# Exercise 1: Setting Up JUnit

## Objective

To set up JUnit in a Java project and write a simple unit test using JUnit 5.

## Step-by-Step Instructions

## Step 1: Set Up Your Java Project

You can use:  
- An IDE (like IntelliJ IDEA or Eclipse)  
- Or the Command Line  
- Optionally with a build tool like Maven or Gradle

## Step 2: Add JUnit 5 to Your Project

If Using Maven:  
Add the following to your pom.xml:  
  
<dependencies>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter</artifactId>  
 <version>5.10.0</version>  
 <scope>test</scope>  
 </dependency>  
</dependencies>  
  
If Using Gradle:  
Add this to your build.gradle:  
  
dependencies {  
 testImplementation 'org.junit.jupiter:junit-jupiter:5.10.0'  
}  
test {  
 useJUnitPlatform()  
}  
  
If Not Using Maven or Gradle:  
1. Download JUnit JARs from https://junit.org  
2. Add the .jar files to your project’s classpath

## Step 3: Write the Class to be Tested

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

## Step 4: Write the JUnit Test Class

// CalculatorTest.java  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class CalculatorTest {  
  
 @Test  
 void testAdd() {  
 Calculator calc = new Calculator();  
 assertEquals(5, calc.add(2, 3), "2 + 3 should equal 5");  
 }  
}

## Step 5: Run the Test

In IntelliJ or Eclipse:  
- Right-click the test method or class → Run  
- Or use the test runner window  
  
Using Maven:  
$ mvn test  
  
Using Gradle:  
$ gradle test

## Output

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0  
BUILD SUCCESS

## Notes

- Use @BeforeEach for setup and @AfterEach for cleanup if needed.  
- Use more assertions like assertTrue(), assertNotNull(), assertThrows() as you expand tests.