Exercise 1: Setting Up Junit

```
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertTrue;

public class JUnit_Setup {

    @Test
    public void testEnvironmentSetup() {
        assertTrue(true, "JUnit5 setup is working fine!");
    }
}
```

Output:

```
> Task :test

JUnit5 setup is working fine!
BUILD SUCCESSFUL in 1s
1 test passed
```

Exercise 3: Assertions in Junit

```
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class JUnit_Assertions {

    @Test
```

```
    public void testEqual() {
        assertEquals(5, 2 + 3);
    }

    @Test
    public void testTrue() {
        assertTrue(10 > 5);
    }

    @Test
    public void testNotNull() {
        String str = "Hello";
        assertNotNull(str);
    }
}
```

Output:

```
> Task :test

All assertions passed.

BUILD SUCCESSFUL in 1s

3 tests passed
```

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

```
import org.junit.jupiter.api.Test;
```

```
import static org.junit.jupiter.api.Assertions.*;

public class JUnit_AAA_ExceptionTesting {

    @Test
    public void testDivideByZeroException() {
        int a = 10;
        int b = 0;

        assertThrows(ArithmeticException.class, () -> {
            int c = a / b;
        });
    }

    @Test
    public void testAAAStyle() {
        int num = 5;
        int result = num * 2;
        assertEquals(10, result);
    }
}
```

Output:

```
> Task :test

Test AAA style passed.
```

Divide by zero exception caught as expected.

BUILD SUCCESSFUL in 1s

2 tests passed