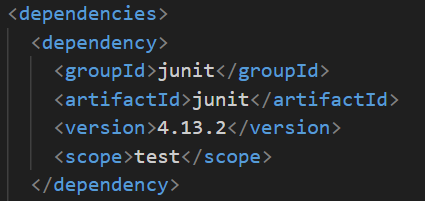
**JUnit Hands-On Exercises:**

**Exercise 1: Setting Up Junit**

**ANSWER:**

**PROGRAM:**

//Dependencies



//InterestCalculator.java

package com.week2.bankapp;

public class InterestCalculator {

    public double applyMonthlyInterest(double balance) {

        return balance \* 1.01;

    }

}

//InterestCalculatorTest.java

package com.week2.bankapp;

import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class InterestCalculatorTest {

     @Test

    public void testApplyMonthlyInterest() {

        InterestCalculator calc = new InterestCalculator();

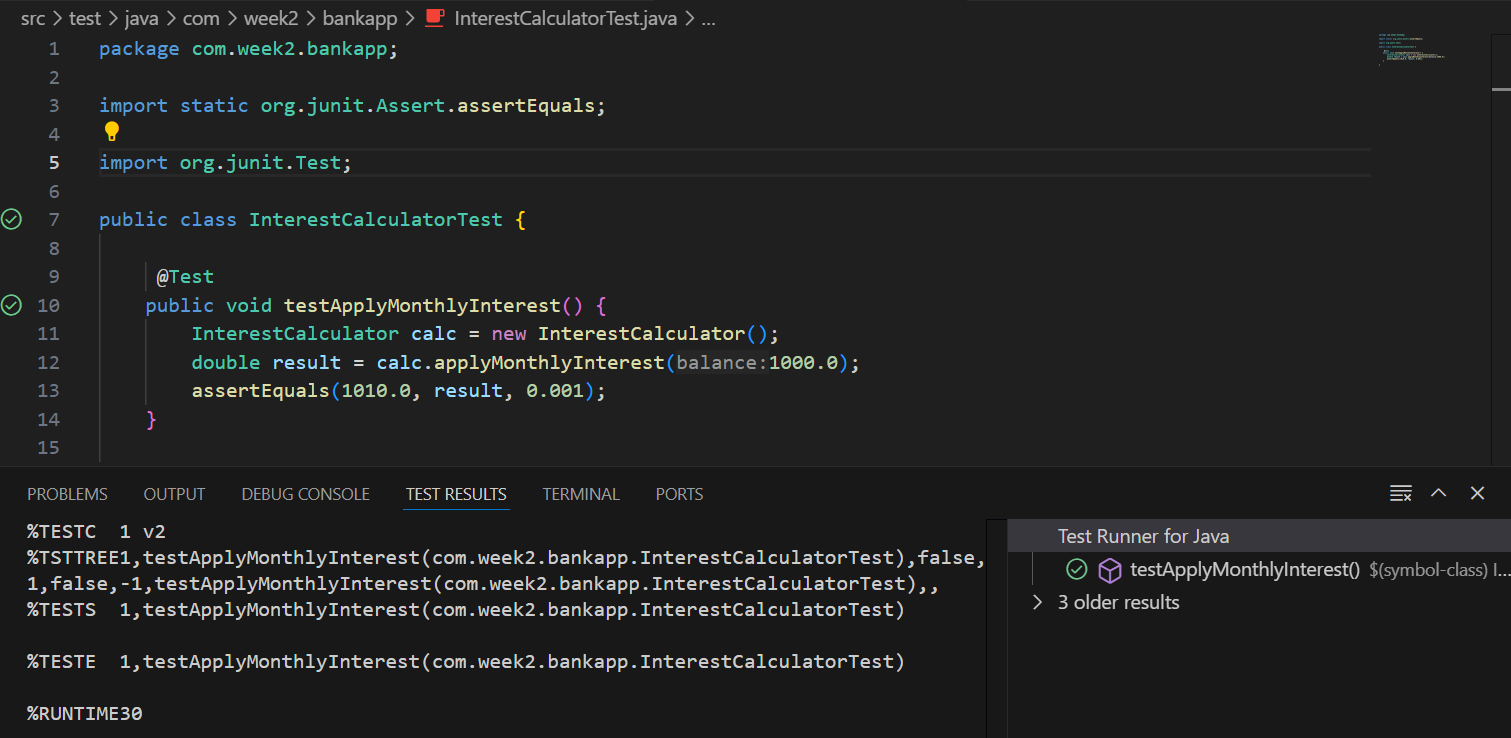
        double result = calc.applyMonthlyInterest(1000.0);

        assertEquals(1010.0, result, 0.001);

