Program #1
CS 162: Introduction to Computer Science

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

LATE work will be accepted, but only by 6pm Wednesday of the following week, for 5% off (the last assignment cannot be turned in late).
Assignments may not be turned in later than the late due date. There are no exceptions.

Overview

The purpose of this first program is get you familiar with the fundamental syntax of C++ and the process of writing and turning in programs at PSU. Make sure to start by writing an algorithm. It is worth 20% of your grade on the program.

Specification

When you are with your best friend or colleague, how do you decide who goes first? One way is to play the game "rock, paper, scissors". You both select a choice. The rock beats scissors, scissors beats paper, and paper beats the rock. If the same selection is made, the game is tied and it must be played again. This continues until there is a winner. The player that is the winner and how many times a tie takes place should be displayed.

After a player enters a choice you will want to quickly clear the screen.

Your job will be to:

1. Welcome the user to the problem and instruct them on how the purpose of the program
2. Prompt the two players to enter in their choice each following by an enter. Describe what choices the player has when entering in values. Start by something simple such as R, P, or S for the three choices.
3. Read in the first player's choice, clear the screen, then prompt and read in the second player's choice. Clear the screen again.
    a. The grader will enter in the correct types of data. So if you ask for an integer, the grader will enter a whole number. If you ask for a character (R, P, S), then the grader can enter in any digit or letter
    b. Capitalize the letter so that if the user enters an 'r', the program will be using a 'R'.
    c. Your program will also need to error check that the appropriate values are entered. So if you ask for R (for rock) but the user enters

a V (for volcanic rock) it isn't a match and they will need to try again!

d. Lastly, be concerned about the situation where the user enters in too many characters (Rock instead of R). Your program needs to make sure those extra characters do not impact subsequent user inputs later in the program. It is unclear from the user's answer what their selection is, then please have them reenter their choice and try again.

4. Determine who is the winner based on the two players' inputs. Write the algorithm for this!
5. Output the results
6. If there is a tie, do this again
7. At the end display how many ties there were and which player was the winner

When creating your assignment, really think about how to make the program as user friendly as possible. Does the user understand how to use the program? Are the prompts clear? Does the user know what to enter and what will happen if they enter in something incorrect? Try to have someone that is NOT a programmer use your program and give you feedback as to what could be improved from a user interface perspective.

For **EXTRA CREDIT**, provide the additional features: (+3 points for each)

1. Find out if the player wants to play rock, paper, scissors against the computer instead of playing against another player. If they do, then use the random number generator to determine the computers choice of one of those three selections. If you use the random number generator, you will need to "seed" it first (otherwise the results are not random!).
   a. Use stdlib.h and time.h
   b. srand(time(0); will seed the random number generator
   c. variable = rand() %3; will use the random number generator and keep it within the range of three values (0, 1, 2)
2. Use arrays of characters to represent the two players' names.

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. You will need to use a loop if there is a tie. Consider a do-while
3. Do not use "break" unless using a switch statement
4. NEVER use a "goto"
5. Never use global variables (global constants are fine)

6. You may not use the string class but you are allowed to use arrays if you want to work ahead!
7. Make sure to use C++'s I/O (iostream library) (and not C's stdio)
8. Although functions are not required for this assignment, it is advised to start learning how to write functions in C++

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

1. Turn in an algorithm written using full English sentences.
2. Program using a consistent style of indentation, header comments for each function, inline comments for each major block of code
3. 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)