Computer Science 1 — CSci 1100
Lab 3 — Strings and Functions
Fall 2014

Lab Overview

In this lab, you will write a series of short Python programs to manipulate strings, read input, and output greetings. Start by making a folder for Lab 3 in your Dropbox where you keep your Computer Science 1 material. Then start working on the following four checkpoints. This is one of the few — perhaps only — labs of the semester that has four.

Checkpoint 1: Framing Spam

Write a short (three line) Python program that prints

```
**********
** spam **
**********
```

In doing so, make sure you use

```
print '*' * 10
```

rather than

```
print '**********'
```

This will come in handy when you modify your code in later checkpoints. Save the program in a file called `check1.py`. Show the TA or a mentor both the program and the result of running it. Congratulations, you are done with Checkpoint 1.

Checkpoint 2: Framing Four-Letter Input and Functions

Copy your program from Checkpoint 1 into a new program, `check2.py`, and open it in the WingIDE. There are two parts to this checkpoint:

- Modify `check2.py` so the output that frames the word is performed in a function called `framed`. This function should accept a single string as an argument. Assume, for now, that this string has four characters, just as `spam` does. Add code to call your function and test it, e.g.

  ```
  framed("spam")
  ```

  or

  ```
  framed("eggs")
  ```

- Add code to use the `raw_input` function call discussed at the end of Lecture 5 to read a four letter word into a string. Then, pass this string to the function `framed` you just wrote. The output when you run your program should look like

  ```
  Enter a four letter word: eggs
  **********
  ** eggs **
  **********
  ```

When you have this working, show it to the TA or a mentor. Make sure your function follows the program structure we discussed and required in the homeworks. Congratulations, you have completed Checkpoint 2.

Checkpoint 3: Framing Any Word

Be sure you save `check2.py` and make a copy of it called `check3.py`. You will modify this for Checkpoint 3.

If the user types a word that is either longer or shorter than four letters, your output will look a bit funny. For example,

```
Enter a four letter word: inquisition
**********
** inquisition **
**********
```

Hence, in this checkpoint, you must modify your code to ask for a single word of any length and then frame it properly. To do so, you need to use the string `len` function to help you decide how many `'*'` to output. The result of running your program should look like

```
Enter a word: inquisition
****************
** inquisition **
****************
```

When you have this working, show it to a TA or mentor. Congratulations, you have completed Checkpoint 3.

Checkpoint 4: Framed Greeting

In this last checkpoint, you will write a new program that outputs a framed greeting for a person who enters a first and a last name. An example of running this is

```
Please enter your first name: John
Please enter your last name: Cleese

************
** Hello,  **
** John    **
** Cleese! **
************
```

This will take a bit more work than the previous checkpoints. You will need to ask the user for a first name and read it; you will need to ask the user for a last name and read it; you will then have to pass the two names to a function that outputs the framed greeting. In this function you will need to calculate the maximum of the lengths of `"Hello,"`, the first name, and the last name (with the `!` added to it). This will help you determine how many `'*'` to output for the first and last lines. Go ahead and write and test this much. In doing so, your output might look like

```
Please enter your first name: John
Please enter your last name: Cleese

*************
** Hello, **
** John **
** Cleese! **
*************
```

Finally, you need to figure out how to get the appropriate number of spaces between the end of each word and the `'*'`. This is the difference between the length of the string and the maximum of the lengths of three strings. See if you can figure it out.

Test your program with different sets of first and last names, including using three different cases: where both names are shorter than `Hello,`, where the first name is the longest string, and where the last name is longest.

When you have your program fully working, save your code, and then show the result to a mentor or a TA. Congratulations, you have finished Checkpoint 4 and all of Lab 3.

Incremental Development and Testing

At several points during the lab, we asked you to write and test code that only completed part of the requirements for a checkpoint. You should adopt this approach of "incremental" development and testing in all of your programming. Do not expect to be able to write a complete working program from scratch without testing parts of it first.

Extra Challenge

If you have the time and interest, modify your Checkpoint 4 code so that the greeting is centered, e.g.

```
**************
**  Hello,  **
** Jonathan **
**  Smith!  **
**************
```

While we offer no extra credit for this, it is a good challenge problem.