Exercise 1: Setting Up JUnit

JUnit Dependency (For Maven):Pom.xml

```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>4.13.2</version>
  <scope>test</scope>
</dependency>
```

Code:

```
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }
}
import org.junit.Test;
import static org.junit.Assert.assertEquals;

public class CalculatorTest {
    @Test
    public void testAdd() {
        Calculator calc = new Calculator();
        int result = calc.add(3, 4);
        assertEquals(7, result);
    }
}
```

Output:

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running CalculatorTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESSFUL
```

Exercise 3: Assertions in JUnit

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

public class AssertionsTest {
    @Test
    public void testAssertions() {
        assertEquals(5, 2 + 3);
        assertTrue(5 > 3);
        assertFalse(5 < 3);
        assertNull(null);
        assertNotNull(new Object());
    }
}
```

Output:

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running AssertionsTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESSFUL
```

Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit

```
import org.junit.Before;
import org.junit.After;
import org.junit.Test;
import static org.junit.Assert.assertEquals;

public class CalculatorTest {
    private Calculator calculator;

    @Before
    public void setUp() {
        calculator = new Calculator();
    }

    @After
    public void tearDown() {
        calculator = null;
    }

    @Test
    public void testAdd() {
        int a = 10;
        int b = 5;
        int result = calculator.add(a, b);
        assertEquals(15, result);
    }

    @Test
    public void testSubtract() {
        int result = calculator.subtract(10, 5);
        assertEquals(5, result);
    }
}

public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }

    public int subtract(int a, int b) {
        return a - b;
    }
}
```

Output:

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running CalculatorTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
BUILD SUCCESSFUL
```