Exercise 1: Setting Up JUnit

---

1. Creating New Java Project

I openned my vs code and made a new folder called `JUnitSetUpTest` and created the files inside it. This is where I wrote my java code and test class.

---

2. Add JUnit Dependancy

Since the question gave JUnit 4.13.2, I used that in `pom.xml` file because it’s a maven project. I did this:

```
<!-- pom.xml -->
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
         http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>
  <artifactId>JUnitSetUpTest</artifactId>
  <version>1.0-SNAPSHOT</version>

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

I saved the file and VS Code auto downloaded the dependancy.

3. Create a New Test Class

I created a very basic Java class first. It just adds two numbers. Then I created a test class to test it.

Calculator.java (inside `src/main/java/`)

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

CalculatorTest.java (inside `src/test/java/`)

```
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(2, 3);
        assertEquals(5, result);
    }
}
```

How I Ran It

In terminal I typed:

```
mvn test
```

and it showed me something like this:

Output:

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