## ITSE 1359 – Lab 1 Assignment (Refer to Intro, Ch. 1 & 2 as needed):

NOTE: Lab 1 requires you to work on and submit two python programs (python_facts.py and hello_world.py). Read this document carefully and comply with submission and grading criteria. Use video provided on last page to do your Python Facts program. As for the Hello World! program, follow the specifications in the problem definition.

### Problem 1 (Python Facts - python_facts.py):

Problem Definition: Using an editor of your choice (Sublime recommended by author of your book), create a Python program that prints a student's name and a couple of facts about Python. Use the facts provided (see output below).

Coding: The following coding concepts must be exhibited: string variables, changing case, and formatted strings. You must also use escape codes for tabbing and newline. VIP: Name your program: python_facts.py. Do not forget general and specific comments.

Screenshot of Output: Your program output should match this:

```
My name is Fred Flinstone.

Things I have learned about Python:

    Python is a general purpose programming language.
    Python was created by Guido van Rossum and released in 1991.

[Finished in 0.2s]
```

Detail Specifications: Instead of "fred flintstone" use *your name*. The student's name must be constructed using first and last name variables. The formatted string should utilize the .title() method to change the case of the first and last names.

A blank line can be effected by using the print() function with no argument. You can also use the "\n" escape code to create a blank line.

To contrast procedure above, assign the full name of Python's creator to a string variable using title case in the string. Make sure you access this variable when you print out things learned.

**Specific Comments:**

# declare string variable for student's first name & initialize (lower case)
# declare string variable for student's last name & initialize (lower case)
# declare string variable for python's creator's full name & initialize (title)

# create formatted string of student's full name & assign to string variable
# pass string variable to print function
# print a blank line using print() or \n

# print things learned about python using tab and newline escape codes:
# print string literal saying python is general purpose
# use f string to show Pythons creator

**Code Examples:**

How to create a string variable?

```
# declare string variable for python's creator's full name & initialize
python_creator = "Guido van Rossum"
```

How to create an f string?

```
# create formatted string of student's full name & assign to string variable
message = f"My name is {first_name.title()} {last_name.title()}.\n"
```

How to use tab and new line escape codes?

```
# use tab and newline escape codes to list things learned
print('\n\tPython is a general purpose programming language.')
print(f"\tPython was created by {python_creator} and released in 1991.")
print()
```

## Problem 2 (Hello, World!):

Problem Definition: Create program that shows "Hello, World!" on screen five times cascading vertically to the right and then gives credit to person who first wrote it and the year it was written. Program then prints “Hello, World!” cascading vertically to left. See output screen shot on next page.

Coding Specifications:

Use string variables to hold "hello, world!" and name of computer scientist who wrote it, both of which should be lower case. Put the year the program was created in a numeric variable. Use these variables in a formatted string to state name of person who wrote this program and the year.

All string variables should be initialized in lower case. When they are printed, use the title() method to provide title case.

When printing out “Hello, World!”, use the print statement with tab escape codes to provide for tabbing to right and then to left. Make sure there is a blank line before and after the credit line.

The formatted string must use placeholders for the program, computer scientist, and year created. You must show “Hello, World!” in double quotes. For this you may use an escape code ( \” ) or single quote for outer string and double quotes for internal quote.

Specific Comments / Pseudocode:

```
# initialize string variable to hold “hello, world!”
# print the variable five times cascading vertically to right – use title case
# print blank line

# initialize string variable to hold name of person who created program
# initialize numeric variable to hold year program was created
# Use formatted string to print sentence giving credit to that person

# print blank line
# print the variable five times cascading vertically to left – use title case
```

Output (Your program output must match this):

```
Hello, World!
    Hello, World!
        Hello, World!
            Hello, World!
                Hello, World!

Brian Kernighan wrote the first "Hello, World!" program in 1978.

                Hello, World!
            Hello, World!
        Hello, World!
    Hello, World!
Hello, World!

[Finished in 0.1s]
```

Coding Examples:

How to your print hello world cascading?