    }

}

**OUTPUT:**



**Exercise 3: Assertions in Junit**

**Answer:**

**PROGRAM:**

//AssertionTest.java

package AssertionTest;

import static org.junit.Assert.assertEquals;

import static org.junit.Assert.assertFalse;

import static org.junit.Assert.assertNotNull;

import static org.junit.Assert.assertNull;

import static org.junit.Assert.assertTrue;

import org.junit.Test;

public class AssertionTest {

    @Test

    public void testAssertions() {

        assertEquals(5, 2 + 3);

        assertTrue(5 > 3);

        assertFalse(5 < 3);

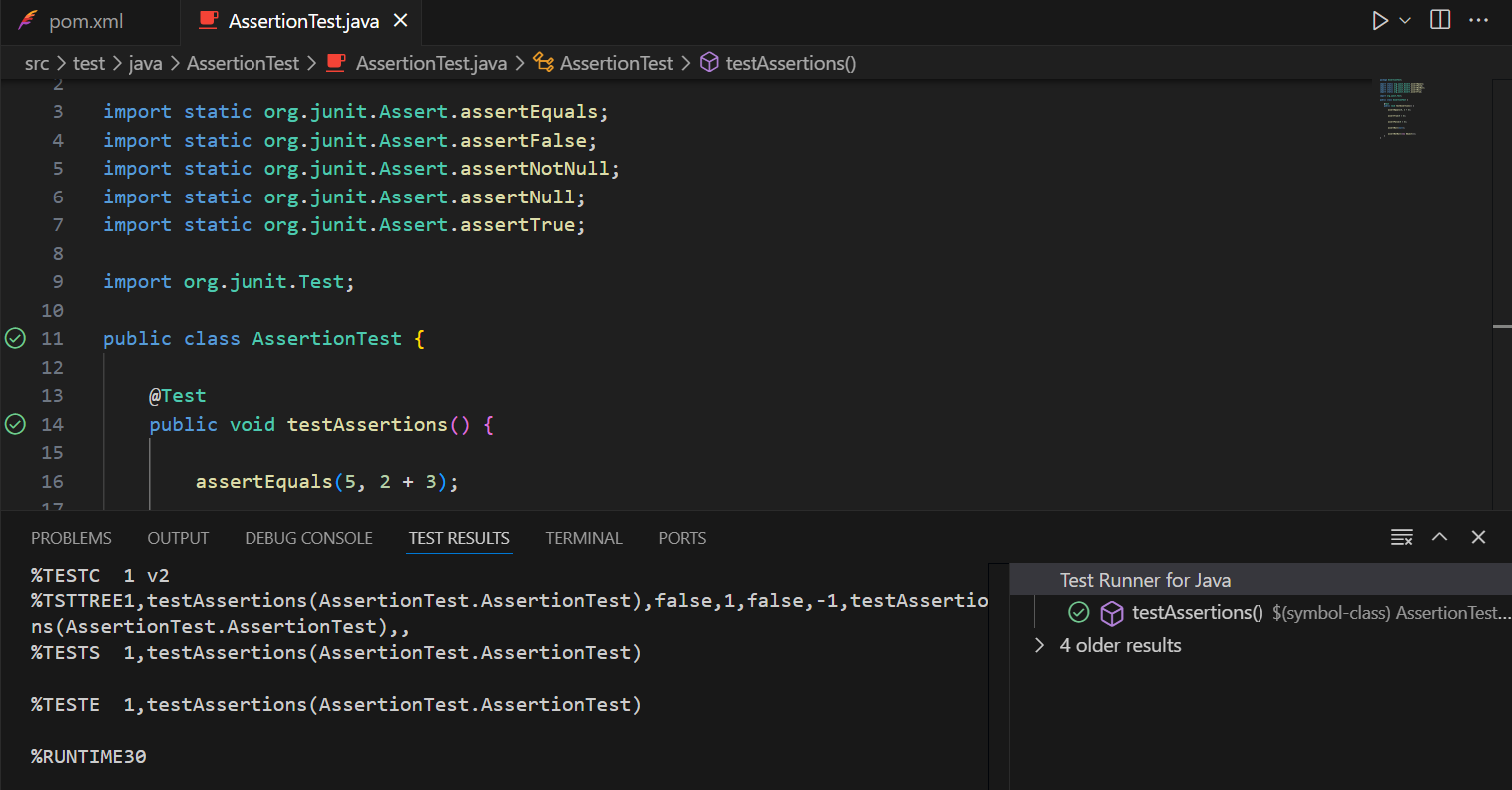
        assertNull(null);

        assertNotNull(new Object());

    }

}

**OUTPUT:**



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

**Teardown Methods in Junit**

**Answer:**

**PROGRAM:**

//InterestCalculator.java

package com.week2.bankapp;

public class InterestCalculator {

    public double applyMonthlyInterest(double balance) {

        return balance \* 1.01;

