CS-U1 111: Introduction to Programming and Problem Solving
Fall 2017

<table>
<thead>
<tr>
<th>Professor/Katz</th>
<th>Section</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor Katz</td>
<td>EXL1</td>
<td>Mon, Wed 3:00 to 4:20</td>
</tr>
<tr>
<td>Professor Katz</td>
<td>EXL2</td>
<td>Mon, Wed 10:30 to 11:50</td>
</tr>
<tr>
<td>Professor Katz</td>
<td>ILC5</td>
<td>Mon, Wed 4:30 to 5:50</td>
</tr>
<tr>
<td>Professor Moore</td>
<td>ILC3</td>
<td>Mon, Wed 3:00 to 4:20</td>
</tr>
<tr>
<td>Professor Tal</td>
<td>ILC1</td>
<td>Mon, Wed 12:00 to 1:20</td>
</tr>
<tr>
<td>Professor Tal</td>
<td>ILC4</td>
<td>Mon, Wed 4:30 to 5:50</td>
</tr>
</tbody>
</table>

Contact Information:
Professor Tal: itay.tal@nyu.edu, 2 MetroTech Center, Room 10.056
Professor Katz: dkatz@nyu.edu, 2 MetroTech Center, Room 10.027
Professor Moore: pkm266@nyu.edu, 2 MetroTech Center, Room 10.

Office Hours: Will be posted by your instructor

Course Prerequisites: None

Course Description: This course introduces problem solving and computer programming and is for undergraduate Computer Science and Computer Engineering majors who have limited prior experience in programming in any language. The course covers fundamentals of computer programming and its underlying principles using the Python programming language. Concepts and methods introduced in the course are illustrated by examples from various disciplines.

Course Objectives:
1. Basic understanding of how data is represented and how computers execute instructions to use and modify data in order to solve problems.
2. Understanding of computational thinking: how to develop algorithms using decisions, repetition, and decomposition into manageable components to solve problems.
3. Ability to implement and test moderate sized programs in the Python programming language, using constructs including variables, operators, decision statements, loops, functions, and built-in data types (numbers, strings, lists, tuples, dictionaries).

Course Structure
Most of the material will be presented in lectures. Reading assignments from the textbook, programming and other exercises in the lab, and weekly homework assignments will reinforce this material. Programming can be very time-consuming. You should expect to spend a substantial time outside the labs working on the homework assignments.

Assignments and other important announcements will be posted on NYU Classes. You should check the course page every day and sign up for e-mail notification of announcements.
Readings
The recommended textbook for the course is: The Practice of Computing Using Python, 3/E
William F. Punch, Michigan State University
Richard Enbody, Michigan State University
ISBN-10: 0134379764
If you prefer, you may use the second edition of the book, which presents the material in a
slightly different order. There is also an electronic version of the textbook, which is less
expensive. Copies are available in the NYU bookstore, as well as other book vendors. Several
copies are on reserve in the Dibner Library.

Course requirements
Tests: There will be two midterm exams and a final exam. These exams will include some
short answer and/or multiple choice questions, as well as programming problems.

Homework assignments: will reinforce the material covered in the lectures and in the
textbook. Some will be “paper and pencil” exercises and most will involve programming in
Python. Although these count for a relatively small percentage of your grade, it is essential
that you do them and understand the solutions. It is unlikely that you will do well on the
exams if you do not understand how to solve problems like the homework exercises.

Grading:
Grades will be computed roughly as follows:
15% your lowest exam + 20% your middle exam + 25% your highest exam
+ Homework 20% + Lab Grade 20% + Attendance & Participation up to 5% bonus

We may tweak the formula a little, for example, by slightly changing the weights.

Policy on Academic Dishonesty:
Please review the NYU School of Engineering Policy on Academic Dishonesty
http://engineering.nyu.edu/academics/code-of-conduct/academic-dishonesty

In this class, some of the labs will be done with a partner. For homework assignments, you
may discuss the general idea of how to approach a program with other students. You may get
help debugging a program from another student, though it would be better to get help from a TA.
You may NOT show completed or substantially completed code to other students, copy code
written by others, or get others to write code for you. Violations of this policy will result in a
grade of ZERO on the work in question and may result in further disciplinary action.

Moses Center Statement of Disability
If you are student with a disability who is requesting accommodations, please contact New York
University’s Moses Center for Students with Disabilities at 212-998-4980 or
mosescsd@nyu.edu. You must be registered with CSD to receive accommodations. Information
about the Moses Center can be found at www.nyu.edu/csd. The Moses Center is located at 726
Broadway on the 2nd floor.
Important Dates
Tue 9/5: Fall 2017 Classes Begin
9/7 – 9/8: No Labs in First Week
Mon 10/9: Fall Recess, No Class
Tue 10/17: Midterm 1
Mon 11/6: Last day to withdraw with a “W”
Tue 11/14: Midterm 2
11/22 – 11/26: Thanksgiving Recess, No Class
Tue 12/12: Legislative Day - Classes will meet according to a Monday schedule
Fri 12/15: Last Day of Fall 2017 Classes
12/18 – 12/22: Fall Semester Exams: The final exam could be any time during this week. Do not buy plane tickets or make any other travel arrangements that involve leaving before 12/23!!!

Course Topics
• Introduction (chapter 0)
• Binary representation of numbers and other positional number systems
• Parts of a program, variables (chapter 1.1 – 1.5)
• Basic Data Types (chapter 1.6)
• Operators and Expressions (chapter 1.7, 1.9, 1.11)
• Using modules: math, random, turtle (chapter 1.8 - 1.10, 2.3)
• Booleans and Boolean expressions (chapter 2.1, 2.2.1 – 2.2.7)
• Selection Statements: if, if–else, if-elif-else (chapter 2.1, 2.2.8 – 2.2.9)
• Iterative Statements: while, for (chapter 2.1, 2.2.10 – 2.2.15, 2.5)
• Strings (chapter 4)
• Functions (chapter 5, 8)
• Lists and Tuples (chapter 7)
• Files, input/output and exceptions (chapter 6)
• Object Oriented Programing (chapter 11.1 – 11.7)
• Dictionaries (chapter 9.1 – 9.3)