EGR 680
Fall 2018
Laboratory #6

Introduction to Python & the Spyder IDE

Due Date: By beginning of next lab section

Objectives

- Download and install the Anaconda Distribution
- Get familiar with the Spyder Python IDE
- Expand your understanding of the Python language
- Create a simple Python program & an ATM application in Python

---

Third and Final Language of the Course!

Up to this point in the course, you have become familiar with low-level languages like Verilog and C. You have probably noticed how fussy Verilog and C can be when it comes to syntax. A missing or misplaced semicolon can lead to unexpected results or errors. That's where the high-level language Python rises above its lower-level counterparts. Python allows you to think like a programmer and not worry about the picky syntax. Python is an easy to use, powerful and versatile language.

Part I – Installing Anaconda & Launching the Spyder IDE

The Anaconda Distribution is a Python environment that is tailored to the data science and engineering fields. There are over 1,000 data science packages that can be installed and managed from Anaconda. For this course, we will utilize the Spyder and Jupyter environments. Anaconda comes pre-loaded with these two environments, so there is no need to install any other packages.

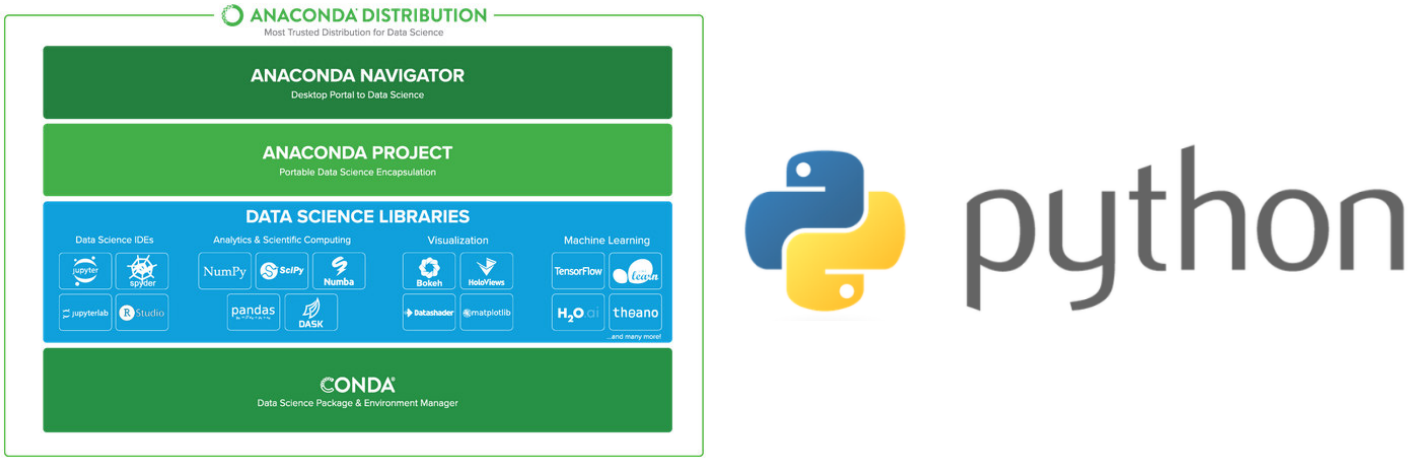

Figure 1: The Anaconda Distribution

1. Before any programming begins, Anaconda needs to be installed on your computer. Type the link below in your web browser to bring up the Anaconda distribution page.
   - https://www.anaconda.com/distribution/

2. At the distribution page, click the **Download Now** button at the top of the page. This link brings you to the distribution download page. Scroll down until you see the **Python 3.6** Windows version download

button. For this course, it will be assumed that you are using Microsoft Windows. Click on the **64-bit** or **32-bit** (depends on your computer) **Graphical Installer** link. Click **Save File** when the dialog box appears.

3. When the download completes, browse to your Downloads folder on your PC and **right-click** the **Anaconda3-5.0.1-Windows-xxx_xx.exe** file. Click **Run as Administrator** and click **Yes** at the **User Account Control** screen.

