```java
import static org.junit.Assert.*;

import org.junit.Before;
import org.junit.Test;

public class CycleCalculatorTest
{
    private static final double ACCURACY = 0.001;

    private CycleCalculator cycleCalculator;

    @Before
    public void setup()
    {
        cycleCalculator = new CycleCalculator();
    }

    @Test
    public void testCompetency()
    {
        CycleRide cycle1 = new CycleRide(10, true);
        Cyclist cyclist1 = new Cyclist("Beginner", 0);
        Weather weather = new Weather(15,0,false);
        assertEquals("Beginner competency level failed:", 1.0, cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);

        CycleRide cycle2 = new CycleRide(15, true);
        Cyclist cyclist2 = new Cyclist("Intermediate", 0);
        assertEquals("Intermediate competency level failed:", 1.0, cycleCalculator.getDuration(cycle2, cyclist2, weather), ACCURACY);

        CycleRide cycle3 = new CycleRide(20, true);
        Cyclist cyclist3 = new Cyclist("Advanced", 0);
        assertEquals("Advanced competency level failed:", 1.0, cycleCalculator.getDuration(cycle3, cyclist3, weather), ACCURACY);
    }

    @Test
    public void testCycleBuddy()
    {
        CycleRide cycle1 = new CycleRide(12, false);
        Cyclist cyclist1 = new Cyclist("Beginner", 0);
        Weather weather = new Weather(15,0,false);
        assertEquals("Test cycling with a buddy failed:", 1.0, cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);

        CycleRide cycle2 = new CycleRide(10, true);
        Cyclist cyclist2 = new Cyclist("Beginner", 0);
        assertEquals("Test cycling alone failed:", 1.0, cycleCalculator.getDuration(cycle2, cyclist2, weather), ACCURACY);
    }

    @Test
    public void testYearsExperience()
    {
        CycleRide cycle1 = new CycleRide(11, true);
        Cyclist cyclist1 = new Cyclist("Beginner", 5);
        Weather weather = new Weather(15,0,false);
        assertEquals("Test years experience failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);
    }

    @Test
    public void testLessThan10()
    {
        CycleRide cycle1 = new CycleRide(9, true);
        Cyclist cyclist1 = new Cyclist("Beginner", 0);
        Weather weather = new Weather(0,0,false);
        assertEquals("Test temperature less than 10 failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);

        CycleRide cycle2 = new CycleRide(10, true);
        Cyclist cyclist2 = new Cyclist("Beginner", 0);
        Weather weather2 = new Weather(15,0,false);
        assertEquals("Test temperature more than 10 failed:", 1.0,
cycleCalculator.getDuration(cycle2, cyclist2, weather2), ACCURACY);
    }

    @Test
    public void testMoreThan20()
    {
        CycleRide cycle1 = new CycleRide(9, true);
        Cyclist cyclist1 = new Cyclist("Beginner", 0);
        Weather weather = new Weather(30,0,false);
        assertEquals("Test temperature more than 20 failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);

        CycleRide cycle2 = new CycleRide(10, true);
        Cyclist cyclist2 = new Cyclist("Beginner", 0);
        Weather weather2 = new Weather(15,0,false);
        assertEquals("Test temperature less than 20 failed:", 1.0,
cycleCalculator.getDuration(cycle2, cyclist2, weather2), ACCURACY);
    }

    @Test
    public void testWindSpeed()
    {
        CycleRide cycle1 = new CycleRide(10, true);
        Cyclist cyclist1 = new Cyclist("Beginner", 0);
        Weather weather = new Weather(15,10,false);
        assertEquals("Test windspeed at 10mph failed:", 1.0,
cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);

        CycleRide cycle2 = new CycleRide(9, true);
        Cyclist cyclist2 = new Cyclist("Beginner", 0);
        Weather weather2 = new Weather(15,20,false);
        assertEquals("Test windspeed at 20mph failed:", 1.0,
cycleCalculator.getDuration(cycle2, cyclist2, weather2), ACCURACY);

        CycleRide cycle3 = new CycleRide(6, true);
        Cyclist cyclist3 = new Cyclist("Beginner", 0);
        Weather weather3 = new Weather(15,70,false);
        assertEquals("Test windspeed at 70mph failed:", 1.0,
cycleCalculator.getDuration(cycle3, cyclist3, weather3), ACCURACY);
```

```
    }

    @Test
    public void testRaining()
    {
        CycleRide cycle1 = new CycleRide(8, true);
        Cyclist cyclist1 = new Cyclist("Beginner", 0);
        Weather weather = new Weather(15,0,true);
        assertEquals("Test raining failed:", 1.0, cycleCalculator.getDuration(cycle1, cyclist1, weather), ACCURACY);

        CycleRide cycle2 = new CycleRide(10, true);
        Cyclist cyclist2 = new Cyclist("Beginner", 0);
        Weather weather2 = new Weather(15,0,false);
        assertEquals("Test not raining failed:", 1.0, cycleCalculator.getDuration(cycle2, cyclist2, weather2), ACCURACY);
    }

}

```