

CptS 111 Laboratory 1:

Python, IDLE, and Submitting Homework

IMPORTANT: Some labs will require you to do a fair amount of reading. *Be patient and read everything in the lab assignments!* If you do so, you should find these labs are relatively straightforward and you are likely to complete all the tasks more quickly than if you skip sections of the reading. (However, for this first lab, if you are a Windows user, you don't have to read the information for Mac users and vice versa.)

Goal: The goal of this lab is to ensure you have Python running and that you are familiar with the submission system that will be used to submit future homework assignments. (This submission system will *not* be used for future labs.) All your work is to be completed in the lab.

Lab Structure: Your overall lab grade is based on your completion of certain tasks within each individual lab. Attendance is also a component of your lab grade. Each lab will identify the tasks for which you must document completion to the TA (i.e., the tasks for which the TA will record that you have successfully completed the work). The tasks are structured so that full credit is awarded for successful completion of the task and no credit is awarded if the task is not successfully completed.¹ Unless explicitly stated otherwise, the tasks should always be worked in order. Thus, other than for the first task, you will not be awarded credit for a task if you have not successfully completed the previous task.

This first lab is different from subsequent labs in that its primary objective is merely to ensure you have the tools in place to start working with Python and to submit homework. Thus, for this first lab, you should work through the tasks at your own pace, seeking the assistance of the TA when needed. But, you should also:

- Ensure the TA records your attendance, i.e., *introduce yourself to the TA!*
- Ensure the TA records that your two programs, written in Task 3, are properly commented and run correctly.
- Ensure the TA records that you have successfully completed the submission of your work to the homework-submission Web site (as evidenced by the email acknowledgment you are sent).

Finally, take some time to get to know your classmates. You may assist each other in the labs, provided that assistance serves to enhance learning. But, please do not talk excessively about things unrelated to CptS 111—save that sort of socializing for other venues.

¹However, awarding of partial credit is left to the discretion of the TA.

Task 1: Installing Python.

Install Python on your laptop (assuming you have one). You were supposed to do this prior to the lab. If you have not done so already, go to

<http://python.org/download>

and download the Python 3.2.2 package that is right for your system. If you encounter any problems, contact the TA.

Mac Users: Assuming you have version 10.6 or 10.7 of the Mac operating system, you should probably use “Python 3.2.1 Mac OS X 64-bit/32-bit x86-64/i386 Installer (for Mac OS X 10.6 and 10.7)” available from the Python Web site. From the Finder (or from the browser’s download window) double-click the downloaded disk image file (with the `.dmg` extension) and then double-click the Installer package that appears (the `.mpkg` file). You should be able to use all the default settings for the installation. The package will appear in your Applications folder.

Also, after installing Python you should install ActiveTcl. To do this, go to

<http://www.activestate.com/activetcl/downloads>

Assuming you are running version 10.5 or higher of the Mac operating system, click the button that says “Download ActiveTcl 8.5.11 for Mac OS X (10.5+, x86_64/x86).” Double-click the disk image file (i.e., the `.dmg` file) and then double-click the `.pkg` file that appears. Use all the default settings for the installation.

Task 2: Running Python.

Here we will use IDLE which is a front-end or “integrated development environment” (IDE) for Python. It provides some nice extra features compared to running Python directly. (The following description is primarily based on what happens when you use one of the departmental machines running Windows XP and version 3.2.1 of Python. There may be some minor differences with what you see on your particular system. If you are a Mac user, there is Mac-specific information below.)

One way to start Python is to click the `Start` button in the lower left corner of the screen. A window should pop up. If you don’t see any mention of Python, click `All Programs`. You will eventually see a large listing of programs. In this list there should be an entry that says `Python 3.2`. Clicking on `Python 3.2` will bring up another list in which you will see `IDLE (Python GUI)` (`GUI` stands for Graphical User Interface). Click on `IDLE (Python GUI)`.

At this point a window should appear that is labeled `Tk Python Shell`. The interactive Python prompt (`>>>`) lets you know that the Python interpreter is awaiting your commands or statements. (Feel free to play around with entering expressions and statements!)

Mac Users: After installation you should find a directory called `Python 3.2` in your Applications folder. Clicking on that leads to a directory containing the applications `IDLE` and `Python Launcher`. Double-click `IDLE` to start it. (If you are a hard-core user of the Terminal application, you can start Python via the Terminal command-line by typing `python3`.)

Task 3: Creating Files within IDLE.

Click `File` in the upper left-hand corner of the window. Click `New Window` or, alternatively, you can type `Control-N`, i.e., hold down the control key and then type the `N` key. On a Mac you would type `Command-N`. In the following we will just write `C-X` with the understanding that “`C`” is the control key on a Windows machine or the command key on a Mac and you hold this down while typing the “`X`” key. A new window should appear that is labeled `Untitled`. Note that the interactive prompt will not appear in this window! That is because Python will not execute the commands we type in this window until we tell Python we are ready to have them executed.

