Assignment #2

CS 162: Introduction to Computer Science

Submit your assignment to the D2L Dropbox
Email a backup copy to karlaf@pdx.edu

Remember to write an algorithm in the header comments of your program!

The purpose of the second program is to continue refining our programming skills in C++ and practice using **arrays of characters, the subscript operator, the cstring library, and functions with arguments**. Our goal is to create programs with a small functions where main delegates to a series of functions where the real work takes place. In this programming assignment, you are **not** allowed to use global variables. Avoid using break *(unless working with a switch statement).* Limit your functions to no more than 20 statements of code (for executable statements… *not counting variables, blank lines, lines with just curly brackets, or comments*).

Program Assignment:
In program #2 we will write another game. This time, it will be a word game. The first player will type in phrase, attempting to hide a word inside of the phrase that the other player will guess. The second player gets 5 chances to guess what the word is that is hidden within the phrase. They get to see the phrase. For each turn, a player gets 10 points. Each time they incorrectly guess a word, 2 points are deducted from their score. After either guessing correctly or incorrectly 5 times, the players switch turns. The second player now gets 10 points, and so on. The player with the highest score wins.

Here are the details about this game:

1. The first player enters in a long phrase (no more than 80 characters) along with the correct hidden word (no more than 10 characters). Hide the correct word quickly by clearing the screen.
2. The second player guesses, by typing in a word (no more than 10 characters); the second player should be able to see the phrase (so echo it out)
    a. The guessed word should NOT be case sensitive (so if the player guesses "HI" and the letters "hi" exist in the phrase, then it should match!)

3. Your program will double check that the word entered actually exists in the original phrase. Spaces and tabs should be ignored.
    a. If it doesn't exist in the phrase, the second player automatically loses their turn and all 10 points. So if the phrase was "Hello world" and the word "help" is guessed. "help" does not exist in the phrase, and the player doesn't get any more chances. Time to switch players (after taking away their 10 points!)

    b. If the word does exist, then it is compared to the correct word. If it is a match, we switch turns
        i. An extra bonus point is awarded if spaces were ignored between characters in the phrase to compose the word. For example, if the phrase was "The elf was small" and the guessed word is "eel", the player will get an extra point!

Things you should know...as part of your program:

1. Make sure to prompt the user for any input requested. Make sure it is clear from your prompts what the user is expected to do.
2. Use the cstring library with strlen to determine the length of the words entered.
3. Use the subscript operator [] to match letters, using a loop (recommend for loops)
4. You may not use any global variables in this program!
5. You may not use the string class – instead use arrays of characters with cstrings
6. Make sure to use C++'s I/O (iostream library) for your input and output.
7. After each input operation, make sure to use cin.ignore to remove the delimiters!
8. FUNCTIONS are required for this assignment

To get full credit for the programming portion, you will need to:

1. Turn in an algorithm written using full English sentences (it may be provided in outline form, paragraph form, or graphical (such as a data flow diagram)).
2. Program using a consistent style of indentation, header comments for each function, inline comments for each major block of code
3. Make sure to put your name in your program
4. Submit an electronic copy of your .cpp file as an attached file to the dropbox on D2L (go to: http://d2l.pdx.edu/ to login). Make sure to hit the submit button after uploading your files (otherwise they will be lost)
5. As a backup, please also email your work (as attached file(s)) to karlaf@pdx.edu