TDD using JUnit5 and Mockito

Exercise 1: Setting Up JUnit

**Scenario:**

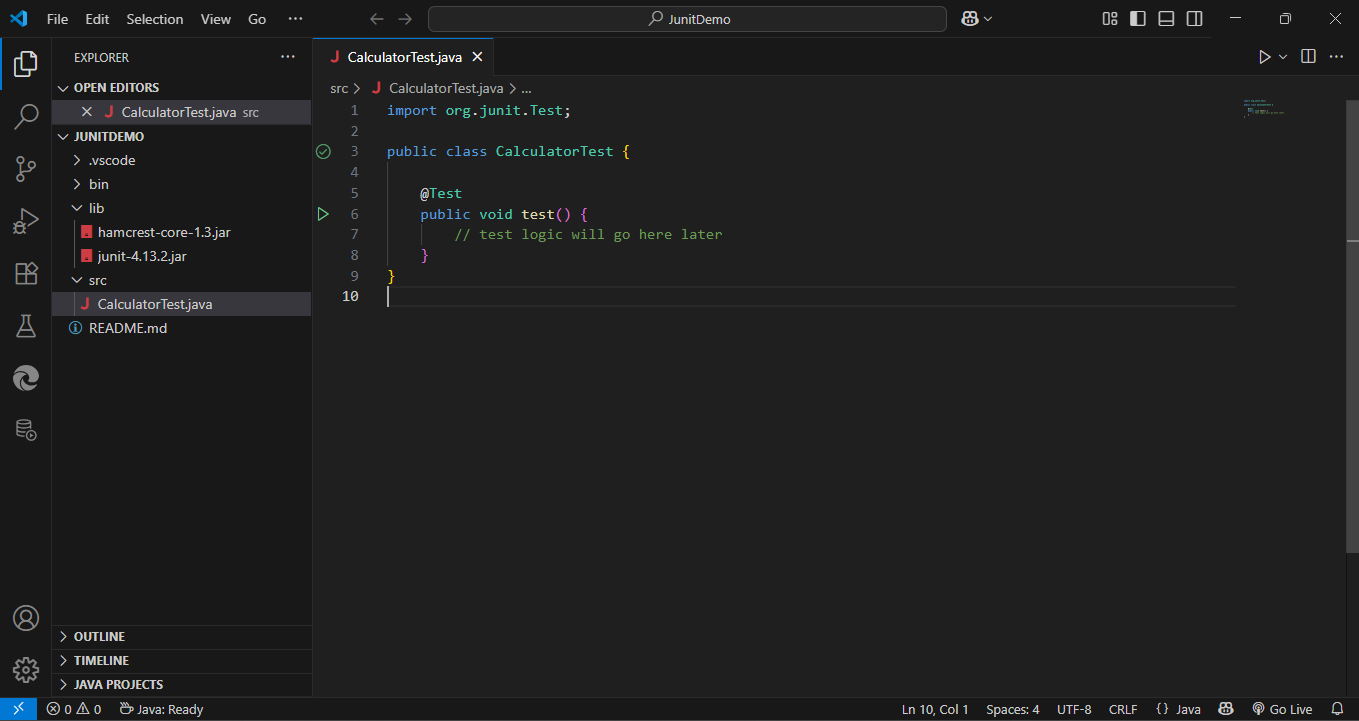
You need to set up JUnit in your Java project to start writing unit tests.

Steps for Setting Up JUnit in VS Code:

1. Create a Java project named **JUNITDEMO.**
2. Install extension in Visual Studio Code.
3. To add JUnit dependency to our project, download the following 2 files namely

* junit-4.13.2.jar
* hamcrest-core-1.3.jar

1. Place these two files in lib.
2. Create a test class in src file.



***Setting-Up JUnit***

Hence, Setting-Up JUnit in Java project is completed successfully and can be used to start writing unit tests.

