <3 for F, 3 for D, 4 for C, 5 for C+, 6 for B-, 7 for B, 8 for B+, 9 for A-, 10 for A

REASONABLE PROGRAMS: When you write a program to answer a test question, you are required to make sensible use of the programming tools you have been given. ***YOU WILL NOT*** get credit for a program — ***EVEN ONE*** that functions correctly — if the approach is **not reasonable** (or is not the one specified in the test question) **in my view**. See the Video Tutorial for Program 1 of Unit 1 for some examples of correct, but unreasonable programs.

CHEATING: What is not okay, at all, is to get help of any kind when you're working on a test. Your answers to test questions should be programs created from scratch entirely by you without reference to or input from anything or anyone else. Anything else counts as cheating and will result, at a minimum, in your receiving a failing grade for the semester: "Hunter College regards acts of academic dishonesty (e.g., plagiarism, cheating on examinations, obtaining unfair advantage, and falsification of records and official documents) as serious offenses against the values of intellectual honesty. The college is committed to enforcing the CUNY Policy on Academic Integrity and will pursue cases of academic dishonesty according to the Hunter College Academic Integrity Procedures."