Community College of Philadelphia

CSCI112 – Computer Science II
ASSIGNMENT 1

Instructor: Michael Hackett
Department: Computer Science
Email: mhackett@ccp.edu

Rock Paper Scissors Game

Create a program that plays a game of rock, paper, scissors.

Your program should allow the user to enter “rock”, “paper”, or “scissors”. The program will randomly select “rock”, “paper”, or “scissors”. The winner is then printed.

GameFunctions Class (GameFunctions.java)

This class should contain the following:

- Two Fields, both private
  - rGen - A Random object for generating random numbers.
  - cpuChoice - A String for holding the computer’s current choice.
- One Constructor, public
  - Accepts no arguments.
  - Instantiates a Random object for the rGen field. Do not give the Random object a seed value.
  - Assigns cpuChoice an empty String.
- Two Methods, both public
  - getRPS
    - Accepts no arguments and returns no value.
    - Randomly picks a number between 1, 2 and 3, each one associated with “rock”, “paper”, or “scissors”.
    - The method should assign the String value “rock”, “paper”, or “scissors” to cpuChoice.
    - Print the computer’s choice.
  - whoWon
    - Accepts one argument, a String called userChoice, and returns a String.
    - The function should return the String “user” if the user won, “computer” if the program won, or “tie” if neither won.
    - Rules for determining the winner:
      - Rock beats Scissors
      - Scissors beats Paper
      - Paper beats Rock
      - If the user and computer made the same choice, it’s a tie.

GameDemo Class (GameDemo.java)

This is the class that will use the GameFunctions class you created.

In the main method:

- Instantiate a GameFunctons object.
- Prompt the user to enter their choice. Reject any values other than “rock”, “paper”, or “scissors” by displaying an error message and prompt the user to reenter their choice.
    - Uppercase/lowercase letters should be ignored.
- Call the GameFunctions object’s getRPS method so that is selects the computer’s choice.
- Pass the user’s choice to the GameFunctions object’s whoWon.
- Based on the String value returned by the method, print the winner.
- Ask the user if they would like to play again. If they choose to play again, start the game over from the beginning.
    - Allow Y or y to restart the game.

UML Diagram

You will also need to create a UML diagram of your GameFunctions class. You may do this in MS Word, MS PowerPoint, or even MS Paint (or comparable programs). Acceptable file formats will be PDF, JPG, or PNG.

Sample Input/Output

```
Enter your choice (rock, paper, or scissors): paper
The computer chose scissors.
You lose!
Would you like to play again? (Y for yes): Y

Enter your choice (rock, paper, or scissors): dog
Invalid choice. Please try again.
Enter your choice (rock, paper, or scissors): rock
The computer chose rock.
It’s a tie!
Would you like to play again? (Y for yes): Y

Enter your choice (rock, paper, or scissors): rock
The computer chose paper.
You win!
Would you like to play again? (Y for yes): N
```

- Your program’s output must exactly match the formatting in the above example, using complete sentences like the output shown.
- Be sure to use comments to document your code. Comments show me that YOU can explain, in plain English) what your program’s code is doing.

Grading

See Assignment Rubric in Canvas.