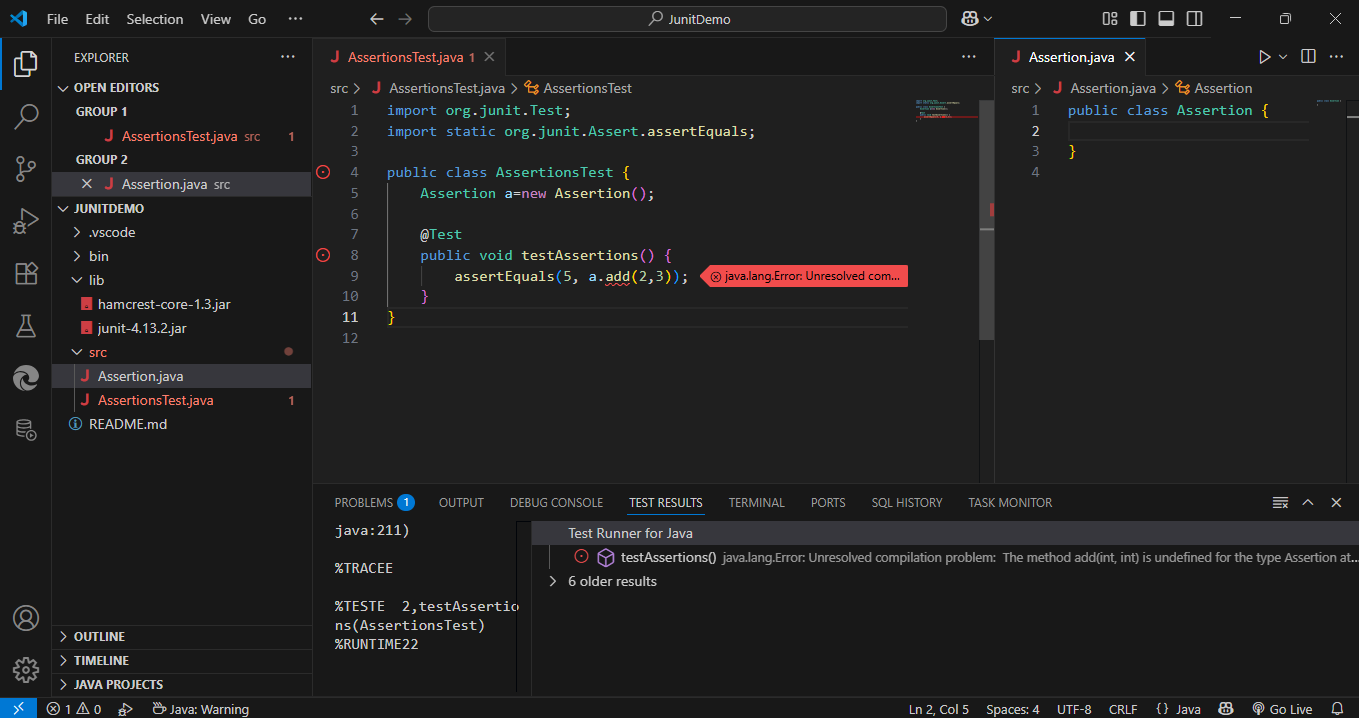
Exercise 3: Assertions in JUnit

**Scenario:**

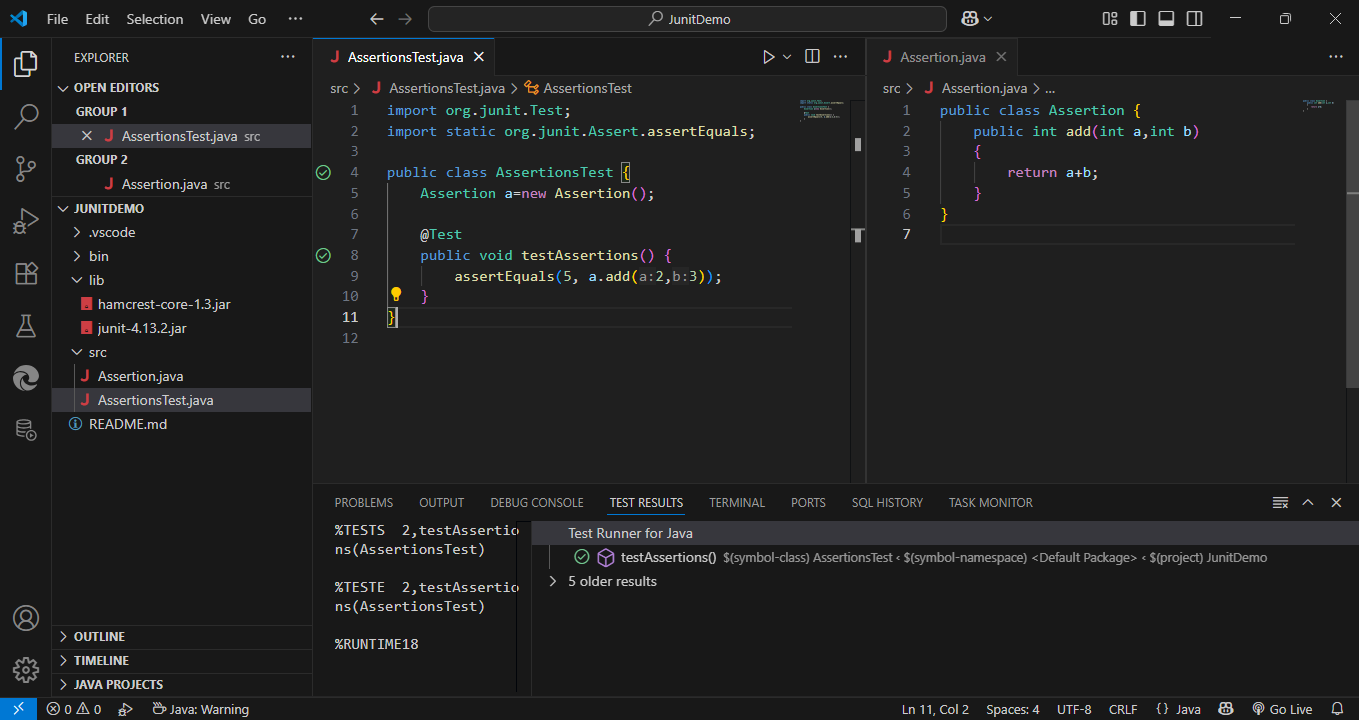
You need to use different assertions in JUnit to validate your test results.