4. Click **Next** on the **Anaconda Setup** screen, then click **I Agree** on the **License Agreement** screen.

5. Click **All Users** on the **Select Installation Type** screen.

6. Make sure that you are installing **Anaconda** in the correct **Destination Folder** and then click **Next**. See below for the correct installation location.

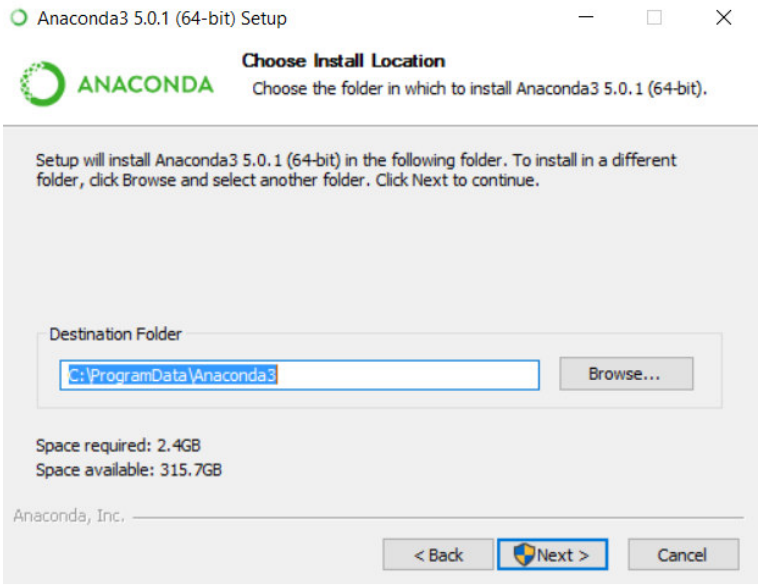

Figure 2: Anaconda install location

7. At the **Advanced Installation Options** screen, **ONLY** select the **Register Anaconda as the system Python 3.6** option and then click **Install**. Note: This installation will take a long time to complete. I suggest you review the lecture slides on the Python language while you wait.

8. When the installation completes, click **Next.**

9. Deselect the **Learn more** boxes and click **Finish**.

10. Now, **Run** the **Anaconda Navigator** desktop app.

11. If you have done everything properly, you should now see the **Anaconda Navigator** app open as shown in Figure 3 below.

