CSCI 1411: Fundamentals of Computing - Lab
Lab 2 – Using IDLE and Debugger
Due Date: September 8, 2023

Name: _____

Goals:

This lab will cover the following topics:

- Python IDLE
- Using IDLE Shell
- Using Files
- Customizing IDLE

Development Environment: Python IDLE

Deliverables:

1) This completed document (In either Word or pdf format),
2) Python file created in Part III and IV and
3) Screenshots of your outputs (included in Word document)

How to take a **screen shot**:

- For a Windows 10: Use Snipping Tool to copy and press CTRL + V to paste screen shot.
- For Mac: Press Shift + Command (⌘) + 4 to copy and press Command (⌘) + V to paste screen shot.

Creating and Running Python Program

We can create a Python program using any text editor like Notepad (on Windows) and TextEdit (on Mac). We can run a Python program using any Command Line Interpreter like Power Shell (on Windows) and Terminal (on Mac). In CSCI 1410 and CSCI 1411 we will be using Integrated Development and Learning Environment or IDLE. IDLE is installed when we install Python. IDLE is a very simple tool which includes an interactive interpreter or shell and a very basic editor. Its simplicity makes it a perfect tool for a beginning programmer. We can use the shell to experiment with simple Python statements. We can use the shell to write and run all our Python programs, but it does not save our Python program. This means that we will have to retype our Python program in the shell when we want to run our Python program. The good news is that IDLE also include a very simple file editor which can be used to write our Python program and save it in a file. You can watch the following video to learn about installing and using IDLE:

Mac User: https://ucdenver.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=e0b705b7-e9f8-432e-8219-b05a01340bf6

Windows User: https://ucdenver.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=7f40ef51-c780-4b9c-871e-b05a0157d407

You will have to login to Panopto web site using your CU Denver username and password to watch the video.

Part I – Using IDLE Shell (5 points)

Start the Python IDLE shell and run the following Python statements in it. Make sure that you press `Enter` key after each statement. Write the result in the box next to the statement. If there is an error then write the error message in the box next to the statement:

Number	Python Statement	Result or Error Message
1	`2 + 3`	
2	`17 * 6`	
3	`17 / 3`	
4	`17 // 3`	
5	`12 13 +`	
6	`Print ('Python is fun')`	
7	`Print ('Hello world)`	
8	`x = 45`	
9	`x`	
10	`9 + x`	
11	`y = 10`	
12	`x + y`	
13	`print (x)`	
14	`print (x + y)`	
15	`Print ('My name is','yourname')`	

Output for statement 8 and 11 will be blank. You can leave the boxes in front of those statements blank. Make sure that you replace `yourname` in statement 15 with your first name.

Optional Questions:

What is the difference between statement 3 and 4?

How will you fix statement 7?

Part II - Using IDLE Shell to Write Python Program (5 points)

Type the following Python statements in your shell (**do not copy and paste your code as it will not work**):

```
def main():
   x = 15
   y = 19
   z = x + y
   print ('x', x)
   print ('y', y)
   print ('z', z)
```

Note that Python shell will automatically indent all the lines after the line 1 (`def main():`). Press `Enter` key twice after the last statement. We have just created our first Python program using IDLE shell. We can run it by typing its name (`main()`) in the Python shell. Run the Python program, take a screen shot of the result and paste it in the box below:

Part III – Using a File to Write Python Program (10 points)

Create a folder on desktop and name it as FirstnameLastName (replace FirstName with your first name and LastName with your last name.

Example: If your name is Joe Doe then you will name the folder JoeDoe.

You will save all your work in this folder.

Click on File menu and select New File. This will open the file editor window. Type the following code in your editor window (**do not copy and paste your code as it will not work**):

```
def main():
    print ('Hello World')
    print ('My name is', 'yourname')
    # replace yourname with your first name
    print ('Python is fun!')
    print ('********************')
    x = 4
    y = 15
    print ('x', x)
    print ('y', y)
    print ('sum of x and y is', x + y)
    print ('difference of x and y is', x - y)
    print ('x multiply by y is', x * y)
    print ('x divide by y is', x / y)
```

Note that Python editor will automatically indent all the lines after the line 1 (`def main():`). Click on File and select Save As. This will open the Save As dialog box. Browse around to select the folder that you created on the desktop and name the file using following convention:

lastNameLab2Part3.py

Note that you do not have to type py extension as it will be added automatically by the IDLE editor.

Now click on Run menu and select Run Module. This will load your program into the Python IDLE shell. If there are any error in your Python program then you will see a pop-up dialog box with error message. Make sure that you have typed the code as given above. Fix the error (if any) and reload the code into the Python IDLE shell. Now type the name of the program (`main()`) to run your program. Take a screen shot of your output and paste it in the following box:

Go back to your editor window, add a blank line at the end of the code and type: `main()`

in the next line. Make sure main starts in the first column. Now your code will look as follows:

```
def main():
    print ('Hello World')
    print ('My name is', 'yourname')
    # replace yourname with your first name
    print ('Python is fun!')
    print ('********************')
    x = 4
    y = 15
    print ('x', x)
    print ('y', y)
    print ('sum of x and y is', x + y)
    print ('difference of x and y is', x - y)
    print ('x multiply by y is', x * y)
    print ('x divide by y is', x / y)

main()
```

Save the file. Click on Run menu and select Run Module. This time your program will be loaded into the shell and the shell will run it automatically. You do not have to type program name in the shell to run it. Take a screen shot of your output and paste it in the following box:

Submit your Python file with your lab submission.

Part IV – Open an Existing Python File (10 points)

Download the file Lab2Part4.py from Canvas (it is located under Lab 2 module), save it in the folder that you created in part III of the lab. Click on File menu and select Open. This will open the File Open dialog box. Browse around to the select the folder that you created on the desktop and click on the file Lab2Part4.py to open it. Save the file (click on File menu and select Save As) using following naming convention:

lastNameLab2Part4.py

Click on Run menu and select Run Module to load your program into the Python IDLE shell and run it. Copy and paste your output in the following box:

Look at the output. Is the output correct? Fix the error and run it.

Copy and paste your output in the following box:

Submit your Python file with your lab submission.

Part V – Customize Python IDLE (Optional)

There are many ways to customize Python IDLE. To access the customization window, click on:

- Options menu and select Configure IDLE (on Windows)
- IDLE menu and select Settings (on Mac)

You can customize the following items:

- Fonts – you can change the font face and size.
- Highlights – you can change the how keyword are color coded in Python editor. You can also change the theme to one of the following:
    - IDLE Classic
    - IDLE Night
    - IDLE New
- Keys – You can customize keyboard shortcuts. You can also create your own shortcuts.

Please see the following web page for more details:

https://realpython.com/python-idle/#how-to-customize-python-idle

You can also turn on line numbering in editor by clicking on Options and selecting Show Line Numbers.

Rubric for Lab 2:

Criteria	Rating
Part I	Number of correct answers: 13 to 15 – 5 points Number of correct answers: 10 to 12 – 4 points Number of correct answers: 7 to 9 – 3 points Number of correct answers: 4 to 6 – 2 points Number of correct answers 1 to 3 – 1 point Number of correct answers: 0 – 0 points
Part II	Screenshot included – 5 points No screenshot included – 0 points
Part III	Include two screenshots and submitted Python file – 10 points Include two screenshots but Python file is missing – 5 points No screenshot or Python file included – 0 points
Part IV	Included two screenshots and Python file with corrected code – 10 points Included two screenshots but did not correct the Python code – 2 points No screenshot or corrected Python code – 0 points