**Test case1:** assert equals

Test



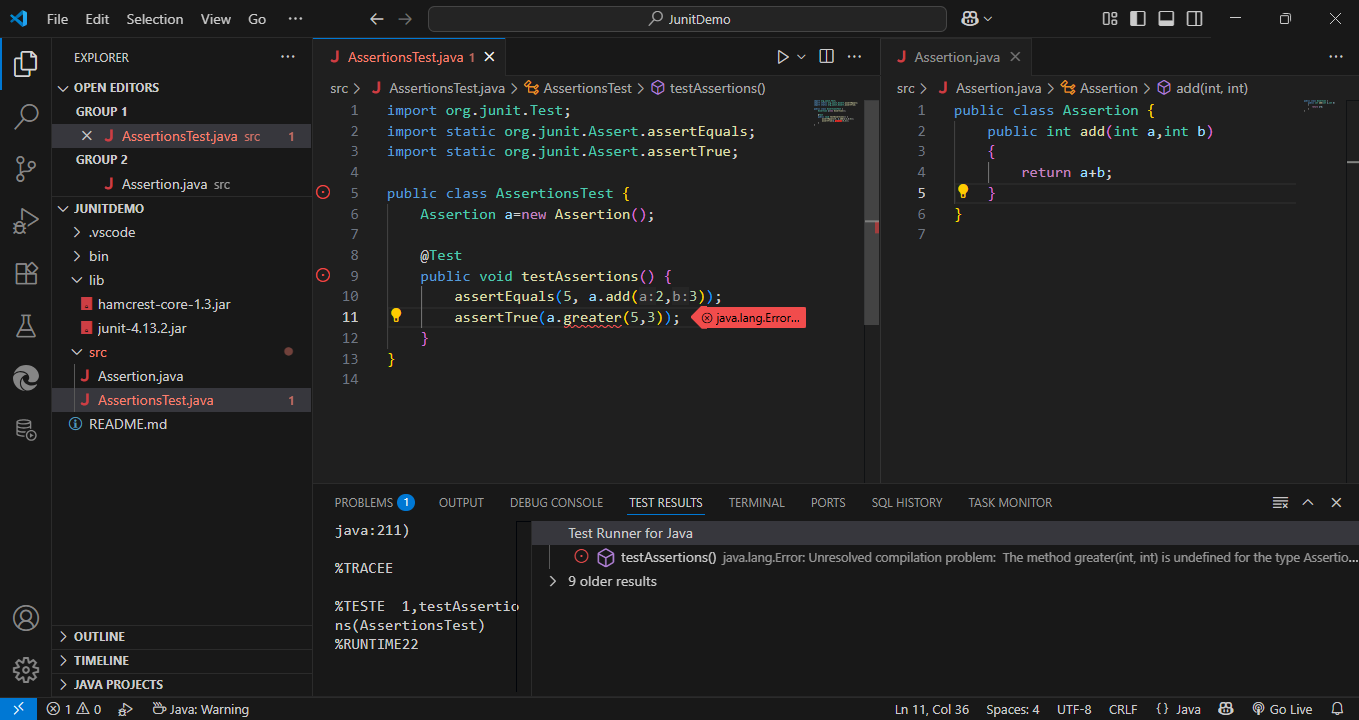
Write Code