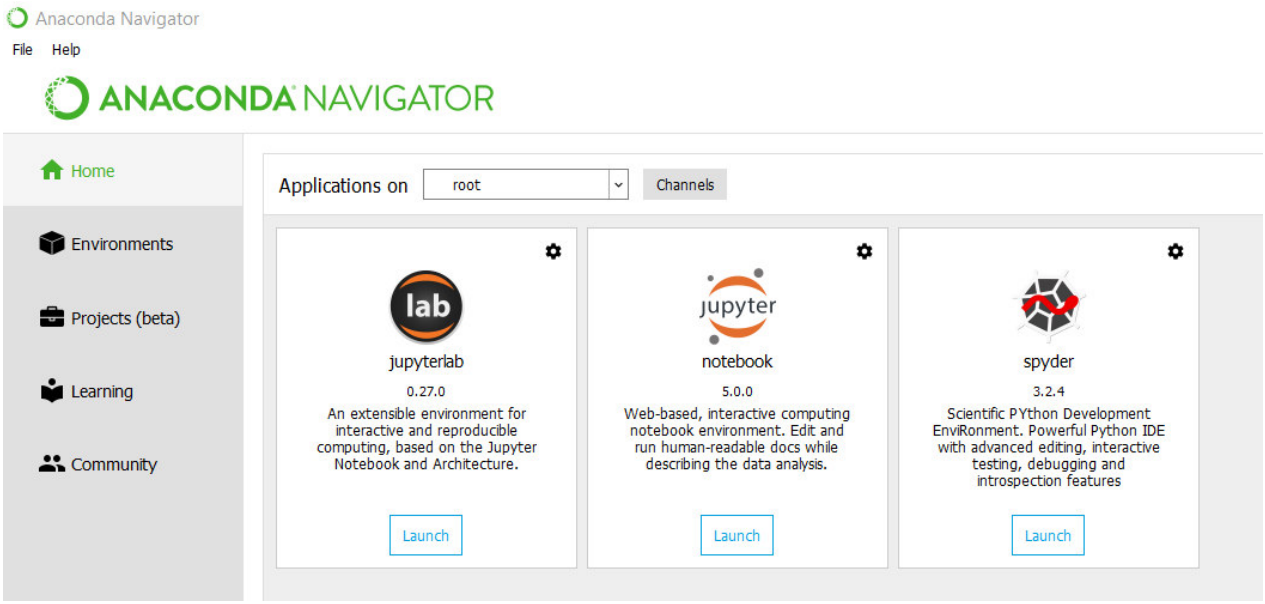

Figure 3: Anaconda Navigator

12. Launch the **spyder IDE** from the navigator screen. This may take a minute to launch for the first time.

13. When **spyder** opens, you should see a screen as shown in Figure 4.

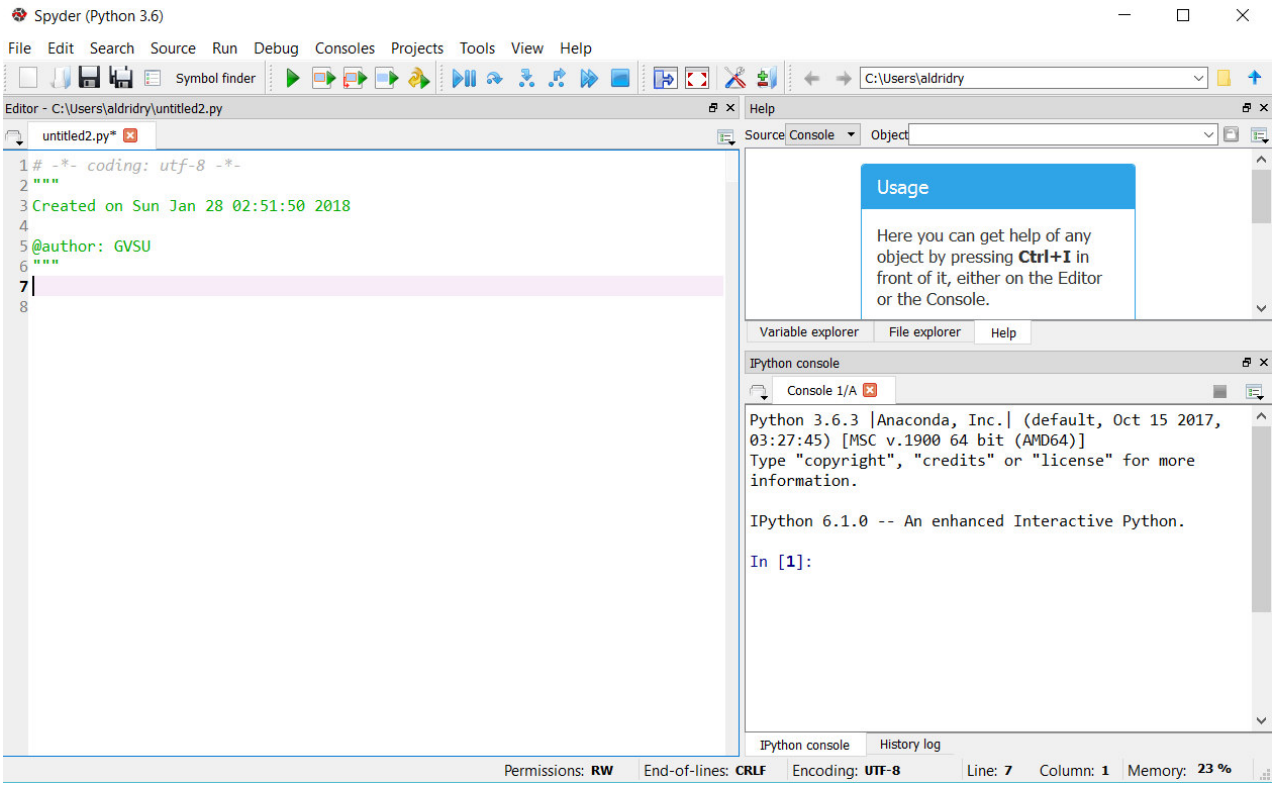

Figure 4: Spyder IDE

Part II – Introduction to Programming in Python

We will create a simple **Python** program to get familiar with the tool and the language. The simple program will print **"Hello Human"** to the **IPython Console** and wait for the user to respond with ***"Hello Computer"***. After the user responds with the correct response, the computer should say ***"Goodbye Human"*** and wait for the user to respond with ***"Goodbye Computer"***. After the final goodbye, the program should exit. This program will introduce **print** statements, **conditional** statements, and **loops**.

