**CIS 246 – Spring 2020**

|  |  |
| --- | --- |
| **Program:** | **5** |
| **Points:** | **20** |
| **Chapter(s):** | **6** |
| **File(s) to Submit:** | **RPSDriver.cpp, Player.h (as one zip file)** |
| **Due:** | **March 18, 11:59 pm** |

**Summary**

Write a C++ application that uses overloaded functions and random number generation to simulate a game of rock, paper, scissors between a player and the computer.

Extra credit is available for this assignment.

**Description**

Write a driver program that creates two objects of the Player class. One object represents the player, and the other object represents the computer.

Prompt the player to type in their play. Generate a random number to determine the computer’s play. Determine the winner and print the results to the screen.

**Requirements for the RPSDriver Program**

1. **Create two objects of the Player class, for the player and the computer.**
2. **Prompt the player to type in their play as a STRING.**

Plays are the values “rock”, “paper”, or “scissors”. Call the appropriate set function in the player object using whatever value is typed. You do not have to check for spelling mistakes or values other than rock, paper, or scissors.

1. **Generate a random number to determine the computer’s play.**

Call the appropriate set function in the computer object using the value generated.

1. **Determine the winner.**

Use the get function from each object to retrieve the play for each and determine the winner of the game, using standard rock, paper, scissors logic.

1. **Display the result to the screen.**

**Requirements for the Player Class**

Private data member:

play (as an integer)

Public functions:

1. A function to return the value of **play** as a **string** return value
2. An overloaded function to set the value of **play**. This function uses a **string** parameter.
3. An overloaded function to set the value of **play**. This function uses an **int** parameter.

**Extra Credit #1 – Maximum 10 Points**

Give the user the option to play the game by typing in the words rock, paper, or scissors, or by typing numbers to represent the plays.

1. Add another function to the Player class to return the play as an **int** return value.
2. In RPSDriver, move the logic to determine the winner into **two overloaded functions**. One function must use two string parameters, and one must use two integer parameters. In both cases, the parameters are the player’s and computer’s play.
3. Call the appropriate overloaded function from main using the appropriate functions of the Player class as arguments. **This function call must contain two other function calls as arguments.**

**Extra Credit #2 – Maximum 10 Points**

You must implement Extra Credit Option #1 first.

Replace the two overloaded functions in RPSDriver that determine the winner with **one template function** that determines the winner.

Prototypes are not used with template functions, so this function must be defined above main.

**General Requirements**

For complete credit, you must:

1. **MEET ALL REQUIREMENTS ACCORDING TO THE INSTRUCTIONS** – Follow the instructions as written for completing this project, even if you [think you] know a “better” way to do something.
2. **INCLUDE COMMENTS** – Include comments in your code. There must be a comment at the top of each source code or header file that includes your name, the assignment number, and a description of the code in that file. There must be comments at each important step in your algorithm that describes that step.
3. **FOLLOW BEST PRACTICES** – Follow best practices in C++ programming, including, but not limited to, appropriate use of private/public, appropriate use of classes and/or header files, sets & gets, white space, alignment, meaningful variable names, naming conventions, using statements, etc. Points will be deducted for sloppy code that is hard to read, even if it works, so pay attention to these details.
4. **SUBMIT ALL FILES BEFORE THE DUE DATE** – Submit a .zip of ONLY source code files to the dropbox for this assignment on Canvas before the due date. Do not submit anything except .cpp and/or .h, within a zip. Do not submit .exe files. Do not submit a folder structure. Do not submit project files from an IDE.

**Sample Runs**



