Lab 1 - C++ Review

If you attend lab but don’t fully complete the work, you can still earn credit. However, you are required to finish your work prior to the next scheduled lab. For extenuating circumstance, contact your lab TAs and the course instructor.

You will receive 7 points for attending lab and the remaining 3 points after showing your working solution to a TA or the instructor. The reason for this approach is so that we can have a chance to review your work and help you to understand any mistakes (rather than repeating these mistakes on future homework).

Pair Programming: In this lab you can work in pairs (if desired). In this case each student should submit a copy of their work to Canvas (although the submitted work will be identical and the names of both students will be included on the document). Be sure that both of you are contributing to the work.

When you design functions in C++ there are 3 ways in which you can pass parameters: pass by value, pass by reference, and pass by pointer. If you would like to review these topics, I recommend reading chapter 9.5 and 9.10 on the LearnCPP website.

In order to practice pass-by-reference and pass-by-pointer, you will write a short program that:

1. Asks the user to enter a string
2. Accepts the user’s string and creates a copy that has all non-space letters replaced with a tilde symbol (~)
3. Prompts the user to enter a character
4. Searches the original string to locate all instances of the character. When a match is found, the corresponding tilde in the new string is changed into the requested character.
5. The number of replaced characters is reported to the user.

Here is an example operation of the completed program:

```
Please enter a string:
the weather is great!
The tilde-version of your string is:
~~~ ~~~~~~~ ~~ ~~~~~~
Please tell me the char that you would like to locate and replace:
e
The new version of your string is:
~~e ~e~~~e~ ~~ ~~e~~~
I located 4 ‘e’ characters!
```

You should design first. You can have more functions, but you must have at least the three below with the EXACT function prototypes.

Each of your functions should have a proper function header/description.

```
void get_string(string *);
```

Explanation: get_string will accept a pointer to a string and replace the contents of that string. The user's input will be stored in the string.

```
string convert_to_tilde(string *);
```

Explanation: convert_to_tilde will accept a pointer to a string and create a copy. The copy will be identical to the original string (which was passed by pointer) except for one major change… in the copied string, all non-space characters will be replaced by a tilde (~).

```
int search_and_replace(char, string, string &);
```

Explanation: search_and_replace will accept a char, a string (the original user input string), and a reference to a string (the new version that contains tilde symbols). The function will search though the original string and look for the requested char. Any time that character is discovered in the original string, the tilde-version of the string will have the tilde replaced by that requested character. This function will return an int, which represents the number of times the character was replaced.

Write the function headers/descriptions for each of the functions above, as well as any other functions you create. This includes information about parameters, return values, and pre/post conditions.

As a reminder, here is an example function header template that you can use as a model in your program. I'm generally lenient with the formatting of **lab** work, but your **homework** submissions need to include a header for each function (except for **main**).

```
/******************************************************
** Function: convert_to_tilde
** Description:
**    This function accepts a pointer to a string and creates a copy.
**      The copy will be identical to the original string (which was passed by
**      pointer) except for one major change... in the copied string, all
**      non-space characters will be replaced by a tilde (~)
** Parameters
**    str_in --> A pointer to the C++ string that will be used as the basis for
**      the copy. This string will not be modified.
** Pre-Conditions: None
** Post-Conditions:
**    Returns a newly created string that is exactly the same length as the string
**      parameter that was passed via pointer. However, each non-space character
**      in this new string will be replaced by a tilde.
******************************************************/
```

Design – As you design your program, consider the important points listed below.

- Why do the function prototypes have the specific parameter types on them?
- Why do the functions have void or a return value?
- How do all the functions interact together?

Testing – Provide testing values with expected results.

- What do you plan to use as bad values? Good values?
- What do you expect to happen with bad values? Good values?

If you have any questions, feel free to ask for feedback before beginning to implement your design. This will help with logic and function-related mistakes**.**

How to get checked off…

You will receive 7 points for attending lab and the remaining 3 points after showing your working solution to a TA or the instructor. If you do not complete the entire work prior to the end of lab, be sure that you finish it outside of class (before next week's lab).