    }

    public double applyCustomInterest(double balance, double ratePercent) {

        return balance + (balance \* ratePercent / 100);

    }

}

//InterestCalculatorTest.java

package com.week2.bankapp;

import static org.junit.Assert.assertEquals;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

public class InterestCalculatorTest {

    private InterestCalculator calculator;

    @Before

    public void setUp() {

        calculator = new InterestCalculator();

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

    }

    @After

    public void tearDown() {

        System.out.println("Teardown: Cleaning up after test");

    }

    @Test

    public void testApplyMonthlyInterest() {

        double balance = 1000.0;

        double result = calculator.applyMonthlyInterest(balance);

        assertEquals(1010.0, result, 0.001);

    }

    @Test

    public void testApplyCustomInterest() {

        double balance = 2000.0;

        double rate = 5.0;

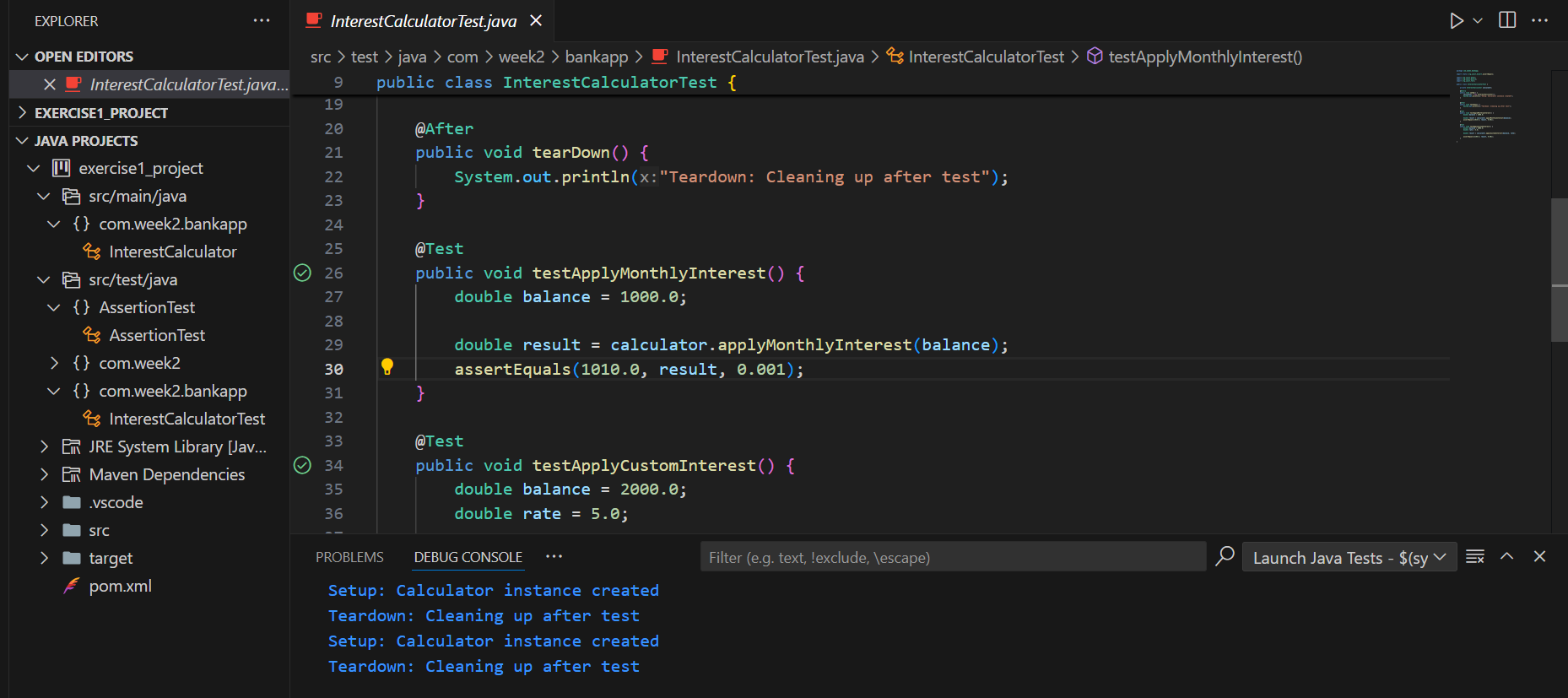
        double result = calculator.applyCustomInterest(balance, rate);

        assertEquals(2100.0, result, 0.001);

    }

}

**OUTPUT:**



**Mockito Hands-On Exercises**

**Exercise 1: Mocking and Stubbing**

**Answer:**

**PROGRAM**

//ExternalApi.java (public Interface)

package com.week2.api;

public interface ExternalApi {

    String getData();

