**JUnit Testing Exercises**

**Exercise 1: Setting Up Junit**

// Calculator.java

**package** app;

**public** **class** Calculator {

**public** **int** add(**int** a, **int** b) {

**return** a + b;

}

}

// CalculatorTest.java

**package** app;

**import** org.junit.jupiter.api.Test;

**import** **static** org.junit.jupiter.api.Assert.\*;

**public** **class** CalculatorTest {

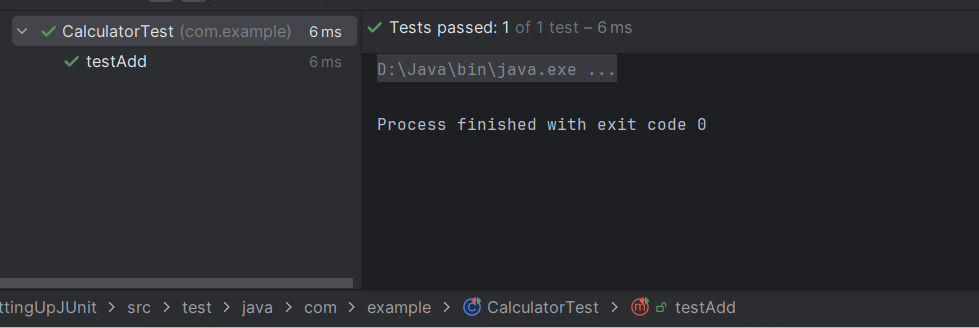
@Test

**public** **int** testAdd(**int** a, **int** b) {

**return** a + b;}

}

**Output :**



**Exercise 3: Assertions in Junit**

// AssertionsTest.java

**package** app;

**import** org.junit.jupiter.api.Test;

**import** **static** org.junit.jupiter.api.Assert.\*;

**public** **class** AssertionsTest {

@Test

**public** **void** testAssertions() {

assertEquals(10, 6 + 4);// Equals

assertTrue(28 > 10);// True

assertFalse(5 < 4);// False

String S= **null**;

assertNull(S);// Null

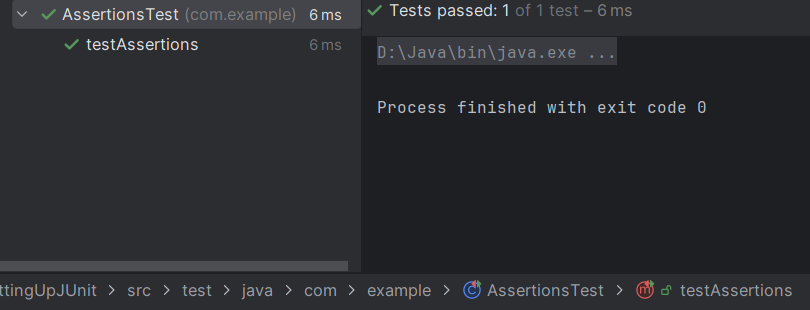
String St="Test";

assertNotNull(St);// Not Null

}

}

**Output:**



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

//Calculator1.java

**package app;**

**public class Calculator1 {**

**public int add(int a, int b) {**

**return a + b;**

**}**

**public int multiply(int a, int b) {**

**return a \* b;**

**}**

**}**

// CalculatorTest1.java

**package** app;

**import** org.junit.Before;

**import** org.junit.After;

**import** org.junit.Test;

**import** **static** org.junit.Assert.\*;

**public** **class** CalculatorTest1 {

**private** Calculator1 calculator;

@Before

**public** **void** setUp() {

calculator = **new** Calculator1();

System.***out***.println("Setup before each test.");

}

@After

**public** **void** tearDown() {

calculator = **null**;

System.***out***.println("Teardown after each test.");

}

@Test

**public** **void** testMultiplication() {

**int** result = calculator.multiply(10, 4);

assertEquals(40, result);

}

@Test

**public** **void** testAddition() {

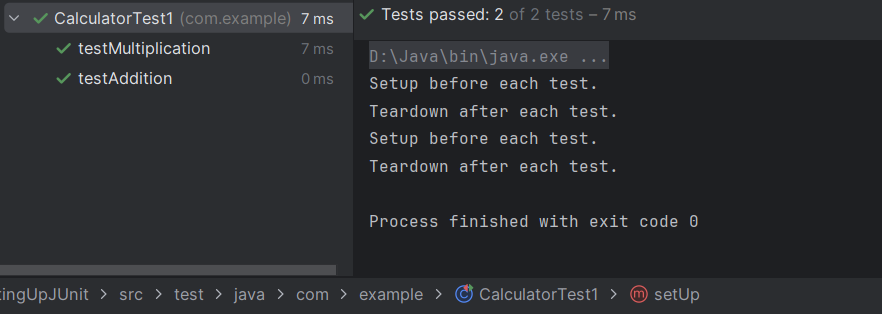
**int** result = calculator.add(10, 28);

assertEquals(38, result);

