

```

1 public class TennisGameTest {
2     @Test
3     public void player1ShouldHaveOnePoint() {
4         //given
5         TennisGame tennisGame = new TennisGame();
6         String firstPlayerName = "player1";
7         tennisGame.firstPlayerPoints = 0;
8
9         //when
10        tennisGame.getPoint(firstPlayerName);
11
12        //then
13        assertEquals(1, tennisGame.firstPlayerPoints);
14    }
15
16    @Test public void player2ShouldHaveOnePoint() { }
17
18    @Test public void shouldReturnLoveToLove() throws Exception {
19        tennisGame.firstPlayerPoints = 0;
20        tennisGame.secondPlayerPoints = 0;
21        String score = tennisGame.score();
22        assertEquals("Love-Love", score);
23    }
24
25
26    @Test public void shouldReturn15ToLove() throws Exception {
27        tennisGame.firstPlayerPoints = 1;
28        tennisGame.secondPlayerPoints = 0;
29        String score = tennisGame.score();
30        assertEquals("15-Love", score);
31    }
32
33    @Test public void shouldReturn15To15() throws Exception {
34        tennisGame.firstPlayerPoints = 1;
35        tennisGame.secondPlayerPoints = 1;
36        String score = tennisGame.score();
37        assertEquals("15-15", score);
38    }
39
40
41    @Test public void shouldReturn30To15() throws Exception {
42        tennisGame.firstPlayerPoints = 2;
43        tennisGame.secondPlayerPoints = 1;
44        String score = tennisGame.score();
45        assertEquals("30-15", score);
46    }
47
48    @Test public void shouldReturn30To40() throws Exception {
49        tennisGame.firstPlayerPoints = 2;
50        tennisGame.secondPlayerPoints = 3;
51        String score = tennisGame.score();
52        assertEquals("30-40", score);
53    }
54
55
56    @Test public void shouldWin1Player() throws Exception {
57        firstPlayerPoints = 4; secondPlayerPoints = 0;
58        tennisGame.firstPlayerName="Player1";
59        String score = tennisGame.score();
60        assertEquals("Win-Player1", score);
61    }
62
63    @Test public void shouldReturnDeuce() throws Exception {
64        firstPlayerPoints = 5; secondPlayerPoints = 5;
65        String score = tennisGame.score();
66        assertEquals("Deuce", score);
67    }
68
69    @Test public void shouldReturnAdvantage() throws Exception {
70        firstPlayerPoints = 8; secondPlayerPoints = 9;
71        String score = tennisGame.score();
72        assertEquals("Advantage", score);
73    }
74 }

```