}

//MyService.java

package com.week2.service;

import com.week2.api.ExternalApi;

public class MyService {

    private ExternalApi api;

    public MyService(ExternalApi api) {

        this.api = api;

    }

    public String fetchData() {

        return api.getData();

    }

}

//MyServiceTest.java

package com.week2.test;

import com.week2.api.ExternalApi;

import com.week2.service.MyService;

import org.junit.jupiter.api.Test;

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

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);

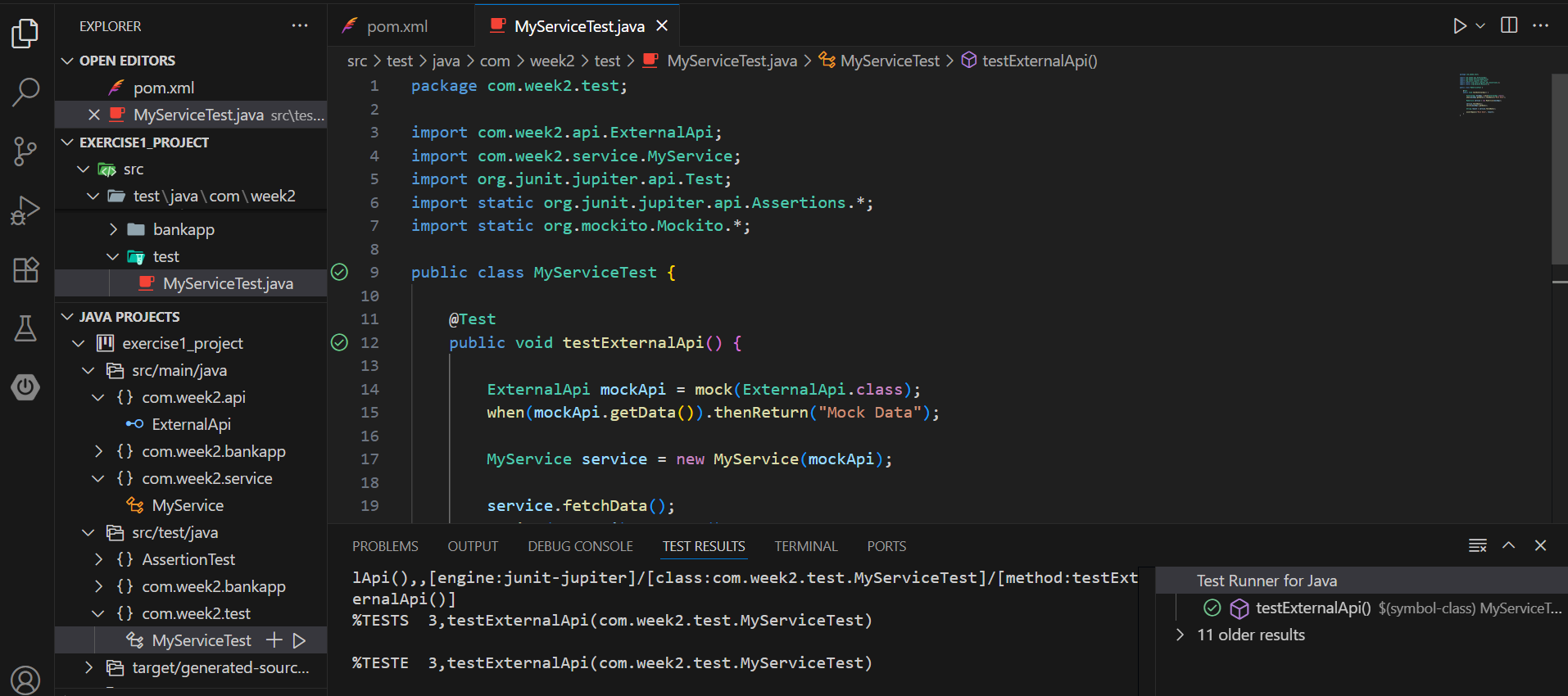
        String result = service.fetchData();

        assertEquals("Mock Data", result);

    }

}

OUTPUT:



**Exercise 2: Verifying Interactions**

**Answer:**

**PROGRAM:**

//ExternalApi.java (public Interface)

package com.week2.api;

public interface ExternalApi {

    String getData();

}

//MyService.java

package com.week2.service;

import com.week2.api.ExternalApi;

public class MyService {

    private ExternalApi api;

    public MyService(ExternalApi api) {

        this.api = api;

    }

    public String fetchData() {

        return api.getData();

    }

}

//MyServiceTest.java

package com.week2.test;

import com.week2.api.ExternalApi;

import com.week2.service.MyService;

import org.junit.jupiter.api.Test;

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

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);

        service.fetchData();

        verify(mockApi).getData();

    }

}

**OUTPUT:**