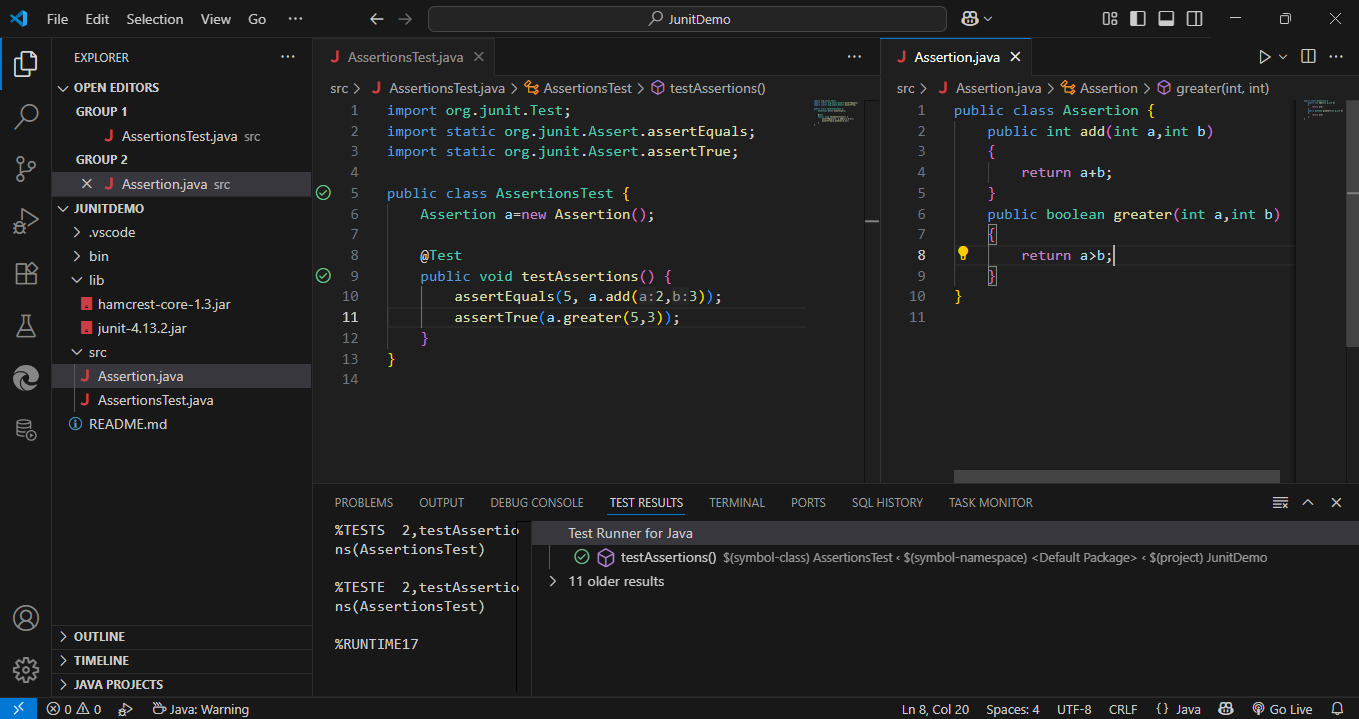
Refactor

**Test case2:** assert true

Test