1. Select **File → New File** to create a new file, or you can continue on the blank file that opened when you launched the **IDE**.

2. Select **File → Save As** and name your program ***"hellogoodbye.py*** and then click **Save**.

3. Copy the code shown in Figure 5 into your **spyder** editor. Pay close attention to indentation and spacing!

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Sun Jan 28 03:07:28 2018
4
5  @author: GVSU
6  """
7
8  user = 0                          #set up the user variable since we need
9                                    #to compare in the while loop before using
10
11 print("Hello Human")              #print to the screen
12
13 while user != "Hello Computer":   #start the loop and only accept
14                                   #the correct string
15     user = input()                #assign keyboard input to "user"
16
17     if user == 'Hello Computer':  #Checking for correct string
18
19         print("\nGoodbye Human")  #if the string is correct, print
20
21     else:                         #if the string is incorrect, print
22
23         print("\nIncorrect Greeting!")
24
25
26 while user != "Goodbye Computer": #start the loop and only accept
27                                   #the correct string
28     user = input()                #assign keyboard input to "user"
29
30     if user == 'Goodbye Computer': #Checking for correct string
31
32         exit                      #if the string is correct, exit
33
34     else:                         #if the string is incorrect, print
35
36         print("\nIncorrect Greeting!")
37
```

Figure 5: Simple Python program

4. After you finish copying the code above, click the **Run File** button or press **F5** to execute your program. You should see your program output in the **IPython** console.

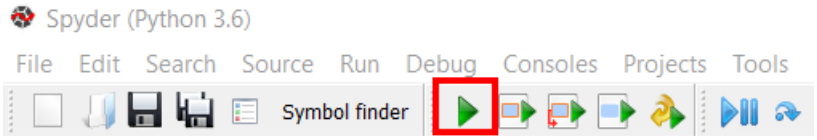

Figure 6: Running the program

5. Type ***"Hello Computer"*** in the **IPython** console and press **Enter**. You should see the computer response.

