Info 6079: Security Application lab 1

FALL 2022

Lab 1 objective:

- Install Python 3.10.x or later
- Install Wing 101 or Wing Personal
- Familiarize with Wing IDE
- Familiarize with basic code in Python
    - Working with numbers
    - If statement
    - Getting information from user

Lab Time:

You will be required to submit the slides for this lab to dropbox no later than 10 minutes before the end of the class. Late submissions will receive 0 marks.

Lab Scoring:

This is a Lab is not graded. It is for you give you time to install Python and Wing IDE.
At the same time to give to the chance to see how to work with Wing and to code or code in Wing, if you have not done so.

Download Files, if you have not done so already

- Python 3.10 or later: https://www.python.org/downloads/
- Wing 101 or Personal: https://wingware.com/downloads

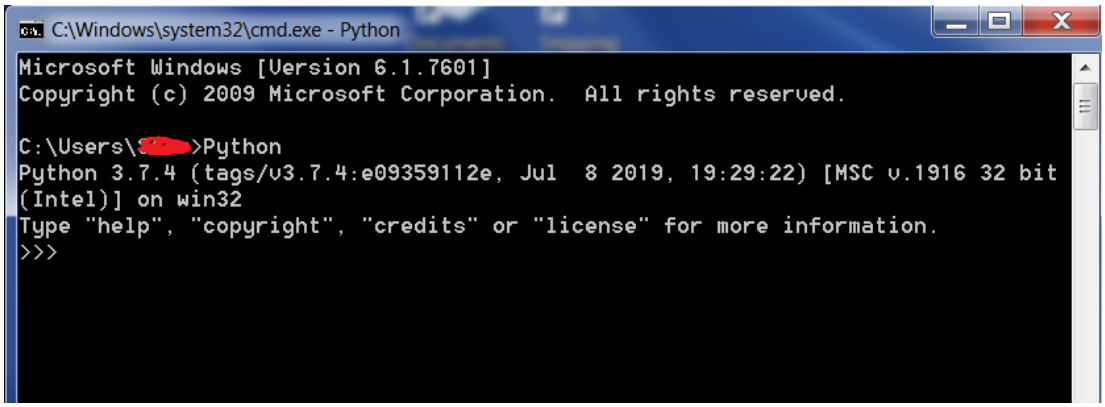
Start of Lab

1. Start by installing Python 3.10 or later
    1.1. Run the execution installation file of Python and start running it
    1.2. When you reach the option to Add Python 3.10 (or later) to Path do so
        1.2.1. This will give you the option to get to python from any where in your computer
    1.3. Select where you would like Python 3.10 program be saved and press Install
    1.4. Follow the rest of the instructions
2. Install Wing IDE, Either 101 or Personal.
   Both are free and both are suitable for the course
    2.1. Fallow the instructions for installation
3. Check if you can open Python 3.10 + through command prompt

FALL 2022

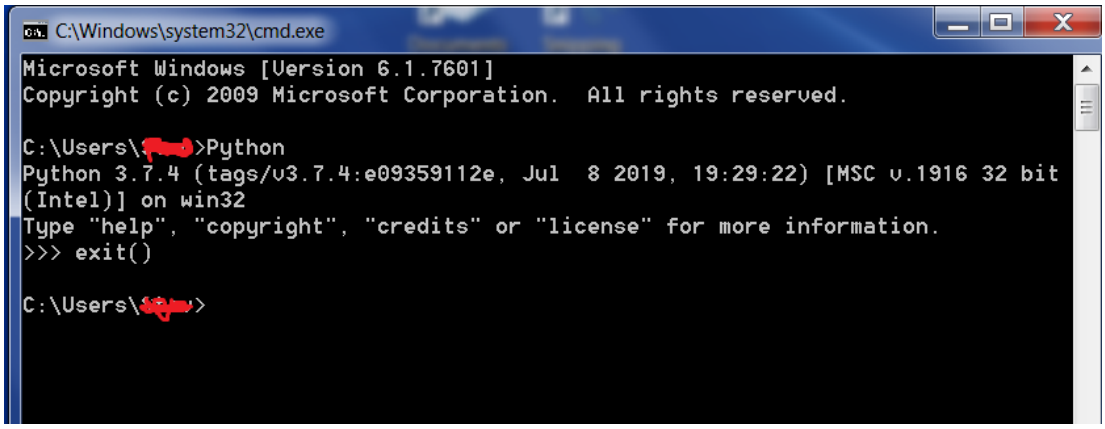
3.1. When your command prompt opens, type python and press enter. You should get something like:

If you did then you have python 3.10+ installed and connected to path.



