**DIGITAL\_NURTURE\_4.0**

**Superset ID – 6404833**

**Junit\_Basic\_Testing\_Exercise**

**Exercise 1: Setting Up Junit**  
package com.example.junit\_exercise;

import static org.junit.jupiter.api.Assertions.assertTrue;

import org.junit.jupiter.api.Test;

/\*\*

\* Unit test for simple App.

\*/

public class AppTest {

/\*\*

\* Rigorous Test :-)

\*/

@Test

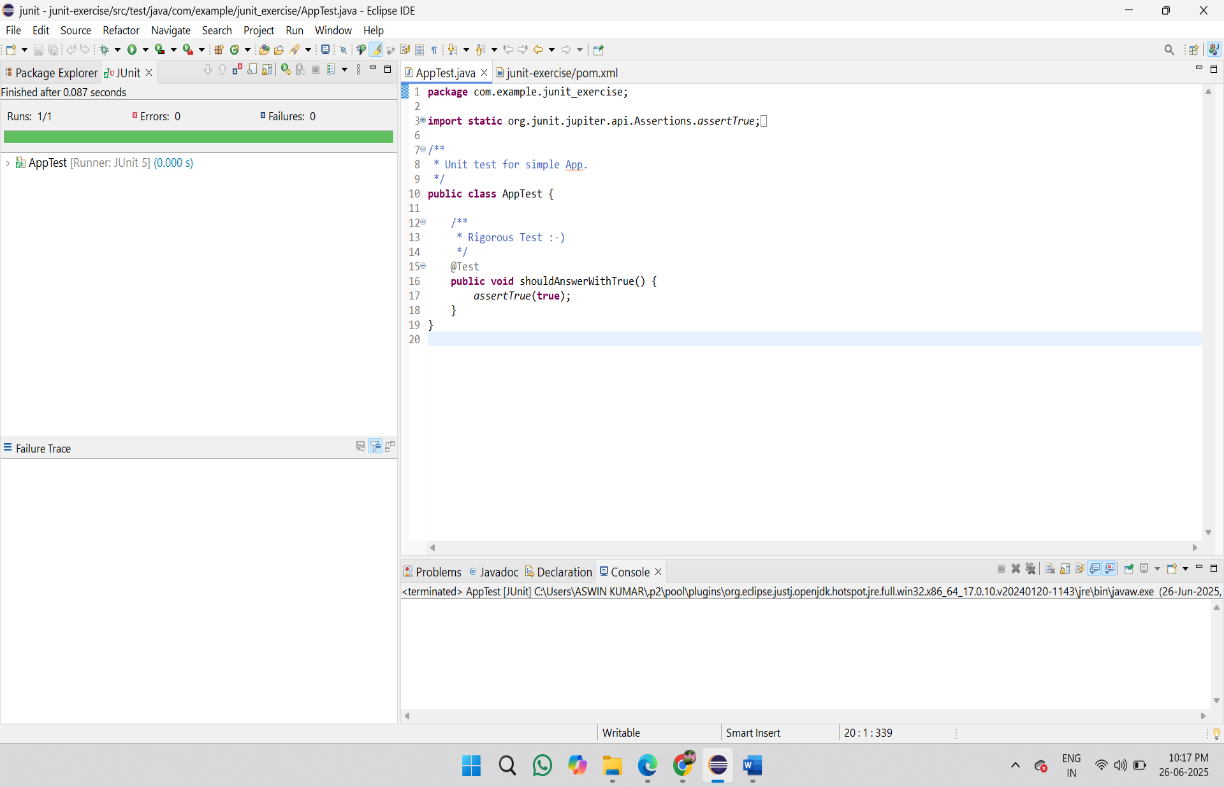
public void shouldAnswerWithTrue() {

assertTrue(true);

}

}

**OUTPUT**



**Exercise 3: Assertions in Junit**package com.example.calculator;

import org.junit.jupiter.api.Test;

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

public class AssertionsTest {

@Test

public void testEquals(){

*assertEquals*(10, 5 + 5);

}

@Test

public void testTrueFalse(){

*assertTrue*(7 > 3);

*assertFalse*(2 > 5); }

@Test

public void testNulls(){

Object obj1 = null;

Object obj2 = new Object();

*assertNull*(obj1);