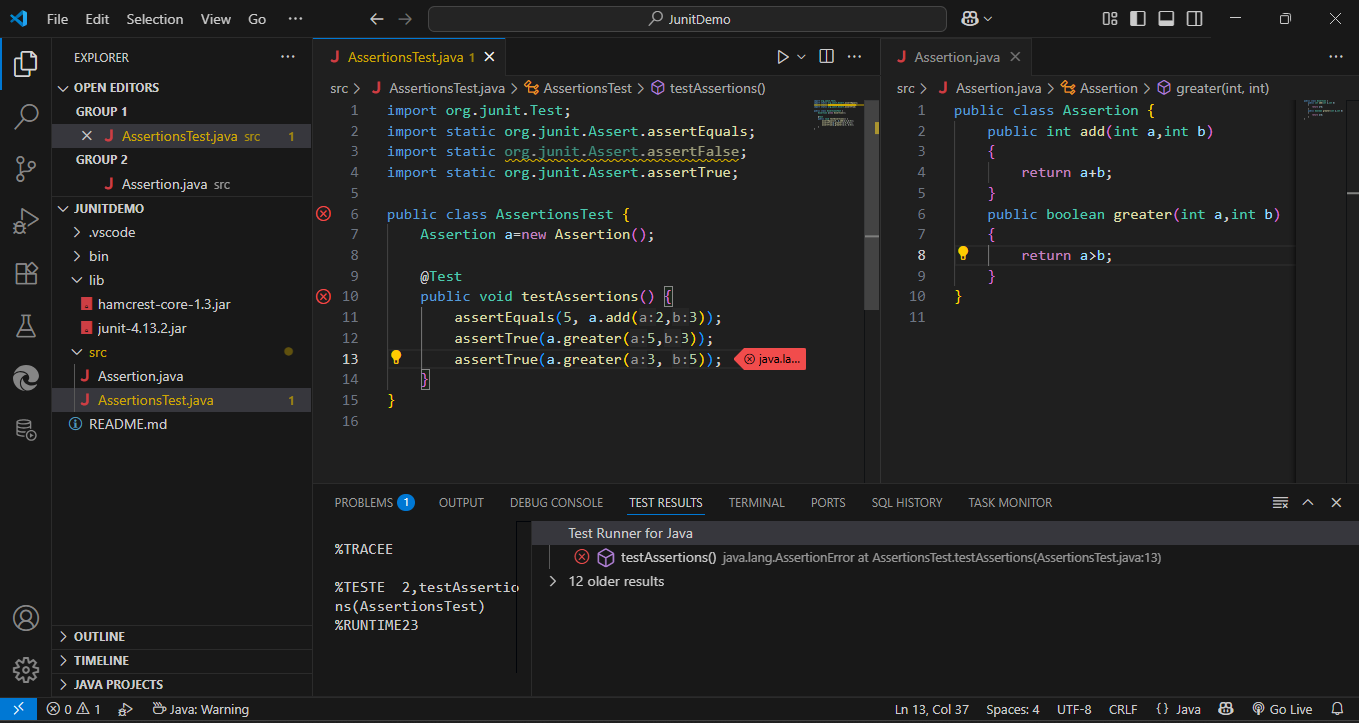
Write Code



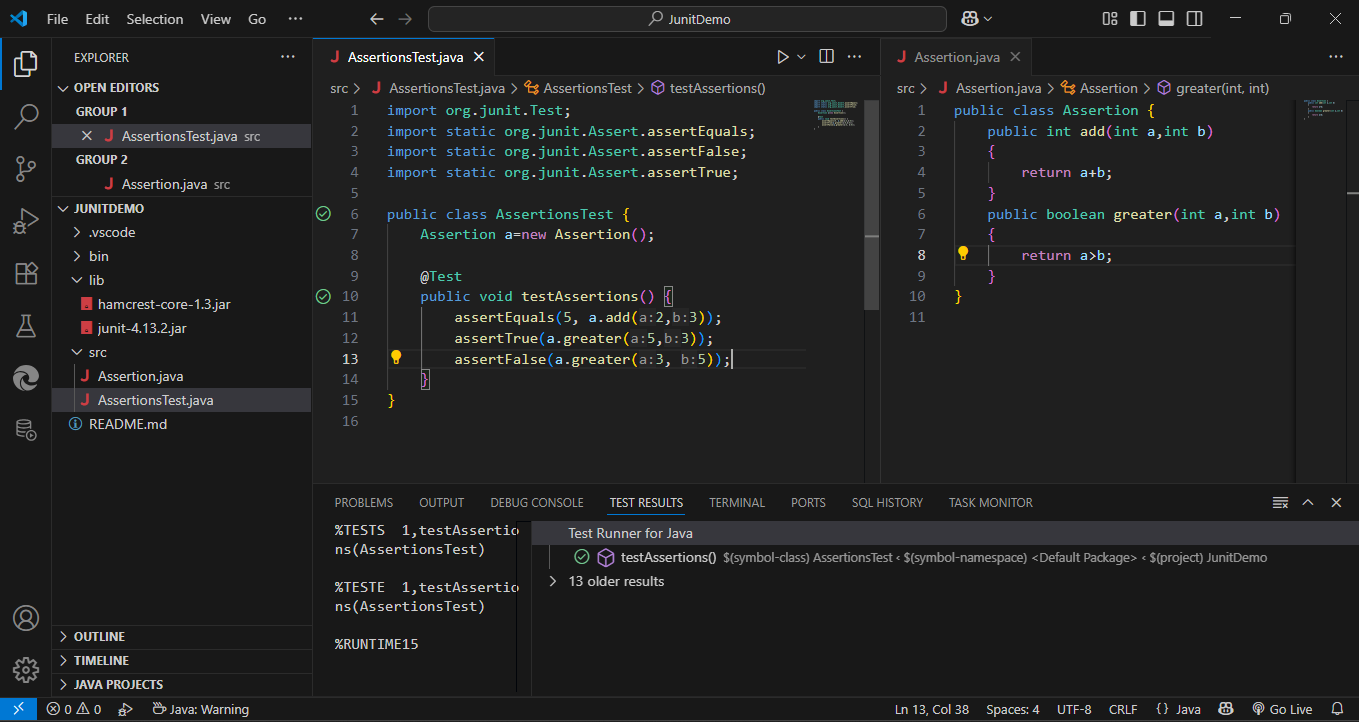
Refactor