You could write python from here but we won't.

3.2. Type exit() and press enter. You should get something like this:



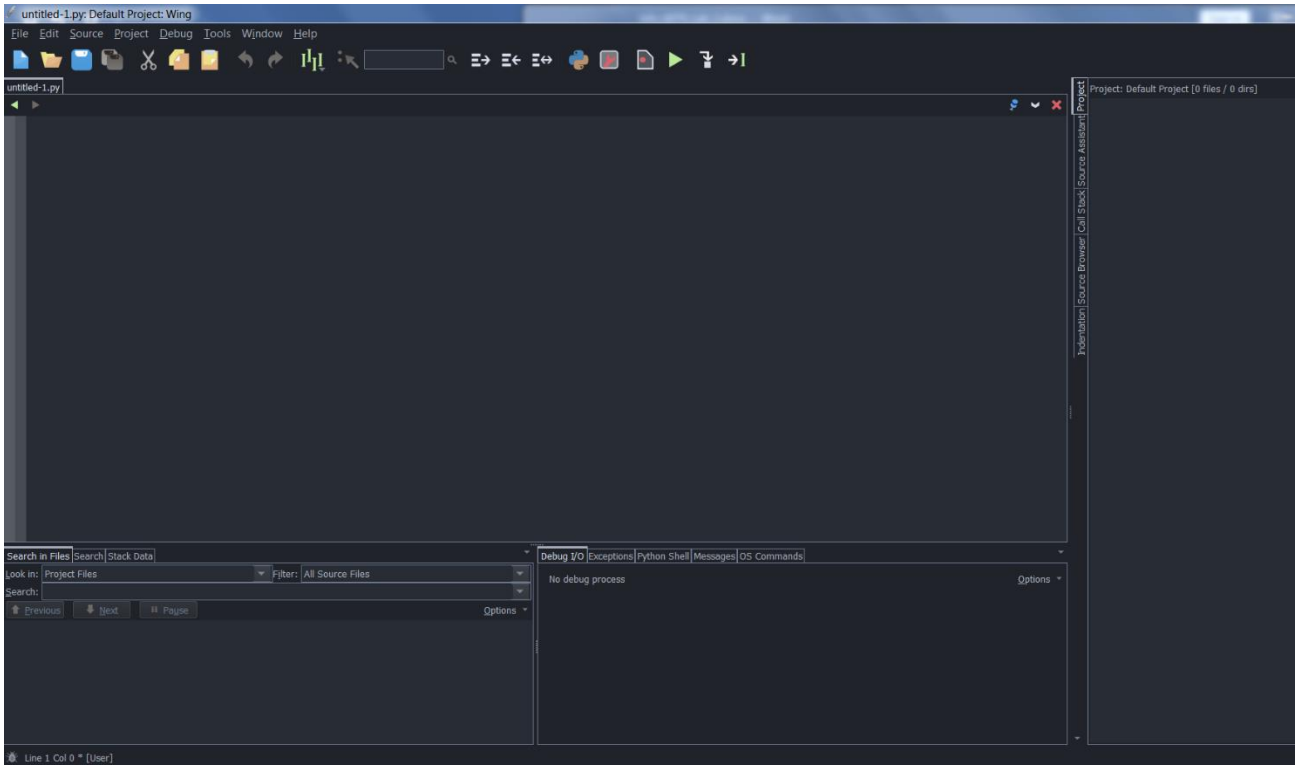
We are back to the command prompt. If you did not get errors you could run Python files from your prompt files.

4. Open your Wing IDE.
   4.1. Close the tip pop window.
   4.2. Press on file tap, at top left corner, then new, or you could press the blue page under the file.

This opens a new file. You should get something like the picture on the bottom. With a unsaved.py.

4.3. At this point, what is important for us?

4.3.1. In the upper left window, should be the biggest one, is where we will be coding. *Could see it in the above picture.