*assertNotNull*(obj2); }

@Test

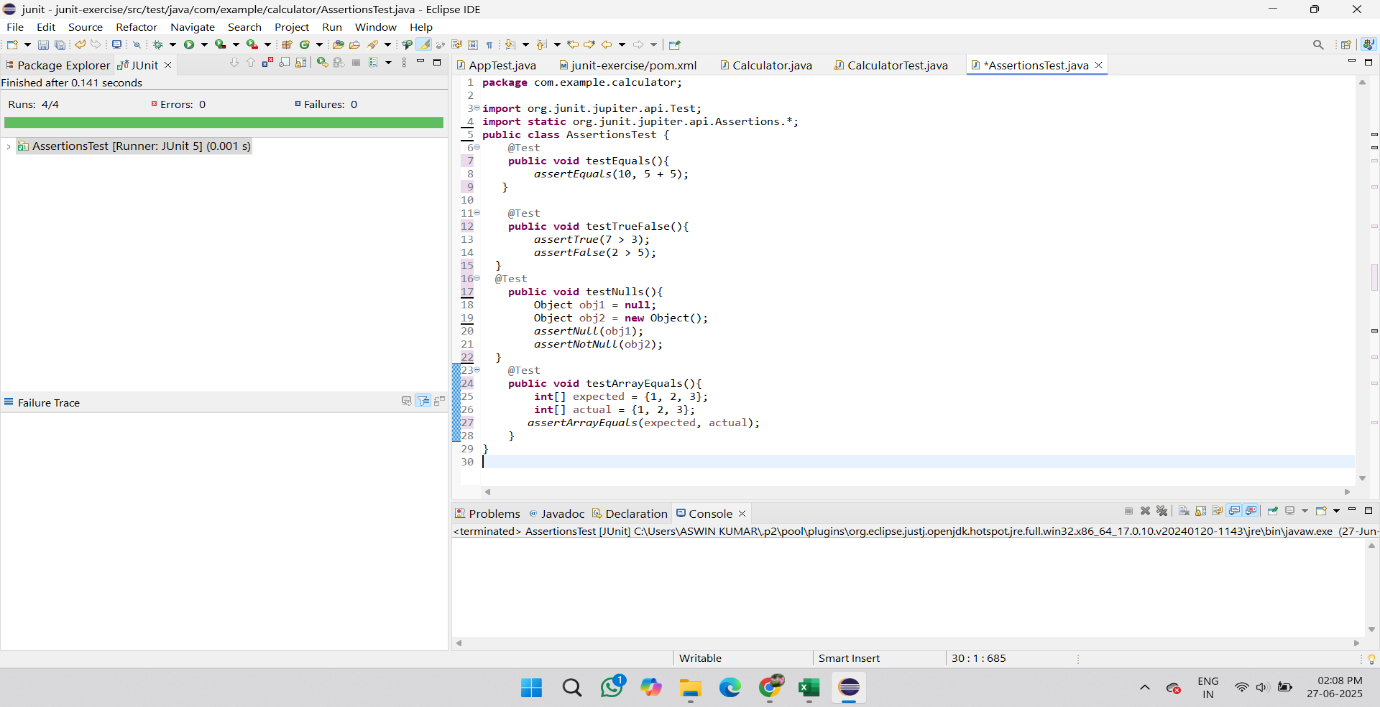
public void testArrayEquals(){

int[] expected = {1, 2, 3};

int[] actual = {1, 2, 3};

*assertArrayEquals*(expected, actual);

}}

**Output**

**Exercise 4: AAA Pattern, Setup, and Teardown**

package com.example.calculator;

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

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

public class Setupteardowntest {

private Calculator calc;

@BeforeEach

public void setUp(){

System.*out*.println("Setup...");

calc = new Calculator();

}

@AfterEach

public void tearDown(){

System.*out*.println("Teardown...");

calc = null;

}

@Test

public void testMultiply(){

int result = calc.multiply(3, 4);

*assertEquals*(12, result);

