CMPSCI 119 LAB #1 – Bar Graphs Professor William T. Verts

The goal of this first Python programming assignment is to successfully create, enter, debug, and run a simple program in the JES environment. If you have not done so already, you must first install the JES environment on your own personal computer. Please set up a special folder on your desktop or in your documents folder to hold all your Python projects. I also suggest that you buy a flash drive (SanDisk recommended) and put a folder for your Python projects there as well; the flash drive can then be used for backup and for bringing code in for us to help you debug.

Background

In this assignment you are going to write a program to enter a set of grades for students who have taken an exam, and then generate a set of text-based bar graphs based on those grades. For example, if student Fred received a 78 on his exam, the line of output for Fred will be 78 copies of the | character (the vertical bar), followed by a space, his name, another space, and his score. When your program runs, the expected output from JES will look like the following image:

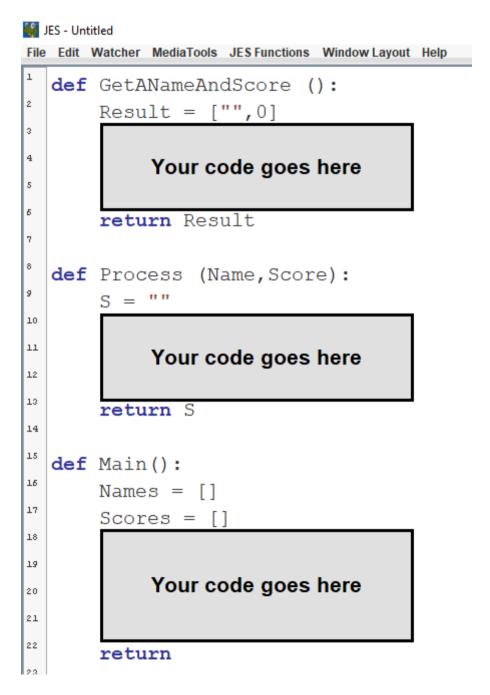
```
Main()
Enter Name ---- Fred
Enter Score --- 78
Enter Name ---- Sam
Enter Score --- 34
Enter Name ---- Mary
Enter Score --- 97
Enter Name ---- Carol
Enter Score --- 69
Enter Name ---- Joe
Enter Score --- 82
Enter Name ---- Bob
Enter Score --- 46
Enter Name ---- Sue
Enter Score --- 89
                    ||||||| Carol 69
                  ||||||||||| Joe 82
```

Setting up the Assignment

In the JES environment, write a comment at the <u>top</u> of your program containing *your own name* and the lab number. For example, I would put in a comment as the first line of text:

William T. Verts - Lab #1

After that line, type in the following program code framework just as you see it here, and then save it in your Python folder with Lab1.py as the filename. For now, leave the gray areas blank; you will write your own code there later.



If you load this program and try to run it by typing **Main()** at the Python >>> command line prompt, it should do nothing, but neither should it contain any errors. Try it.

You will notice that there are three functions which need to be finished, GetANameAndScore, Process, and Main. Functions GetANameAndScore and Process are independent of each other; you can write and debug each of them, in either order, before working on Main.

Task #1 - GetANameAndScore

The GetANameAndScore function asks the user to enter a name and a score, and then returns a list containing those two items as variable Result (which is preinitialized to a list containing an empty string and zero). To get a string for the name from the user, use the raw_input function; to get a number for the score from the user, use the input function. However, do NOT ask for a score if the user enters an empty string for the name. See the screenshot on the first page of this assignment to determine how the inputs should look.

Test your function by calling it from the interactive area of JES. That is, type the command GetANameAndScore() at the >>> command line prompt and verify that the returned result is either a name and a score, depending on what you enter (such as ["Fred", 78] or ["Sam", 34] for example), or an empty string and zero (the exact list ["", 0]). It should <u>never</u> be possible for this function to return a list with an empty name string and a non-zero score number.

Task #2 - Process

The Process function has two parameters, Name and Score. Those parameters will be passed the name and the score for a particular student. For example, if the function is called as Process ("Fred", 78) then Name will contain the string "Fred" and Score will contain the integer 78.

The Process function initializes a string S to the empty string, and then builds up and returns S as the value of the function, where S will contain the correct number of vertical bar characters, a space, the student's name, a space, and the student's score.

Finish the function by writing code in the gray area to do this.

This function requires the use of a loop to build up the string; for this assignment I want you to use a while-loop (not a for-loop). The Process function <u>must not</u> contain any print statements. It returns its result as the value of the function; printing happens elsewhere.

Test your function by calling Process from the >>> command line prompt as Process ("Fred", 78), which would return as the value of S the following string:

Similarly, manually test Process("Sam", 34) and Process("Mary", 97) to verify that your function works correctly.

Task #3 - Main

Now we need to complete the main program, which is in two parts.

The first part of Main is the interactive portion, which asks the user for names and scores by calling the GetANameAndScore function many times in a loop. Each time GetANameAndScore returns a list containing a non-blank name (regardless of the score), the name is added to the end of the Names list and the score is added to the end of the Scores list. Only when the user enters a blank name does this process stop, and the program will then proceed on to the second part of Main.

The second part of Main, once the Names and Scores lists have been constructed, is to call the Process function for every set of associated values from those two lists, and in each case print out the result returned from Process. If you've done everything correctly, Names and Scores will contain the same number of elements, but your program must not make any assumptions about what that number is (i.e., you <u>cannot</u> assume a constant for the length of the lists — your program MUST be adaptable to any number of students).

This function also requires the use of a loop; for this assignment I want you to again use a while-loop (not a for-loop).

Notice that "Fred" is the first name to be entered, and 78 is the first score to be entered. These values will be the first elements of Names and Scores, respectively. These will be the first values passed to the Process function. The result of that call is printed. Then, in the second call to Process, your program must use the second values from Names and Scores ("Sam" and 34), in the third call to Process it uses the third values from Names and Scores ("Mary" and 97), and so on for as many items as there are in the lists.

Running the Program

Run the program by typing Main () at the command prompt.

You should be asked for names and scores. Use the following values: Fred=78, Sam=34, Mary=97, Carol=69, Joe=82, Bob-46, and Sue=89. After Sue's name and score have been entered, enter a blank string to terminate the data entry process. You should then get a histogram as shown in the screenshot on the first page.

If the output does not contain the correct information, debug your program and run it again. Continue to edit and test your program until the resulting output <u>does</u> contain the correct information.

Finishing Up

When you are finished and everything runs correctly, go to the class site and click on the link for submitting lab assignments. In JES select all the text, copy it to the clipboard, in the Web page paste the text into the program area of the submission form, fill in your name and ID number and the lab number in the appropriate slots, and then submit the assignment for grading.

The graders will score your program by running it to see if it correctly generates the appropriate graphs. You will also be graded on efficiency and completeness, and the graders will be explicitly looking to see if your program will work for an arbitrary number of students (number of entries in Names and Scores), that it does not ask for a score if the name is blank, and so on.