

```

1 import static org.junit.Assert.*;
2
3 import org.junit.Before;
4 import org.junit.Test;
5
6 public class CycleCalculatorTest
7 {
8     private static final double ACCURACY = 0.001;
9
10    private CycleCalculator cycleCalculator;
11
12    @Before
13    public void setup()
14    {
15        cycleCalculator = new CycleCalculator();
16    }
17
18    @Test
19    public void testCompetency()
20    {
21        CycleRide cycle1 = new CycleRide(10, true);
22        Cyclist cyclist1 = new Cyclist("Beginner", 0);
23        Weather weather = new Weather(15,0,false);
24        assertEquals("Beginner competency level failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);
25
26        CycleRide cycle2 = new CycleRide(15, true);
27        Cyclist cyclist2 = new Cyclist("Intermediate", 0);
28        assertEquals("Intermediate competency level failed:", 1.0,
cycleCalculator.getDuration(cycle2, cyclist2, weather), ACCURACY);
29
30        CycleRide cycle3 = new CycleRide(20, true);
31        Cyclist cyclist3 = new Cyclist("Advanced", 0);
32        assertEquals("Advanced competency level failed:", 1.0,
cycleCalculator.getDuration(cycle3, cyclist3, weather), ACCURACY);
33    }
34
35    @Test
36    public void testCycleBuddy()
37    {
38        CycleRide cycle1 = new CycleRide(12, false);
39        Cyclist cyclist1 = new Cyclist("Beginner", 0);
40        Weather weather = new Weather(15,0,false);
41        assertEquals("Test cycling with a buddy failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);
42
43        CycleRide cycle2 = new CycleRide(10, true);
44        Cyclist cyclist2 = new Cyclist("Beginner", 0);
45        assertEquals("Test cycling alone failed:", 1.0,
cycleCalculator.getDuration(cycle2, cyclist2, weather), ACCURACY);
46    }
47
48    @Test
49    public void testYearsExperience()
50    {
51        CycleRide cycle1 = new CycleRide(11, true);
52        Cyclist cyclist1 = new Cyclist("Beginner", 5);

```

```

53     Weather weather = new Weather(15,0,false);
54     assertEquals("Test years experience failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);
55 }
56
57 @Test
58 public void testLessThan10()
59 {
60     CycleRide cycle1 = new CycleRide(9, true);
61     Cyclist cyclist1 = new Cyclist("Beginner", 0);
62     Weather weather = new Weather(0,0,false);
63     assertEquals("Test temperature less than 10 failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);
64
65     CycleRide cycle2 = new CycleRide(10, true);
66     Cyclist cyclist2 = new Cyclist("Beginner", 0);
67     Weather weather2 = new Weather(15,0,false);
68     assertEquals("Test temperature more than 10 failed:", 1.0,
cycleCalculator.getDuration(cycle2, cyclist2, weather2), ACCURACY);
69 }
70
71 @Test
72 public void testMoreThan20()
73 {
74     CycleRide cycle1 = new CycleRide(9, true);
75     Cyclist cyclist1 = new Cyclist("Beginner", 0);
76     Weather weather = new Weather(30,0,false);
77     assertEquals("Test temperature more than 20 failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);
78
79     CycleRide cycle2 = new CycleRide(10, true);
80     Cyclist cyclist2 = new Cyclist("Beginner", 0);
81     Weather weather2 = new Weather(15,0,false);
82     assertEquals("Test temperature less than 20 failed:", 1.0,
cycleCalculator.getDuration(cycle2, cyclist2, weather2), ACCURACY);
83 }
84
85 @Test
86 public void testWindSpeed()
87 {
88     CycleRide cycle1 = new CycleRide(10, true);
89     Cyclist cyclist1 = new Cyclist("Beginner", 0);
90     Weather weather = new Weather(15,10,false);
91     assertEquals("Test windspeed at 10mph failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);
92
93     CycleRide cycle2 = new CycleRide(9, true);
94     Cyclist cyclist2 = new Cyclist("Beginner", 0);
95     Weather weather2 = new Weather(15,20,false);
96     assertEquals("Test windspeed at 20mph failed:", 1.0,
cycleCalculator.getDuration(cycle2, cyclist2, weather2), ACCURACY);
97
98     CycleRide cycle3 = new CycleRide(6, true);
99     Cyclist cyclist3 = new Cyclist("Beginner", 0);
100    Weather weather3 = new Weather(15,70,false);
101    assertEquals("Test windspeed at 70mph failed:", 1.0,
cycleCalculator.getDuration(cycle3, cyclist3, weather3), ACCURACY);

```

```
102     }
103
104     @Test
105     public void testRaining()
106     {
107         CycleRide cycle1 = new CycleRide(8, true);
108         Cyclist cyclist1 = new Cyclist("Beginner", 0);
109         Weather weather = new Weather(15,0,true);
110         assertEquals("Test raining failed:", 1.0, cycleCalculator.
        getDuration(cycle1, cyclist1, weather), ACCURACY);
111
112         CycleRide cycle2 = new CycleRide(10, true);
113         Cyclist cyclist2 = new Cyclist("Beginner", 0);
114         Weather weather2 = new Weather(15,0,false);
115         assertEquals("Test not raining failed:", 1.0, cycleCalculator
        .getDuration(cycle2, cyclist2, weather2), ACCURACY);
116     }
117
118 }
119
```