CS 112 - Week 2 Lab Exercise - 2022-09-02

Deadline

Due by the end of lab on 2022-09-02.

How to submit

Submit your `.cpp` and `.h` files for the problems below on https://canvas.humboldt.edu.

IF you prefer, you may instead compress your `.cpp` and `.h` files to be submitted into a single `.zip` file and submit that `.zip` file to Canvas.

(I'll also accept the `.zip` file created when one downloads a folder from the CS50 IDE, as long as it includes all of your lab's `.cpp` and `.h` files -- I suspect it will also contain your resulting executables, but that's OK.)

Purpose

- To practice more with C++ programs featuring separate compilation.
- To practice at least one C++ repetition statement.

Important notes

- Be sure to put your name and today's date in each of the files for the new functions `five_letter_word` and `ask_for_word`.

Problem 1

Provided with this lab exercise is a function `is_letter`, which expects a character (of type `char`) and returns `true` if it is a letter character, which we are defining as one of

`{'a', 'b', ..., 'z', 'A', 'B', ... 'Z'}`

Use the design recipe to design a function **`five_letter_word`** that expects a string, and returns whether it is a five-letter word -- that is, whether it contains exactly 5 characters, and all of them are letters. For full credit, it needs to make appropriate use of function `is_letter`.

(BEWARE -- this is NOT exactly the same as `five_letter_str` from Homework 1, Problem 4! That problem only cares if the string contains exactly 5 characters, no matter what those characters are -- this lab exercise function wants it to contain exactly 5 *letters*.)

Submit the files `is_letter.h`, `is_letter.cpp`, `five_letter_word.cpp`, `five_letter_word.h`, and `five_letter_word_test.cpp`.

(Note: I think there are several different reasonable ways to approach this!)

Problem 2

REMINDER:

You can request a string from a user -- and be sure it contains no blanks, tabs, or newline characters -- using something like this:

```
string user_reply;
cout << "Enter a string: ";
cin >> user_reply;
// user_reply now contains a string containing at least 1 character
```

It would be convenient to have a function that asks the user to enter a string, and it makes sure it contains exactly 5 letters -- and if not, it keeps making them enter such a string until they do!

Using the design recipe, design a function **`ask_for_word`** that expects a user's name, has the side-effect of asking them by name to enter a 5-letter word and reading in what they enter until they successfully do so, and returns the resulting word. For full credit, it needs to make appropriate use of function `five_letter_word` from Problem 1.

This one is unusual to write tests for, so in this case I am providing the tests. In `ask_for_word.cpp`, in its `tests:` section, put:

```
  tests:
      if the user Jenna enters
1
moo
12345
moo23
sandstone
moons
      when prompted, then:
      ask_for_word("Jenna") == "moons"

    if the user Carlos enters
Slick
      when prompted, then:
      ask_for_word("Carlos") == "Slick"
```

(You can add additional tests if you would like.)

And, in `ask_for_word_test.cpp`, since you need to let the user know what to do, put:

```
    cout << "*** Testing ask_for_word ***" << endl;
    cout << "Should see prompts to Jenna." << endl;
    cout << "Enter 1, moo, 12345, moo23, sandstone, and moons when prompted."
         << endl;
    cout << "Should see true at the end if test passed:" << endl;
    cout << (ask_for_word("Jenna") == "moons") << endl;

    cout << endl;
```

```
cout << "Should see prompts to Carlos." << endl;
cout << "Enter Slick when prompted." << endl;
cout << "Should see true at the end if test passed:" << endl;
cout << (ask_for_word("Carlos") == "Slick") << endl;
```

(You can add additional tests if you would like, and you can also just try out this function in additional ways if you would like!)

Hint: here is pseudocode for the body of this function:

ask given-name for a string of 5 letters

read in what user enters

while entered string is NOT a 5-letter string

    complain to given-name, tell them it needs to be exactly 5 letters, and ask them to try again

    read in what user enters

return the entered string

Submit the files `is_letter.h`, `is_letter.cpp`, `five_letter_word.cpp`, `five_letter_word.h`, `ask_for_word.cpp`, `ask_for_word.h`, and `ask_for_word_test.cpp`.