}

@Test

public void testDivide(){

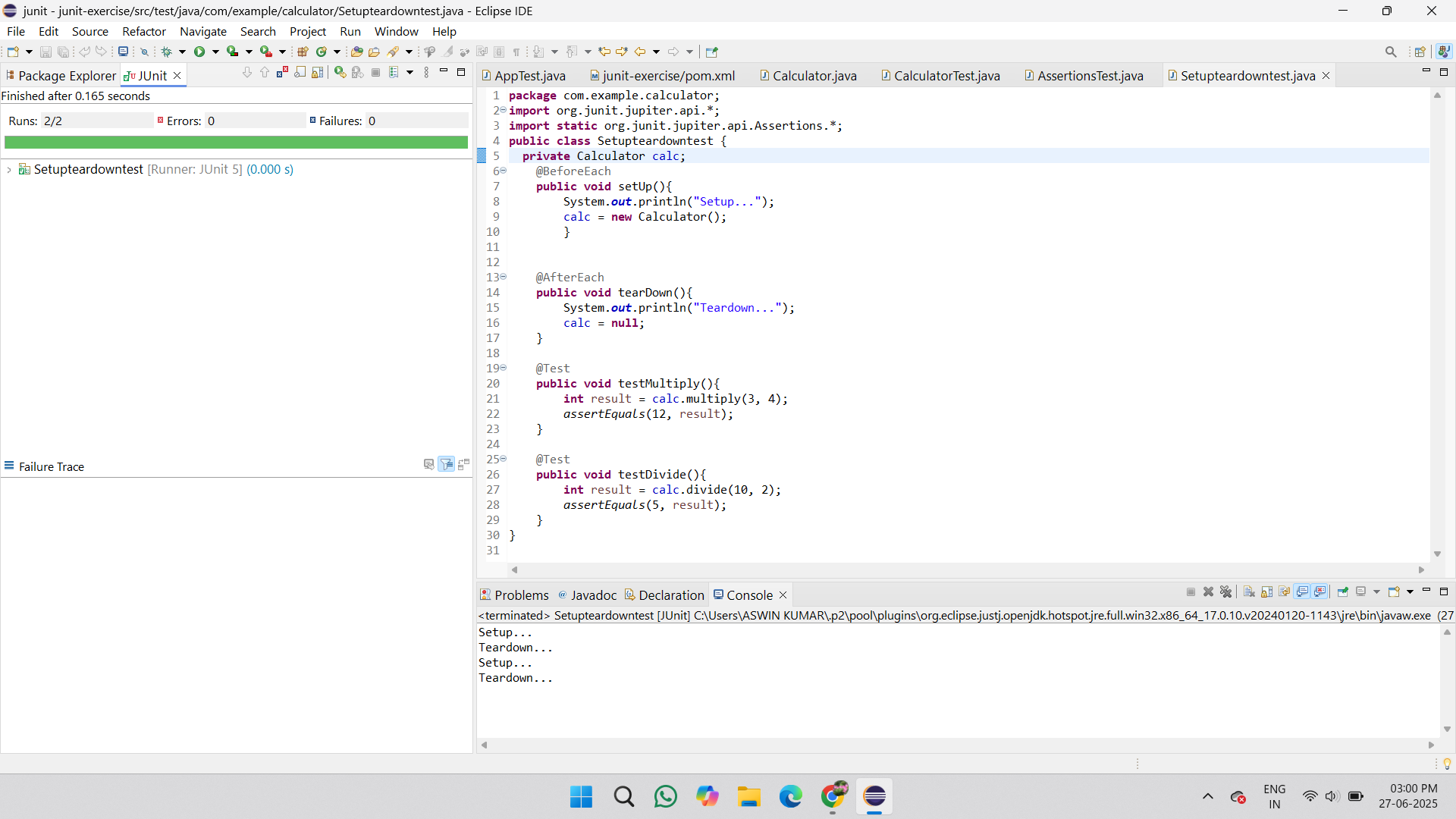
int result = calc.divide(10, 2);

*assertEquals*(5, result);

}

}

**OUTPUT**



**Mockito Exercises**

**Exercise 1: Mocking and Stubbing (Mockito)**

**Externalapi.java:**

package com.example.mockito;

public interface Externalapi {

String getData();

}

**Myservice.java:**

package com.example.mockito;

public class Myservice {

private Externalapi api;

public Myservice(Externalapi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**Myservicetest.java:**

package com.example.mockito;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

public class Myservicetest {

@Test

public void testExternalApi() {

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

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

Myservice service = new Myservice(mockApi);

