Setting Up JUnit

# Goal:

Set up JUnit 4 in Eclipse for unit testing a Java Maven project.

# Prerequisites:

- Eclipse IDE  
- Java JDK (8 or later)  
- Maven support in Eclipse

# Step 1: Create a Maven Project in Eclipse

1. Open Eclipse.  
2. Go to File > New > Project… > Maven > Maven Project > Next.  
3. Select 'Use default Workspace location' and click Next.  
4. Choose 'maven-archetype-quickstart' and click Next.  
5. Group Id: com.example  
6. Artifact Id: JUnitTestDemo  
7. Click Finish.

# Step 2: Add JUnit Dependency to pom.xml

1. Open pom.xml from the project.  
2. Add the following inside the <dependencies> tag:  
  
<dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.13.2</version>  
 <scope>test</scope>  
 </dependency>  
</dependencies>  
  
3. Save and right-click project > Maven > Update Project.

# Step 3: Create a Java Class to Test

1. Right-click src/main/java > com.example > New > Class  
2. Name it Calculator and click Finish.  
3. Paste this code:  
  
package com.example;  
  
public class Calculator {  
 public int add(int a, int b) {  
 return a + b;  
 }  
}

# Step 4: Create a JUnit Test Class

1. Right-click src/test/java > com.example > New > JUnit Test Case  
2. Name: CalculatorTest, Select JUnit 4, Click Finish  
 OR create a class manually with this code:  
  
package com.example;  
  
import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class CalculatorTest {  
 @Test  
 public void testAdd() {  
 Calculator calc = new Calculator();  
 int result = calc.add(2, 3);  
 assertEquals(5, result);  
 }  
}

# Step 5: Run the JUnit Test

1. Right-click CalculatorTest.java  
2. Click Run As > JUnit Test  
  
✅ Green bar = test passed  
❌ Red bar = test failed

# Optional Tips:

- Add more @Test methods to test other behaviors  
- Use @Before and @After for setup/cleanup  
- Use @Test(expected = Exception.class) to test exceptions