

Lab 1

Mad Libs

23 January 2025

Most weeks there will be a short “preview” 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 preview, that’s cool too, of course.)

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

1. First, open up the codeboard assignment “160 Lab 1 preview”. Click `main.cpp`, and in the pane on the right, put your name and the assignment into that first line after the double slashes, and then update the next few lines to match what we’ve done before. Then inside the curly brackets, write an instruction to print a *prompt* that tells the user to “Please type your name and age:”. See §2.3 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 print the prompt and immediately stop—because that’s all you’ve told it to do.
3. In fact, go ahead and submit it now. It’ll fail the tests but that’s ok.
4. 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 input field at the bottom of the window.)
5. 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,
```

6. Update the program to define a second input (this time for a number) and to also read that before printing the greeting. That is, you're inserting extra lines into the middle of your program.
7. 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.

8. Confirm that the final order of the pieces is: prompt; then input the name and 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 again!

After the preview: Mad Libs

After you've completed the steps listed in this week's preview, 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 preview, 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 more specifics on what that looks like. But even for the codeboard assignments, I want to build the habit. So:

In *every* codeboard assignment from now on—it's ok if Lab 1 Preview doesn't have it but the full Lab 1 should and all future stuff should—there

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

should be a line (or lines) at the top of the program that (each) start with two slashes and say:

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

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

```
// Author: Your Name
// Date: 23 January 2025
// 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

As the programs start to get longer than a line or two, it's a good idea to compile and run well before you think you're done, after every one or two instructions. (As in step 2 of the preview work!) When you do think you're done, you should test the whole thing with at least a couple different inputs, to make sure it works *in general* and not just with one particular input.

If you get tired of retyping all that, it's not a bad idea to keep a text file open, off to the side, that you can type a standard set of inputs into and then copy-paste from there.

AI policy and frequent submission

Some use of generative AI is fine, but you a) should not paste this assignment or type it verbatim into the AI prompt, and b) should not be asking the AI for the whole program all at once. (Just like you can ask for help from a human, but should not have them write the whole program for you!) If

you get help from an AI chat OR from a person, you should note that in a program comment near whatever you got from them.

Relatedly, I expect that you'll click the Submit button relatively often, and I *require* that you do so at least 2–3 times over the course of working on the lab. (Note that codeboard only lets you submit if the program is successfully compiling, so that's another reason to want to try to compile and run after every few lines of code!) As a rule of thumb, hand it in after completing each 1–2 points on the rubric. That gives me a version history and progress report on your work. Submissions that jump straight to a final, (near-)correct version with no intervening submissions along the way *will receive little or no credit* for that part. ← scoring note!

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; at least one of the things has to be a number, which is stored as a number and which you do actual (simple) math on (adding 1 or multiplying by 2 is plenty); and at least some aspect of the output has to be conditioned on the value of an input using `if` (and maybe also `else`). 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 preview done OR question written down

Part 1

1 Part 1 compiles, runs, and does anything

1 Prints prompt(s), 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

1 Some output is conditional (using `if` and maybe `else`)

See also the scoring note above under “AI policy and frequent submission”.

Handing in

If you don’t finish all of that during the lab period (especially if you had technical difficulties during the preview, or are being very creative with the second program), that’s fine. This lab will be due **Wednesday the 29th at 4pm**. (In general, final work for a lab will be due the following Wednesday at 4pm.)

Submit the program through codeboard as before (and, as mentioned, frequently).

This document was written and prepared without the use of generative AI.