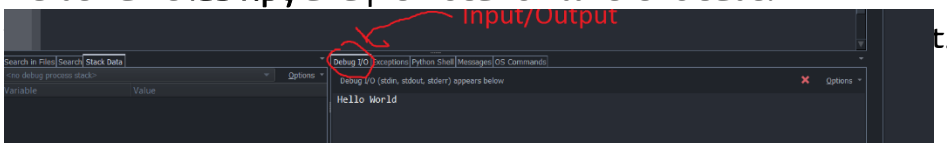
4.3.2. In the bottom right tab, you should have debug I/O option tab. This is where you will see the output of your code. It is also the place where you will be able to enter information to your code, when your code will be asking for information from a user. This is only when you are running your code in Wing IDE. *could see it in above picture.

4.4. Let's start coding!

4.4.1. Let's write

```
print("Hello Word")
```

4.4.2. Save the file by pressing "file" and then "save as".
Give name **lab1.py** and pick location where to save.

4.4.3. 

t. Output will be in the Debug

4.4.4. Congratulations, you just created and ran your first Python code.

FALL 2022

4.4.5. For this lab you could continue in the next line, by delete first line, commenting the first line, or create a new python file.

4.4.6. Create at least four (4) variables, have one string and three (3) numbers. Have one variable filled by user

- `strString = "London"`
- `lngNumber = 999`
- `dblNumber1 = 1.567`
- `dblNumber2 = 345.6789`

Print the variables

- `print("String is ", strString)`
- `print("Number1 is ", lngNumber)`
- `print("Number2 is ", dblNumber1)`
- `print("Number3 is ", strString2)`

4.4.7. If continuing on, commented, and deleted the first line skip to 4.4.9.

4.4.7.1. Save the file by pressing "file" and then "save as". Give name lab1_1.py and pick location where to save. This would leave the other Python file.

4.4.8. Then press the green arrow, ▶, about third bottom on the right. Output will be in the Debug I/O. Example is screen shot below.

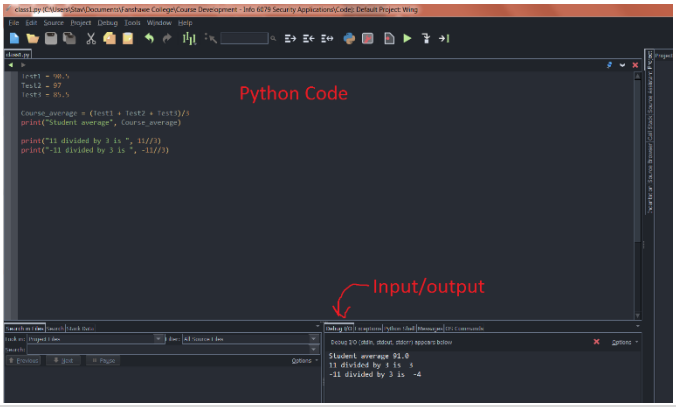
If code won't work you will get an error message.

The program will wait for you to enter information where you asked the user to enter information. If you had the fallowing:

```
Age = int(input('Enter your age: '))
Print ("your age is " , age)
```

The program would print "Enter your age: " and wait for information before print "your age is" followed by integer you entered.

4.4.9. Did you get what you expect? Try to find out why? Ask for help if you can not figure it out.



4.4.10. Now that we are able to set variables lets compare them. Use if statements. Compare integers, float, and strings

Ex:

4.4.10.1. What would happen if you compare a string with a number?

```
age_str ="twenty-two"
age = 22
if age_str == age:
        print("Compared")
else:
        print("Not Compared")
```

What was the result? Why?

4.4.10.2. Can you compare integer to float?

```
if 22 == 22.0:
  print("equal")
else:
  print("not equal")
```

4.4.11. Take a variable that you created and add to it by using +=

If a number variable add 1 ex: x += 1.

If string and some characters name += "aaa"

What do you think will happen?

What happens if you do name += 1

4.4.12. What happens if you add an integer or a float to a string?

Ex :

```
Name = "Bob"
Name += 1
```

Would you get Bob1 or something else?

If not, what do you need to do to get Bob1?

4.4.13. Let's using type command, can use if :

Ex:

```
x=3
if type(x) == int:
  x += 1
  print(x)
else:
  print("x is not an integer!")
```

What happened?

What would happen for x="three", or x=3.3?

4.4.14. Let's practice some math:

Run the fallowing:

Run the fallowing lines:

4.25 % 1 will result?

4 % 1.0 will result?

4 % 1 will result?

4.25 // 1 will result?

4.25 / 1 will result?