

```

1  /**
2   * header
3   */
4
5  import java.util.Scanner;
6  import static java.lang.System.*;
7  import java.util.Arrays;
8
9  public class _101RPS
10 {
11     private String playChoice;
12     private String compChoice;
13     private String[] choicesArray = new String[] {"dynamite", "helicopter", "tank",
14         "sky", "nuke", "laser", "power", "medusa", "lightning", "electricity", "heat",
15         "robot", "math", "video game", "fence", "devil", "gold", "platinum", "diamond",
16         "dragon", "satan", "mountain", "prayer", "alien", "UFO", "rainbow", "TV", "water",
17         "rain", "beer", "cup", "bowl", "guitar", "planet", "air", "toilet", "film",
18         "grass", "moon", "airplane", "cloud", "paper", "book", "butter", "church",
19         "sponge", "vampire", "money", "cross", "community", "brain", "cockroach", "spider",
20         "fish", "bird", "cat", "wolf", "duck", "turnip", "tree", "bicycle", "noise", "car",
21         "train", "home", "man", "baby", "woman", "police", "princess", "prince", "queen",
22         "king", "monkey", "vulture", "porcupine", "blood", "snake", "castle", "computer",
23         "peace", "axe", "cage", "poison", "scissor", "school", "chainsaw", "fire",
24         "camera", "sun", "wall", "death", "rock", "sword", "whip", "law", "gun", "chain",
25         "pit", "quicksand", "tornado"};
26     //default constructor
27     public _101RPS()
28     {
29         playChoice = "";
30         compChoice = "";
31     }
32
33     //loaded constructor
34     public _101RPS(String player)
35     {
36         setPlayers(player);
37     }
38
39     public void setPlayers(String player)
40     {
41         playChoice = player;
42
43         int num = 0 + (int)(Math.random() * 101);
44         compChoice = choicesArray[num];
45     }
46
47     public String determineWinner()
48     {
49         if (compChoice.equals(playChoice))
50             return "!Draw Game!";
51         //if playChoice is the same as compChoice no winner (draw)
52         int index = Arrays.asList(choicesArray).indexOf(playChoice);
53         int c = 1;
54         for (int i = ((choicesArray.length)-1)/2; i > -1; i--)
55         {
56             if (index - c == -1)
57             {
58                 index = choicesArray.length;
59             }
60         }
61     }
62 }

```

```
47         c = 1;
48     }
49     if (compChoice.equals(choicesArray[index - c]))
50     {
51         return ( "Player Wins <<" + playChoice + " beats " + compChoice + ">>");
52     }
53     c++;
54 }
55 return ( "Computer wins <<" + compChoice + " beats " + playChoice + ">>");
56 }
57
58 public String toString()
59 {
60     String output="";
61     output+="player had " + playChoice+"\n";
62     output+="computer had " + compChoice;
63     return output;
64 }
65
66 public boolean validWeapon ()
67 {
68     if (Arrays.asList(choicesArray).indexOf(playChoice) == -1)
69         return false;
70     else
71         return true;
72 }
73 }
74
```