Assignment 3 - Rock Paper Scissors

This assignment will test your ability to use switch statements and loops.

1 The Assignment

You will write a text based version of the game Rock Paper Scissors, to be played by two players. You can find out how to play the game here: www.wikihow.com/Play-Rock,-Paper,-Scissors.

1.1 Instructions

Write the class `RockPaperScissors` in the file `RockPaperScissors.java`. The `main` method of the `RockPaperScissors` class should handle all printing and user input.

Your program should allow an indefinite number of rounds of rock paper scissors to be played. At the beginning of each round, your program should prompt player one to make their choice and store it. Then it should prompt player two to make their choice and store it. After both players have chosen, your program should print out the choice that each player made, the result, and the total score of all rounds played so far. Then a new round will start. Either player should be able to quit at any time, in which case the final result should be printed.

Your program should accept an integer from 1 - 4 as input, where the meaning of each choice is as follows:

Input	Meaning
1	rock
2	paper
3	scissors
4	quit

See the expected input/output section below for more details.

1.2 Tips

- Every string that is printed must end with a newline character.
- The `System.out.println` method automatically appends a newline character to the string it prints.
- Loops can be labelled, and break statements can be given a loop label to break, rather than breaking their containing loop.
- Do not use any system calls such as `System.exit(0)` from within your program. The Java autograder will give you zero if you do this.
- It is very important that your program terminates as shown in the examples below.

1.3 Expected Input/Output

1.3.1 Example 1

```
OUT > Player One: Choose your weapon!
IN  > 1
OUT > Good choice!
OUT > Player Two: Choose your weapon!
IN  > 2
OUT > Good choice!
OUT > Player One chose: rock
OUT > Player Two chose: paper
OUT > Player Two Wins!
OUT > The score is now: 0 - 1
OUT > Player One: Choose your weapon!
IN  > 3
OUT > Good choice!
OUT > Player Two: Choose your weapon!
IN  > 2
OUT > Good choice!
OUT > Player One chose: scissors
OUT > Player Two chose: paper
OUT > Player One Wins!
OUT > The score is now: 1 - 1
OUT > Player One: Choose your weapon!
IN  > 4
```

```
OUT > Player One and Player Two are tied with 1 points.
OUT > Quitting
```

1.3.2 Example 2

```
OUT > Player One: Choose your weapon!
IN  > 2
OUT > Good choice!
OUT > Player Two: Choose your weapon!
IN  > 1
OUT > Good choice!
OUT > Player One chose: paper
OUT > Player Two chose: rock
OUT > Player One Wins!
OUT > The score is now: 1 - 0
OUT > Player One: Choose your weapon!
IN  > 4
OUT > The winner is Player One with 1 points.
OUT > Quitting
```

1.3.3 Example 3

```
OUT > Player One: Choose your weapon!
IN  > 3
OUT > Good choice!
OUT > Player Two: Choose your weapon!
IN  > 1
OUT > Good choice!
OUT > Player One chose: scissors
OUT > Player Two chose: rock
OUT > Player Two Wins!
OUT > The score is now: 0 - 1
OUT > Player One: Choose your weapon!
IN  > 1
OUT > Good choice!
OUT > Player Two: Choose your weapon!
IN  > 2
OUT > Good choice!
OUT > Player One chose: rock
OUT > Player Two chose: paper
```

```
OUT > Player Two Wins!
OUT > The score is now: 0 - 2
OUT > Player One: Choose your weapon!
IN  > 4
OUT > The winner is Player Two with 2 points.
OUT > Quitting
```

1.3.4 Example 4

```
OUT > Player One: Choose your weapon!
IN  > 1
OUT > Good choice!
OUT > Player Two: Choose your weapon!
IN  > 1
OUT > Good choice!
OUT > Player One chose: rock
OUT > Player Two chose: rock
OUT > It's a draw!
OUT > The score is now: 0 - 0
OUT > Player One: Choose your weapon!
IN  > 4
OUT > Player One and Player Two are tied with 0 points.
OUT > Quitting
```

2 Submitting to the Autograder

1. Complete your assignment, making sure your program's output matches the expected output stipulated by the assignment brief.
2. Make sure that your program compiles and runs without any errors.
3. Create a `zip` file containing the `RockPaperScissors.java` file.
4. Upload the `zip` file to Athena.