

```

1 import java.util.Random;
7
8 public abstract class CollegeBasketballTeamComponentSecondary
9     implements CollegeBasketballTeam {
10
11     private Map1L<String, Map1L<Integer, Double>> teamData;
12
13     @Override
14     public void runSimulation(CollegeBasketballTeam team1,
15                             CollegeBasketballTeam team2) {
16
17         SimpleWriter out = new SimpleWriter1L();
18
19         // Retrieve statistics for both teams
20
21         Sequence1L<String> categories1 =
22     team1.getAllCategories();
23         Sequence1L<String> categories2 =
24     team2.getAllCategories();
25
26         // Calculate score for team 1
27         double score1 = 0.0;
28         for (String category : categories1) {
29             Map1L<Integer, Double> statistics = team1
30                 .getStatisticsByCategory(category);
31             for (Map1L.Pair<Integer, Double> entry : statistics)
32             {
33                 score1 += entry.value();
34             }
35         }
36
37         // Calculate score for team 2
38         double score2 = 0.0;
39         for (String category : categories2) {
40             Map1L<Integer, Double> statistics = team2
41                 .getStatisticsByCategory(category);
42             for (Map1L.Pair<Integer, Double> entry : statistics)
43             {
44                 score2 += entry.value(); // Accumulate scores
45             }
46         }
47
48         // Introduce some randomness

```

```

45     Random random = new Random();
46     score1 *= random.nextDouble(); // Scale the score by a
    random factor between 0 and 1
47     score2 *= random.nextDouble();
48
49     // Determine the winner
50     String winner;
51     if (score1 > score2) {
52         winner = "Team 1";
53     } else if (score1 < score2) {
54         winner = "Team 2";
55     } else {
56         winner = "Tie"; // Handle tie scenario
57     }
58
59     // Print simulation results
60     out.println("Simulation Results:");
61     out.println("Team 1 Score: " + score1);
62     out.println("Team 2 Score: " + score2);
63     out.println("Winner: " + winner);
64     out.close();
65 }
66
67 @Override
68 public String toString() {
69     StringBuilder sb = new StringBuilder();
70     sb.append("Team Statistics:\n");
71     for (Map1L.Pair<String, Map1L<Integer, Double>>
category : this.teamData) {
72         sb.append(category.key()).append(":\n");
73         for (Map1L.Pair<Integer, Double> statistic :
category.value()) {
74             sb.append("Rank
").append(statistic.key()).append(": ")
75                 .append(statistic.value()).append("\n");
76         }
77     }
78     return sb.toString();
79 }
80
81 @Override
82 public boolean equals(Object team2) {
83     String team2Str = team2.toString();
84     String team1Str = this.toString();

```

```
85
86         return team1Str.equals(team2Str);
87     }
88
89     @Override
90     public int hashCode() {
91         return this.teamData.hashCode();
92     }
93
94 }
95
```