Experiment 10: Unit Testing with JUnit

Aim:

To perform unit testing of a Java class using the JUnit framework.

Objective:

- Understand how JUnit helps in testing individual units of code.
- Learn to write and run basic test cases using JUnit.

Tool Used:

- Java
- Eclipse or IntelliJ IDE
- JUnit Library

Procedure:

1. **Open your IDE** (Eclipse/IntelliJ).
2. **Create a Java class** named Calculator.java with the following code:

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

    public int subtract(int a, int b) {
        return a - b;
    }
}
```

3. **Create a JUnit Test Class** named CalculatorTest.java:

```
import org.junit.Test;
import static org.junit.Assert.*;

public class CalculatorTest {
    Calculator calc = new Calculator();

    @Test
    public void testAdd() {
```

```
        assertEquals(5, calc.add(2, 3));
    }

    @Test
    public void testSubtract() {
        assertEquals(2, calc.subtract(5, 3));
    }
}
```

4. **Run the test**:
    - Right-click on the test class → Run As → JUnit Test.
    - View the result in the JUnit pane (green = pass, red = fail).

Result:

All unit test cases passed successfully, confirming the logic of the Calculator class.

Conclusion:

JUnit helps test individual methods in isolation and ensures that the program behaves as expected. It is simple, fast, and effective for unit testing in Java projects.