# General Outline: Rock-Paper-Scissors in Java

## 1. Import Statements

* Import necessary Java utilities for user input (e.g., Scanner, Random).

## 2. Class Definition

Java

|  |
| --- |
| public class RockPaperScissors  { |
| public static void main(String[] args)   { |
| // Main method serves as the entry point. |
| } |
| } |

## 3. Initialize Components

* **Game Rules**:   
  Winning conditions are Rock beats Scissors, Scissors beats Paper and Paper beats Rock.  
  Draw conditions are Rock – Rock, Paper – Paper and Scissors – Scissors.
* **User Input**: Set up a Scanner to allow the player to enter their choice.
* **Computer Move**: Use a Random number generator to simulate the computer's choice.

## 4. Main Game Loop

* Repeat the game until the player decides to quit.
* Displayer players high score when prompting if they would like to play again.

1. **Display Choices**: Prompt the user for their move (Rock, Paper, or Scissors).
2. **Generate Computer's Move**: Randomly select one of the three options.
3. **Determine the Winner**:
   * Compare user and computer moves based on the rules.
   * Display the result (win, lose, or draw).
   * Display the Computer and Player Choices (You Choice: Rock | Computer Choice: Paper).
4. **Score Updating**: Calculate the current score and update the high score.
5. **Ask for Replay**: Allow the user to play again or quit.

## 5. Helper commands

* **Display Rules**: A choice given to the player to display the rules (e.g., Help, Tips, Rules).
* **Get Computer Move**: A Command given to test generate the computer’s choices.
* **Determine Winner**: A Command used to give a test win condition for either Player or Computer.

## 6. Features to Expand

* Add a score counter for the user and computer.
* Allow for more complex rules (e.g., Rock-Paper-Scissors-Lizard-Spock).
* Use a graphical user interface (GUI) for a more interactive experience.