Lab 1
Mad Libs

29 August 2019

Most weeks there will be a short "drill" assignment leading up to the actual lab. I will expect you to work on it before lab, but it's okay if you get stuck on something—that means that when you get to the lab you'll be ready to ask a question on how to proceed. (If you do finish the drill, that's cool too, of course.)

The drill for this lab is below. Come to lab on Tuesday either with it completed or with a specific written question in your notebook identifying which drill step you got to and what about it you're stuck on.

1. First, open up the repl.it assignment "Lab 1 drill". In the pane on the left, write an instruction to print a *prompt* that tells the user to "Please type your first name: ". See §2.3.1 in the book for a model on how this might work.

2. Before continuing to the next step, try running the program. Fix any errors. In fact, *always* do that before moving on to the next step. At this point, the program should just prints the prompt and immediately stop—because that's all you've told it to do.

3. After the prompt, write two lines to define a variable to hold the name, and to actually read it in from the user. (Remember to test this before going on, by clicking "run". It shouldn't give you any error messages or warnings! Once it's running, it will expect you to type something in the dark pane in the lower left of the window.)

4. Print a line that says hello to the user, like the start of an email or letter, including their name. For instance, if the name the user typed was Lia, your program should at this point produce

   ```
   Hello Lia,
   ```

5. Add another prompt "Please type your age: ", and another variable to contain the response (a number). This second prompt and input should come after the first prompt and input, but *before* printing the greeting. That is, you're inserting extra lines into the middle of your program.

6. At the bottom, after you print the greeting, write a message matching the following format, wishing them a happy birthday:

   ```
   I hear you turn 4 today.
   Happy birthday!
   ```

   The age printed should be the age they typed in.

7. Confirm that the final order of the pieces is: prompt and input the first name; then prompt and input the age; then print the greeting; then print the rest of the message.

   If you've got all that, go ahead and run against my tests and submit!

Lab 0, take two

Now that I've resolved the password situation, pull out the Lab 0 handout from yesterday and spend a little time on that. Particularly if you're waiting for me to get to you to answer questions about the Lab 1 drill!

After the drill: Mad Libs

After you've completed the steps listed in this week's drill, I have a few more things I want you to do for this lab.

A "Mad Lib" is a story constructed by prompting someone for words in certain categories (adverbs, proper names, numbers, articles of clothing) and using them to fill in blanks in a template.[1] You now have the tools you need to write a program that does exactly that—it will be a lot like the drill, but you get to choose what direction it takes.

Documentation

One program development habit I want you to start building right away is to always include at least minimal documentation with anything you submit. Once you start handing in programs on the department server, I'll have

[1] I didn't invent this idea or the name. Google it.

more specifics on what that looks like. But even for the repl.it assignments, I want to build the habit. So:

In *every* repl.it assignment from now on—it's ok if Lab 1 Drill doesn't have it but the full Lab 1 should and all future stuff—there should be *comments* at the top of the program that say:

- Your name
- The date
- A brief description of what the program does
- Additional instructions on how to work with it (if any are needed)

Start the Lab 1 assignment on repl.it, and at the top of the file (where I've put a comment to "insert documentation here"), type the following:

```
// Author: Your Name
// Date: 29 August 2019
// This program composes a silly story based on user input.
```

You should replace the name and may reformat the date if you like, but otherwise it should look pretty much just like that.

Testing a program

If you've been compiling and running and testing your code as often as I (and the book) have encouraged you to do, you're probably *already* getting a little sick of typing the same sample input over and over again. (If not, you're probably not running and testing enough!)

The normal response of a computer scientist faced with something annoyingly repetitive is to figure out how to automate it, so that's what we'll work on now. As before, there will be a more detailed set of things to do once we're working more on the department server (in particular, you'll be able to mimic the full input-output testing I automated for Hello World and for the Lab 1 drill), but for now:

repl.it lets you set the input that will be send to the "standard input" (that is, `cin`) whenever your program is run. In the running pane (the lower left), click the icon of an arrow pointing into a box. In the resulting popup,

you can type in whatever you want your standard test to be—you won't see the program's prompting here (because the program's not running yet!) but everything you *would* type at the keyboard goes in this box. Click the button to set the input, and from then on, every time you run the code, the system will automate typing in exactly that stuff into the program. You'll still have to verify with your eyes that the result is correct, but it'll save typing!

Unfortunately, repl.it doesn't appear save these input between sessions. (If someone finds a trick to do so, let me know!) Still, this is a good introduction to the idea of test automation, and you should definitely use this as you start working on your Mad Libs program.

The actual instructions

Your program should prompt for at least six things. At least one of the things you have to use twice (or more) in the story; and at least one of the things has to be a number (which you should store as a number—try doing some math on it if you feel comfortable doing so!) in the story. Your story doesn't have to be particularly long, and will probably be fairly surreal. That's ok! Have fun with it.

Rubric

When I write a lab assignment, I usually have a plan for how I will distribute points, and in the interest of transparency, I will often tell you (most of) that plan in advance. The rubric is subject to change, but may give you some ideas of where to focus your attention, and you should generally be able to have a pretty clear idea of what your score will be even before I evaluate it.

RUBRIC

1 Attendance at lab with drill done OR question written down

Drill

1 Drill compiles, runs, and does anything
1 Prints prompt(s)
1 Reads and writes values
1 Letter is correctly written, after all input is done

Mad Lib

1 Documentation at top of file
1 Compiles, runs, reads/writes
1 Read and used ≥ 6 inputs
1 Used an input twice in story
1 Used number variable correctly

Handing in

If you don't finish all of that during the lab period (especially if you had technical difficulties during the drill, or are being very creative with the second program), that's fine. This lab will be due **Wednesday the 4th at 4pm**. (In general, final work for a lab will be due the following Wednesday at 4pm.)

Submit the program through repl.it as before.