Computer Science 1 — CSci 1100
Lab 0 — Getting Started
Fall Semester 2016

Lab Overview

The primary goal of this lab is get everyone started with the Python environment you will be using this semester, including everything from Python modules, to setting up a Dropbox, to starting to use the Python IDE. The secondary goal is to have a little fun playing with the example Python program from Lecture 1 where we found the one word in the English language that has three consecutive double letters. If you complete these two goals — Checkpoints 1 and 2 below — you will have done well in this first and perhaps chaotic lab.

Normally, we would grade you on each of these checkpoints — usually three. Today, we will go through the motions of doing so, but everyone who completes this lab will receive full credit without any time limit.

Once you are finished with the checkpoints, you are welcome to leave. If you are unable to finish this lab during the allotted time, do not worry, but please complete this lab afterwards and, if you need it, seek help from TAs during their office hours (see Piazza course website — you can visit **any** TA for any section) or from other students in the class! In future labs, we will only give you full credit if you complete the lab during lab time.

Throughout the lab period, if you have a question, start by chatting with your neighbor to see if they might have the answer or have a similar question. You are encouraged to help each other as much as possible. If you and your neighbor do not have the answer, raise your hand and ask the grad TA or the undergraduate mentor for help. If many students are requesting help at once, you may need to be patient. That's why it is better to start by asking your fellow students in the lab.

Checkpoint 1: Setting Up The Environment

Please proceed through the following

1. Set up a Piazza account by going to

   `http://piazza.com/rpi/fall2016/csci1100/home`

   and providing your *rpi.edu* email address.

2. Go to the course web site:

   `http://www.cs.rpi.edu/academics/courses/fall16/cs1/`

   and click on the link for setting up the Python environment. This page contains the instructions for setting up the Python environment you will be using for this class.

3. Please proceed through the installation steps, all the way through getting the IDE and starting Dropbox. At the end, you should have your environment running, have tested the WingIDE, and have a Dropbox (or equivalent cloud storage) all set up. In short, you should be ready to go!

In order to complete this checkpoint, raise your hand and show a TA or a mentor that (a) you are signed up for Piazza, (b) you have the environment installed and tested, and (c) you have a Dropbox set up.

Checkpoint 2: Playing with the Triple Double Letter Code

In this checkpoint you will download and play with the Python code we examined in Lecture 1. Start by making sure you have created a folder for Lab 1 in your Dropbox. Then, proceed with the following steps:

1. Go back to the **Resources** page on the course Piazza website, now click on the tab called **Resources**.

   Locate the link for the example code from Lecture 1. Download the file `three_double.py` you see there, storing it in your folder for Lab 0.

2. Start up the Wing IDE.

3. Under the WingIDE File menu, load the `three_double.py` file you just downloaded. You will see the code in the WingIDE editor pane.

4. Look for the green arrow icon across the top of the IDE. Click on this to run the program. You will see it running in the pane on the lower right. This pane is the Python interpreter.

5. Now, see if you can figure out how to change the Python code in `three_double.py` to count the following:

   (i) number of words that have at least two consecutive pairs of double letters,

   (ii) number of words that have three pairs of double letters, and one letter in between each pair (for example: suddenness),

   (iii) number of words that have three pairs of double letters, and two letters in between each pair.

   In each case, there are two types of changes to make. Try and see if you can figure at least one of these out. Try to get them all. But,

   (a) Don't worry if you can't figure it out. It should be fun to try, and ask a fellow student, a TA or a mentor if you can't!

   (b) If you make mistakes, see if you can distinguish between syntax errors and semantic errors. Since you are unfamiliar with the syntax of Python at this point, only make changes slowly and refer back to the original version if you run into too much trouble.