}

}

**Output:**



**Mockito**

**Exercise 1: Mocking and Stubbing**

//ExternalApi.java

**package** com.example.api;

**public** **interface** ExternalApi {

String getData();

}

// MyService.java

**package** com.example;

**public** **class** MyService {

**private** ExternalAPI api;

**public** MyService(ExternalAPI api) {

**this**.api = api;

}

**public** String fetchData() {

**return** api.getData();

}

}

//MyServiceTest.java

**package** com.example.api;

**import** **static** org.junit.jupiter.api.Assertions.assertEquals;

**import** **static** org.mockito.Mockito.\*;

**import** com.example.api.ExternalApi;

**import** org.junit.jupiter.api.Test;

**import** org.mockito.Mockito;

**public** **class** MyServiceTest {

@Test

**public** **void** testExternalApi() {

ExternalApi mockApi = Mockito.mock(ExternalApi.**class**);

when(mockApi.getData()).thenReturn("Mock Data");

MyService service = **new** MyService(mockApi);

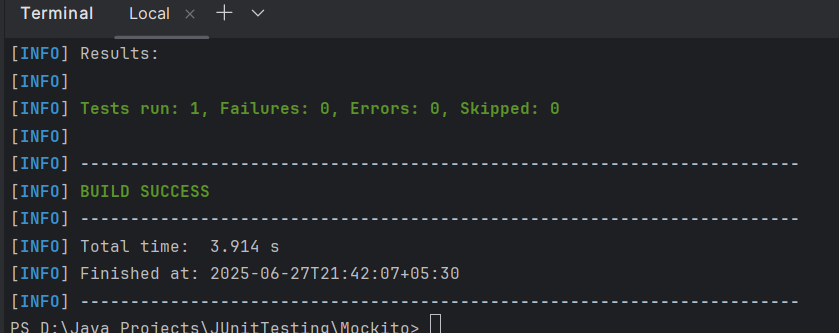
String result = service.fetchData();

assertEquals("Mock Data", result);

}

}

**Output:**

****

**Exercise 3: Argument Matching**

// ExternalApi.java

**package** com.example;

**public** **interface** ExternalAPI {

String getData();

}

// MyService.java

**package** com.example;

**public** **class** MyService {

**private** ExternalAPI api;

**public** MyService(ExternalAPI api) {

**this**.api = api;

}

**public** String fetchData() {

**return** api.getData();

}

}

//MyServiceTest.java

**package** com.example;

**import** org.junit.jupiter.api.Test;

**import** **static** org.mockito.Mockito.\*;

**public** **class** MyServiceTest {

@Test

**public** **void** testVerifyInteraction() {

ExternalAPI mockApi = mock(ExternalAPI.**class**);

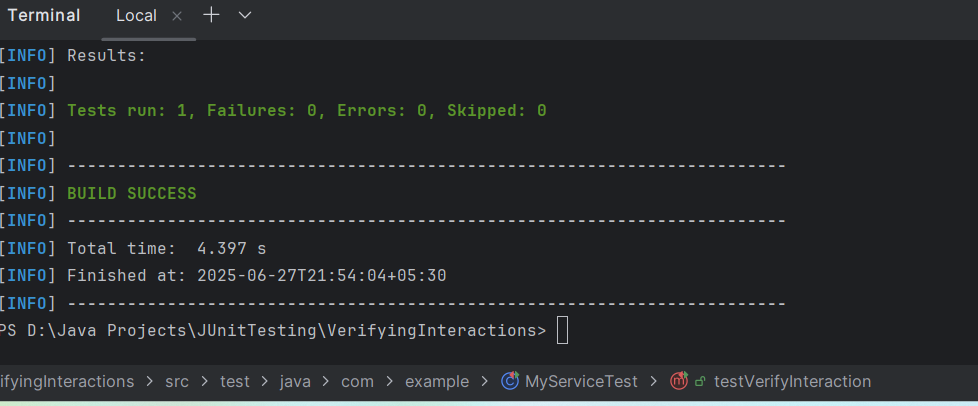
MyService service = **new** MyService(mockApi);

service.fetchData();

verify(mockApi).getData();}

}

**Output:**



**Logging using SLF4J**

**Exercise 1: Logging Error Messages and Warning Levels**

**//**LoggingExample.java

**package** com.example;

**import** org.slf4j.Logger;

**import** org.slf4j.LoggerFactory;

**public** **class** LoggingExample {

**private** **static** **final** Logger ***logger*** = LoggerFactory.getLogger(LoggingExample.**class**);

**public** **static** **void** main(String[] args) {

***logger***.error("This is an error message");

***logger***.warn("This is a warning message");

}

}

**Output:**

