**Overview**

This purpose of this activity is to demonstrate your proficiency in basic Java syntax involving console input and output, variables, flow of control statements, and expressions. These core concepts are the building blocks for more complicated code that is to come.

In this activity, you will write a program that plays the game Rock, Paper, Scissors.

**Task List**

1. Create a flowchart that describes what the program will do.
2. Show your flowchart to your instructor.
3. After receiving approval for your flowchart, begin writing code.
4. Write the code in stages, checking that each stage works before continuing to the next stage.
5. Once all parts of the code work as expected and the program meets all requirements listed below, submit the code using instructions provided by your instructor.

**Rules**

The rules of the game are as follows:

1. Each player chooses Rock, Paper, or Scissors.
2. If both players choose the same thing, the round is a tie.
3. Otherwise:
   1. Paper wraps Rock to win
   2. Scissors cut Paper to win
   3. Rock breaks Scissors to win

**Requirements**

This program will be a Java console application named *RockPaperScissors*.

1. The program first asks the user how many rounds he/she wants to play.
   1. Maximum number of rounds = 10, minimum number of rounds = 1.
   2. If the user asks for something outside this range, the program prints an error message **and quits**.
   3. If the number of rounds is in range, the program plays that number of rounds.
      1. Loop
   4. Each round is played according to the requirements below.
2. For each round of Rock, Paper, Scissors, the program does the following:
   1. The computer asks the user for his/her choice (Rock, Paper, or Scissors).
      1. Hint: 1 = Rock, 2 = Paper, 3 = Scissors
   2. After the computer asks for the user’s input, the computer randomly chooses Rock, Paper, or Scissors and displays the result of the round (tie, user win, or computer win).
      1. Hint: Use the Random class.
3. The program must keep track of how many rounds are ties (int draws), user wins (int userWins), or computer wins.
   1. Hint: Create three variables to keep track of these items and update them correctly after each round.
4. At the end of the last round, the program must print out the number of ties, user wins, and computer wins and declare the overall winner based on who won more rounds.
5. After all rounds have been played and the winner declared, the program must ask the user if he/she wants to play again.
   1. If the user says No, the program prints out a message like, “Thanks for playing!” and then exits.
   2. If the user says Yes, the program starts over, asking the user how many rounds he/she would like to play.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Criteria** | **Meets Expectations** | **Needs Improvement** | **No Credit** |
| 1 | Flowchart: The flowchart for Rock, Paper, Scissors reasonably reflects the flow required for the activity. | ***15*** | ***8*** | ***0*** |
| 2 | Application Name: Application is named correctly. | ***5*** | ***3*** | ***0*** |
| 3 | Variables: Learner names and uses variables correctly. | ***10*** | ***6*** | ***0*** |
| 4 | Loops: Learner uses loops correctly. | ***10*** | ***6*** | ***0*** |
| 5 | Conditionals: Learner uses conditionals correctly. | ***10*** | ***6*** | ***0*** |
| 6 | Methods: Learner uses main methods correctly. | ***10*** | ***6*** | ***0*** |
| 7 | Arrays: Learner uses arrays correctly. | ***10*** | ***6*** | ***0*** |
| 8 | Operators: Learner uses arithmetic operators correctly. | ***10*** | ***6*** | ***0*** |
| 9 | Code Style: Code is written with appropriate indents, naming conventions, and comments so that other developers can read the code easily | ***10*** | ***6*** | ***0*** |