Rhea Mae Edwards

Assignment #3 Design/Testing

Understanding the Problem

This problem is asking me to create a program that will overall conduct a semi-traditional game of Rock, Paper, Scissors. The program must contain class named Tool along with required functions and members. Also, there should be three additional classes named Rock, Paper, and Scissors that will have an “is a” relationship with the class Tool, and these three additional function also have functions that have certain requirements that need to be followed also. As a final addition, another class named RPSGame will need to a part of the program, which will overall play a big role in conducting the entire game. In the file named play_game.cpp, acting as the main file, will overall give a based to the entire program- helping provide an interface to the user. The tools of this programmed game must have certain valued strengths throughout the game, as it is played in the traditional fashion. The game will only be played by one user versus a programmed opponents that cannot be programmed to cheat by reading what the user has chosen as its input. Then after each round, the program must also output the scores of the overall game. The program must allow the user to keep on playing until it does not want to play anymore.

Devising a Plan/Design

Create a Tool header file which will contain the class named Tool.

This class will have an int field called strength and a char field called type.

It will also have a void setStrength(int) function that will set the strength for the Tool.

Create a Tool cpp file which will hold all the functions declared in the Tool class.

Create a Rock header file which will contain the class named Rock.

Create a Rock cpp file which will hold all the functions declared in the Rock class.

Create a Paper header file which will contain the class named Paper.

Create a Paper cpp file which will hold all the functions declared in the Paper class.

Create a Scissors header file which will contain the class named Scissors.

Create a Scissors cpp file which will hold all the functions declared in the Scissors class.

The three previously stated classes will inherit from the Tool class.

Each of the three classes will have a default constructor that sets their strength to 1, and a non-default constructor which will take in an int used to initialize its strength field.

Their constructors should also initialize the type field using 'r' for Rock, 'p' for Paper, and 's' for Scissors.

Each of the three classes will need a public function declared as bool fight(Tool) comparing and changing strength values.

Create a RPSGame header file which will contain the class named RPSGame.

Create a RPSGame cpp file which will hold all the functions declared in the RPSGame class.

In the previous stated class, it must contain two Tool * for the human and computer tool.

This class must also have three int fields named human_wins, computer_wins, and ties.

Create a play_game cpp file which will provide a base to the overall functionality of the program.

Create a make file.

Testing

In making my program, I used the help of information learned from lectures, lab, internet, teacher assistants, and peers. For example, with figuring out how to set up and use an "is a" relationship with four the classes, I studied pervious lecture slides to successfully write up the correlation.

Input Values	Expected Output	Did Actual Meet Expected?
start = 'y'	To ask the user for the initial strength values for each tool of the game.	Yes
start = 'n'	To continue on with the program by asking the user what tool/option they would like to pick.	Yes
start = 1	To end the program along with an error message.	Yes
play = 'e'	To end the program along with an ending message.	Yes
play = 1	To end the program along with an error message.	Yes
play = 'r'	To continue on with the program by outputting what the computer chose as its pick.	Yes
play = 'p'	To continue on with the program by outputting what the computer chose as its pick.	Yes
play = 's'	To continue on with the program by outputting what the computer chose as its pick.	Yes