COM110 Fall 2015
Programming Assignment 1

Due: Wed, Sept 16 by 9:30 pm.
Late deadline: Sat, Sept 19 by 9:30 pm. (1 point penalty per 24 hours late until this date.)
After late deadline: 10 point penalty per 24 hours late.

Using Python, complete all three parts of the assignment below.

Each of the following programs you will create should begin by outputting a clear description of what the program does so that the user knows what to expect.

Before you start coding, here are some points to keep in mind:

- For Program A, make sure there is a purpose to your program; be creative.
- Comment liberally throughout your code, and certainly at the beginning of the file; the comments at the top should include your name, the assignment number, the due date, and a well-written description of the module.
- As always, make sure your variable names are descriptive.
- You will be graded on user-friendliness of your programs. Is it clear to the user (who you must assume is not a programmer and has not seen your code) what the program is for? Is it obvious to the user how s/he should enter input and what the output means?
- Note that the indicated point breakdown doesn’t necessarily represent the difficulty or size of the program. In fact, program A may be the simplest, depending on what you decide to create. Program C on the other hand, may be the trickiest.
- See the grading rubric linked from course website for more grading info.
- Submit three modules (.py files), one for each program. Your files should be placed in a folder called proj1_xxxxx, where xxxxx is your login name. E.g., my folder would be called proj1_cchung. Compress this folder into a single .zip file and then upload it to moodle. We will collect it from there.

Program A (50%): Create your own.

The purpose of this program is entirely up to you, but the following components are required:

- A main function which is also called at the end of the module
- User input and user-friendly instructions and information
- Some numerical computations
- Some variables
- At least one loop
- Some output

You can get some ideas from class, from lab, from examples given in the book and from exercises at the ends of the chapters. If you use an idea from something we’ve done in class or lab, make sure it is not just a minor variation, but is substantially different. You may wish to
consider ideas that are relevant to another class you are taking, your extra- or intra-curricular interests, or current events. Everyone’s Program A should be different. Remember to write a clear description of the program for the user before the first user-prompt so that the user knows what to expect and understands what the program’s purpose is. It is a good idea to run your idea past a TA or the professor before starting.

**Program B (45%):** Height conversion and averages.

Write a program that someone would use to convert a group of people’s heights from English to metric and then find the group’s average height. First, prompt the user for the number of people whose heights s/he wishes to average. Convert each height given by the user from feet and inches to meters (or centimeters). Finally, output the average height in meters (or centimeters) as well as in feet and inches.

A sample run of the program should have the following behavior:

**How many people are in your group? 3**

**Person 1**

Enter height in feet and inches, separated by a comma: 5, 5

Height of person 1 in feet and inches: 5'5"
Height of person 1 in meters: 1.65

**Person 2**

Enter height in feet and inches, separated by a comma: 6, 2

Height of person 2 in feet and inches: 6'2"
Height of person 2 in meters: 1.88

**Person 3**

Enter height in feet and inches, separated by a comma: 5, 9

Height of person 3 in feet and inches: 5'9"
Height of person 3 in meters: 1.75

Average height of your group in meters: 1.76
Average height of your group in feet and inches: 5'9.3"

Remember to write a clear description of the program for the user before the first user-prompt so that the user knows what to expect and understands what the program’s purpose is.

**Program C (5%):** Fibonacci sequence.
Write a program that someone would use to learn the numbers in the Fibonacci sequence. (If you don’t know what the Fibonacci sequence is, refer to this Wikipedia article: http://en.wikipedia.org/wiki/Fibonacci_number.)

Your program should prompt the user for a positive integer, $n$, and then print out the $n$th Fibonacci number. A sample run might look like this:

The first two Fibonacci numbers are 0 and 1. Enter a positive integer greater than 2 --> 10

The first 10 Fibonacci numbers are: 0 1 1 2 3 5 8 13 21 34. So the 10th Fibonacci number is 34.

Your program should use a single for loop to achieve this. (If you, or a TA helping you, are tempted to solve this recursively, don’t. At least not until later when we actually learn what recursion is.)

Again, remember to write a clear description of the program for the user before asking for input so that the user knows what to expect and understands what the program’s purpose is. It would be great if your program actually started out with a brief explanation of the significance of the Fibonacci sequence! Great enough for bonus points? Possibly. (But if you take text directly from a web page or somewhere else, remember to cite your source!!)