- You can use a formatted string:

```
program = "hello, world!"
print(program.title())
print(f"\t{program.title()}")
```

- You can use concatenation ( + ):

```
program = "hello, world!"
print(program.title())
print("\t" + program.title())
```

NOTES: Concatenation and \ are not covered in our book. Concatenation is covered in previous edition of book. The \ is used when you have a long line of code.

How do you create the formatted string for the credit line?

```
computer_scientist = "brian kernighan"
year_created = 1978

print(f'{computer_scientist.title()} wrote the first "{program.title()}" \
program in {year_created}.')
```

More coding examples:

```
# cascase output to left
print()
print("\t\t\t\t" + program.title())
print("\t\t\t" + program.title())
print("\t\t" + program.title())
print("\t" + program.title())
print(program.title())
print()
```

**Where to create your variables?**

You are free to create all your variables at top of a block of code or as you use them. Depending on how you were taught and what language you are using you may find variations. I do find that Java programmers like to declare their variables as they are using them. In that case, you may see variable declaration mixed throughout the program. It may be a matter of style and preference.

**Comments in your Programs:**

You should include general and specific comments in each program that you submit. General comments should appear at the top and include program name, programmer's name, date, and description.

Specific comments should appear along with your code and briefly document the logic of your program. You can use the comments provided in this lab assignment or condense them.

Again, you must have general comments. Points off for non-compliance.

Example of General Comments:

python_facts.py

```
1  # Program:      python_facts.py
2  # Programmer:   student name
3  # Date:         todays date
4  # Description:  lab 1
5  ##############################################################################
```

## Condense Comments?

The specific comments I provide in the Python Facts program are excessive. I provided them to document how the program should work – kind of like pseudocode. So, you are free to condense these specific comments. Consider:

```
# declare variables and initialize

# create f string of student full and pass to print function

# print things learned using tab and new line escape codes
```

Again, the specific comments I provide in the Python Hello World program are excessive. I provide them for beginners who need more help. You are free to condense these specific comments so long as you accomplish the objectives of the problem and match output screen. Consider:

```
# declare and initialize variables

# cascade output to right

# show person who created first hello world

# cascade output to left
```

In summary, feel free to condense my specific comments to something that represents the flow of your program.

## Submit your lab assignment:

Using the Canvas assignment tool, upload your completed work (2 files) to the lab 1 assignment. Attach the first file (python_facts.py) and then the second (hello_world.py) and then submit.

All lab assignments must be submitted using the CANVAS assignments tool. Lab Assignments will not be accepted any other way**.** Make sure you submit your work to the right lab assignment number otherwise you will not get credit.

## Grading Criteria:

- ✓ You must use the provided video to help you do Python Facts.
- ✓ Don't forget general comments for each program.
- ✓ Specific comments are optional but highly recommended.
- ✓ Use white space to make your code easy to read.
- ✓ Comply with PEP-8 conventions for variable names, file names, etc.
- ✓ Satisfy the problem definition and other grading standards.
- ✓ Your work should not have syntax errors.
- ✓ Your work must be your own.
- ✓ Match your output screen to screenshot provided.
- ✓ Any deviations from lab specifications will result in points off.
- ✓ If you want to vary, do so on your own.
- ✓ Do not use coding constructs not in pdf or chapters we have covered.

## VIP VIDEOS:

How to do Python Facts (step-by-step): Lab 1 python_facts.py

Lectures and other videos: VIEW MY PYTHON PLAYLIST

## Closing:

If you have questions about this lab send me a message using canvas inbox or attend zoom office hours. See Unit 0 for link for my office hours.

By-the-way, a great way to get ready for your lab assignments (and exams) is do the Try It Yourself problems in your book. Most of the solutions are on the authors website and I also discuss them in my YouTube video lectures. Another way to get ready for your lab assignments is to review my lecture notes for each chapter. In Unit 0, find the link for my chapter notes.

**Warning:** Labs will increase in complexity with each lab assignment. A beginner may need 5 to 15 hours to complete each lab. Please start early because there will be no extensions.

Have an exception free day! Prof. Benavides