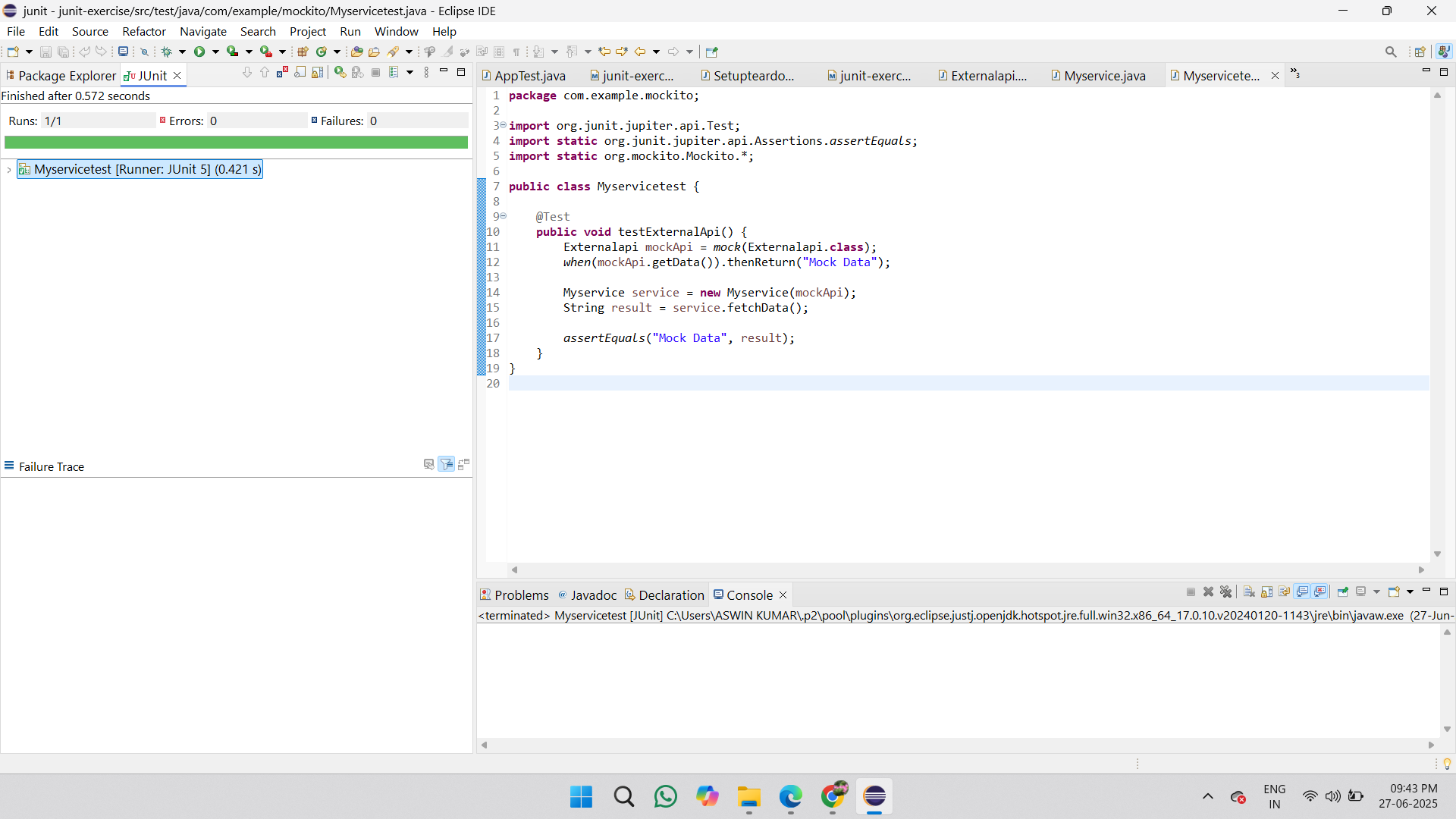
String result = service.fetchData();

*assertEquals*("Mock Data", result);

}

}

**Output**



**Exercise 2: Verifying Interactions (Mockito)**

**Myservicetest.java:**

package com.example.mockito;

import org.junit.jupiter.api.Test;

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

import static org.mockito.Mockito.\*;

public class Myservicetest {

@Test

public void testExternalApi() {

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

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

Myservice service = new Myservice(mockApi);

String result = service.fetchData();

*assertEquals*("Mock Data", result);

}

@Test

public void testVerifyInteraction() {

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

Myservice service = new Myservice(mockApi);

service.fetchData();

*verify*(mockApi).getData();

}

}

**OUTPUT**