Click the `Untitled` window to select it. Now type the following

```
# your name
# the date
# CptS 111
# Lab 1, Part 1
print("Hello world!")
```

For `your name` and `the date` fill in your name and the date. (Python will ignore text to the right of the pound sign as they are comments.)

Now you need to save your work. Click `File` and select `Save` or type `C-S` (more on this below for Mac users). A new window appears that requests information about how to save your file. I suggest you click the `My Documents` icon that appears on the left. Then, create a new folder named `Python` or perhaps `CptS111`. To create a new folder, click the icon in the upper right that looks like a folder with a sparkle in one corner. A new folder will appear and the text will be selected. Type the name you want to use—I’ll assume `CptS111`. Now, double-click this new folder. Let’s create one more new folder named `Lab1` (so you’ve created `CptS111` and within that folder you’ve created the folder `Lab1`). Now double-click `Lab1`. Finally, at the bottom of the window is a field labeled `File name:`. In that field write `hello.py`. Click the `Save` button to the right of the field or simply hit the `Enter` key.

Mac Users: After you type `C-S` and the `Save` window pops up, there is a field in which you can type the name of the file, i.e., the `Save As` field, and there is a pulldown menu that specifies the folder in which to save the file. To the right of the `Save As` field there is a button with a downward-pointing arrow. You should click this button. That expands the window to reveal more of the file structure. You will probably be in your `Documents` directory. Click the button labeled “`New Folder`”. Enter “`CptS111`” for the new folder’s name and click

“Create.” Again click the “New Folder” and now type “Lab1” and click “Create”. Now enter `hello.py` in the `Save As` field and click “Save.”

At this point the label of the window in which you typed the code will change to reflect the name of the file you just saved. Along the top of this window, in addition to the `File` menu, there is a `Run` pull-down menu. Click `Run` and then click `Run Module` in the menu (or hit the `F5` function key; on a Mac laptop hit `fn-F5`). If everything worked properly, `Hello world!` should appear in the Python Shell window.

If you have read all of Chap. 2, you already know that “`**`” is the exponentiation operator. The expression `2 ** 10` evaluates to $2^{10} = 1024$ and, more generally, `x ** y` evaluates to x^y (using whatever values have been assigned to the variables `x` and `y`).

Now, let us create a new file. Again go to `File` and select `New Window` or type `C-N`. Enter the following code

```
# your name
# the date
# CptS 111
# Lab 1, Part 2
x = 3
print(x, "to the 100th power is", x ** 100)
x = 4
print(x, "to the 100th power is", x ** 100)
```

where, as before, you should write your actual name and today’s date. Save these commands to a file called `powerOfHundred.py` by selecting `File` and `Save` or hitting `C-S`. Make sure you save this file in the same directory where you saved `hello.py`, i.e., the `Lab1` directory. Try running this file (select `Run` then `Run Module` or use the `F5` function key).

You’re done creating files/programming! Please demonstrate your running code to the TA so he can sign you off on this task.

Task 4: Create a Zip File of the `Lab1` Directory.

Now you must submit your work, i.e., you must submit the `hello.py` and `powerOfHundred.py` files. The way in which you will submit this work is also how you will submit future homework assignments (but not future labs!). To submit this work, we need to wrap both files up into a single file. To do this we will create a zip file that bundles together the `Lab1` directory and its contents.

There are a couple of ways to do this. The simplest way is to use Windows to navigate to the `Lab1` directory (i.e., find the icon for this folder on your computer—this has nothing to do with Python or IDLE at this point). Right-click the `Lab1` icon. This brings up a menu from which you select `Send To` and then you select `Compressed (zipped) Folder`.

This should create a zip file in the same directory in which the Lab1 folder resides. I believe this file will automatically be named Lab1.zip although, depending on your preferences, you may not see the “.zip” extension. The icon for the file should be the typical folder icon but with a zipper across it.

Mac Users: Use the Finder to select the folder where you saved your two Python files. Click the Files menu at the top of the page. You should see an entry that says “Compress “folder-name”” where folder-name is the name of the folder. Click this menu entry. A new file should appear that is the name of the file with “.zip” appended to the name.

Task 5: Submit your work, i.e., your zip file. Start a Web browser (e.g., Firefox or Chrome or Internet Explorer). Go to

<http://eecs.wsu.edu/~schneidj/cpts111/HwSub>

Click the link to your lab section and then follow the instructions there to submit your zip file.

You should receive confirmation email stating that your submission has been received. *This will be sent to your WSU address.* Check with the TA to ensure your submission has been received and that everything looks okay. If the TA agrees that your submission is correct, you are done! You may leave.

Enjoy the three-day weekend but complete reading Chaps. 1 and 2 before Wednesday’s lecture and study hard!