

Exercise 1: Setting Up Junit

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 https://maven.apache.org/xsd/maven-
4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.example</groupId>

  <artifactId>JUnitSetup</artifactId>

  <version>0.0.1-SNAPSHOT</version>

  <name>JUnitSetup</name>

  <dependencies>

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.13.2</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <build>

    <plugins>

      <plugin>

        <groupId>org.apache.maven.plugins</groupId>

        <artifactId>maven-compiler-plugin</artifactId>

        <version>3.11.0</version>

        <configuration>

          <source>11</source>

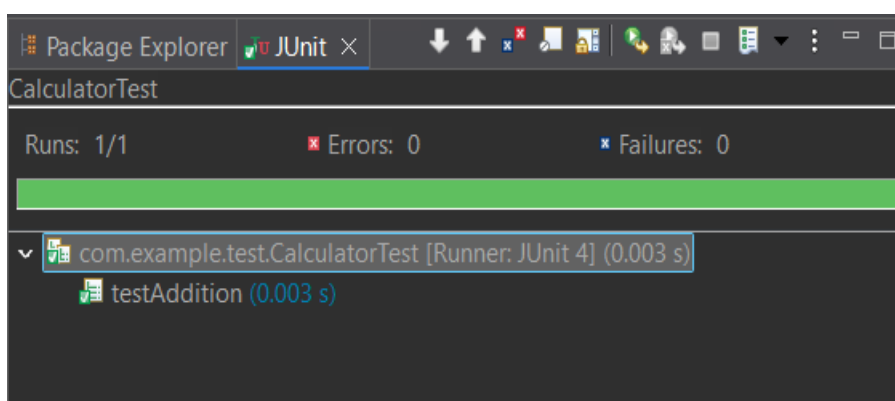
          <target>11</target>
```

```
</configuration>
</plugin>
</plugins>
</build>
</project>
```

CalculatorTest.java

```
package com.example.test;
import org.junit.Test;
import static org.junit.Assert.*;
public class CalculatorTest {
    @Test
    public void testAddition() {
        int result = 2 + 3;
        assertEquals(5, result);
    }
}
```

Output :



Exercise 3: Assertions in JUnit

AssertionsTest.java

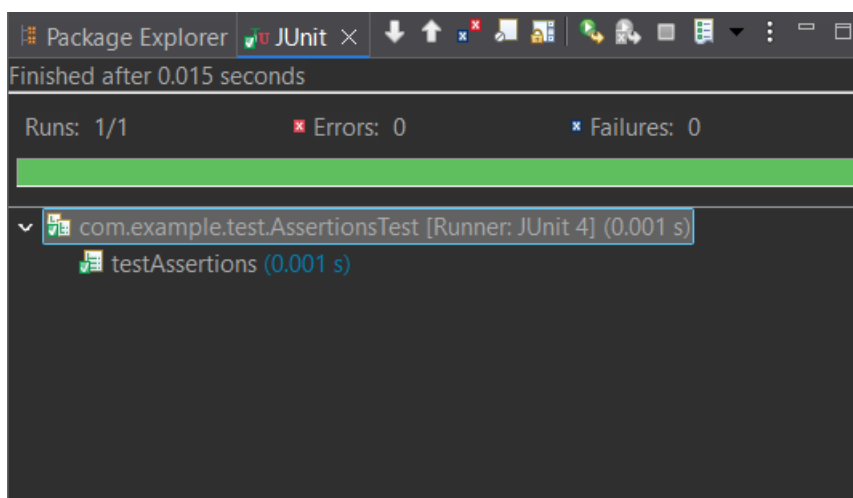
```
package com.example.test;

import static org.junit.Assert.*;

import org.junit.Test;

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

Output :



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

Calculator.java

```
package com.example;

public class Calculator {

    public int add(int a, int b) {

        return a + b;

    }

    public int subtract(int a, int b) {

        return a - b;

    }

}
```

CalculatorTest.java

```
package com.example.test;

import com.example.Calculator;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;

public class CalculatorTest

{

    private Calculator calculator;

    @Before

    public void setUp()

    {

        calculator = new Calculator();

    }

}
```

```

        System.out.println("Setup: Calculator created.");
    }

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

        System.out.println("Teardown: Calculator cleaned up.");
    }

    @Test
    public void testAdd()
    {
        int result = calculator.add(2, 3);

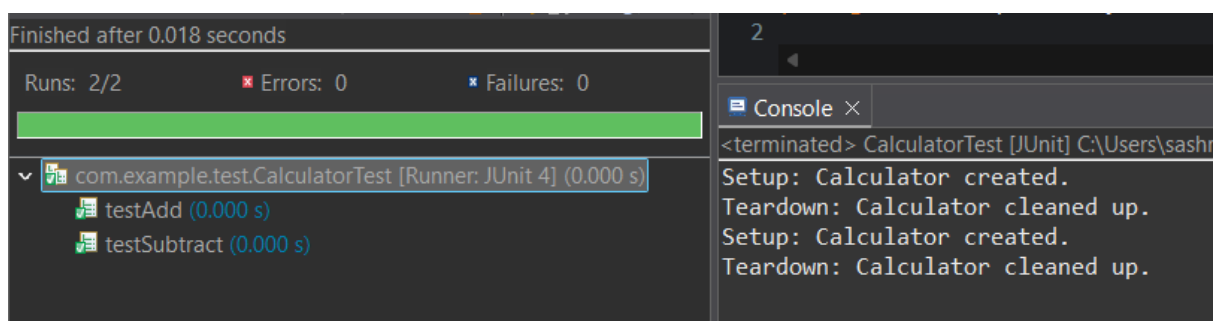
        assertEquals(5, result);
    }

    @Test
    public void testSubtract()
    {
        int result = calculator.subtract(10, 4);

        assertEquals(6, result);
    }
}

```

Output :



The screenshot displays the output of a Java test suite. On the left, a test runner window shows the results for `com.example.test.CalculatorTest`. It indicates that 2 out of 2 runs were successful, with 0 errors and 0 failures. A green progress bar is shown. Below this, the test methods `testAdd` and `testSubtract` are listed, both passing successfully in 0.000 seconds.

On the right, a console window shows the output of the tests. It displays the following messages:

```

<terminated> CalculatorTest [JUnit] C:\Users\sashr
Setup: Calculator created.
Teardown: Calculator cleaned up.
Setup: Calculator created.
Teardown: Calculator cleaned up.

```