6. Type ***"Goodbye Computer"*** in the **IPython** console and press **Enter**. After your press enter, the program should exit. You program should see an output similar to Figure 7.

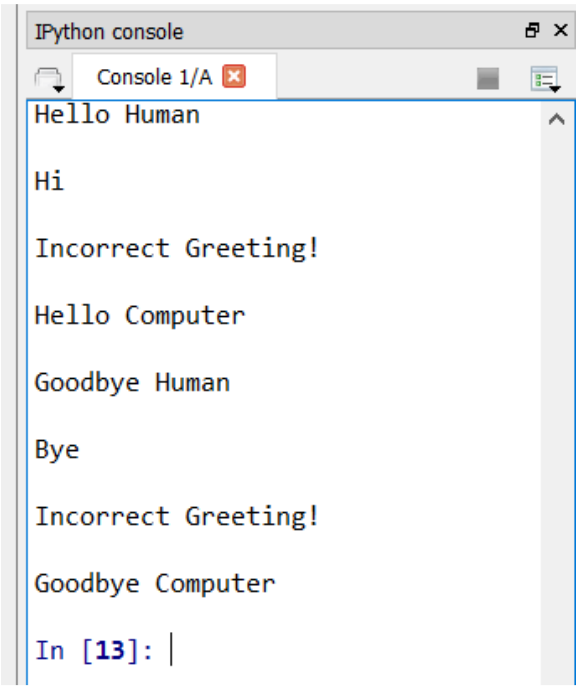

Figure 7: Simple Python program output

7. Select **File → Close** to exit the simple python program.

Part III – Python ATM Machine

In this part, you will be tasked to simulate an Automated Teller Machine (ATM). An ATM machine acts as a mobile banking device that allows customers of banks perform financial transactions such as withdrawals and deposits. Your **Python** ATM should welcome you and ask for a valid Personal Identification Number (PIN). You should then be prompted with options such as check balance, withdraw cash, deposit cash, change PIN, and exit. Your ATM should keep a running balance throughout the duration of the program.

Application Behavior:

1. The greeting should display the logo of your bank.
2. After the logo, the ATM should prompt for a valid PIN number. Invalid PINs are not accepted!
3. After a valid PIN has been entered, the ATM should provide a menu of the following options:
    a. Check balance
    b. Withdraw cash
    c. Deposit cash
    d. Change PIN
    e. Exit
4. The ATM should not allow a customer to take out negative money. This is not a credit machine!
5. The ATM should allow unlimited transactions until the customer selects **Exit** from the menu.
6. The ATM should generate short delays between transactions for readability.
7. The ATM should provide a goodbye message to the customer upon exit.

The below figures are example transactions of the ATM. Your ATM should behave in the same manner.

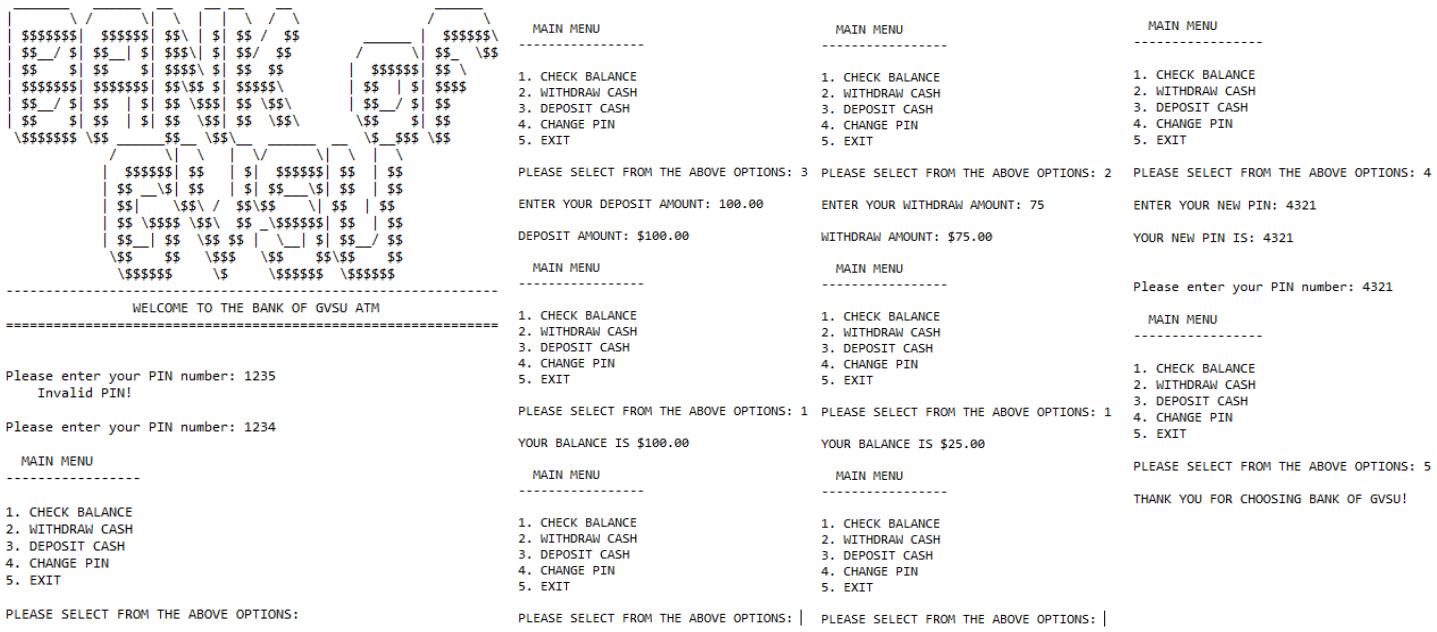

Figure 8: Expected ATM outputs

Laboratory Deliverables

You are required to turn in a hard copy of the report. Report should have the following items:

- Cover page with few paragraphs description of Part III
- Printout of your code with comments (Part III)
- Snapshot of your outputs (highlighting various conditions that you have tested)

You also have to demonstrate your design and turn in a compressed version of the project.