**Test case3:** assert false

Test



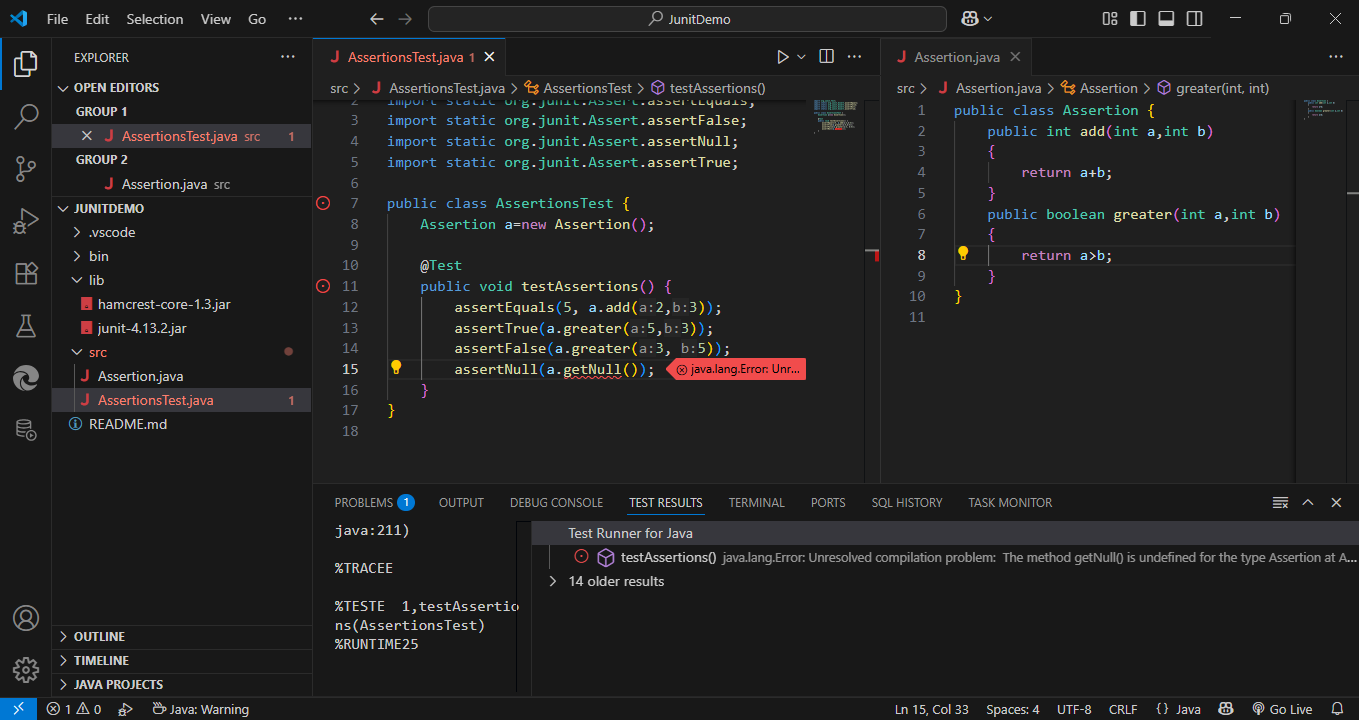
Write Code



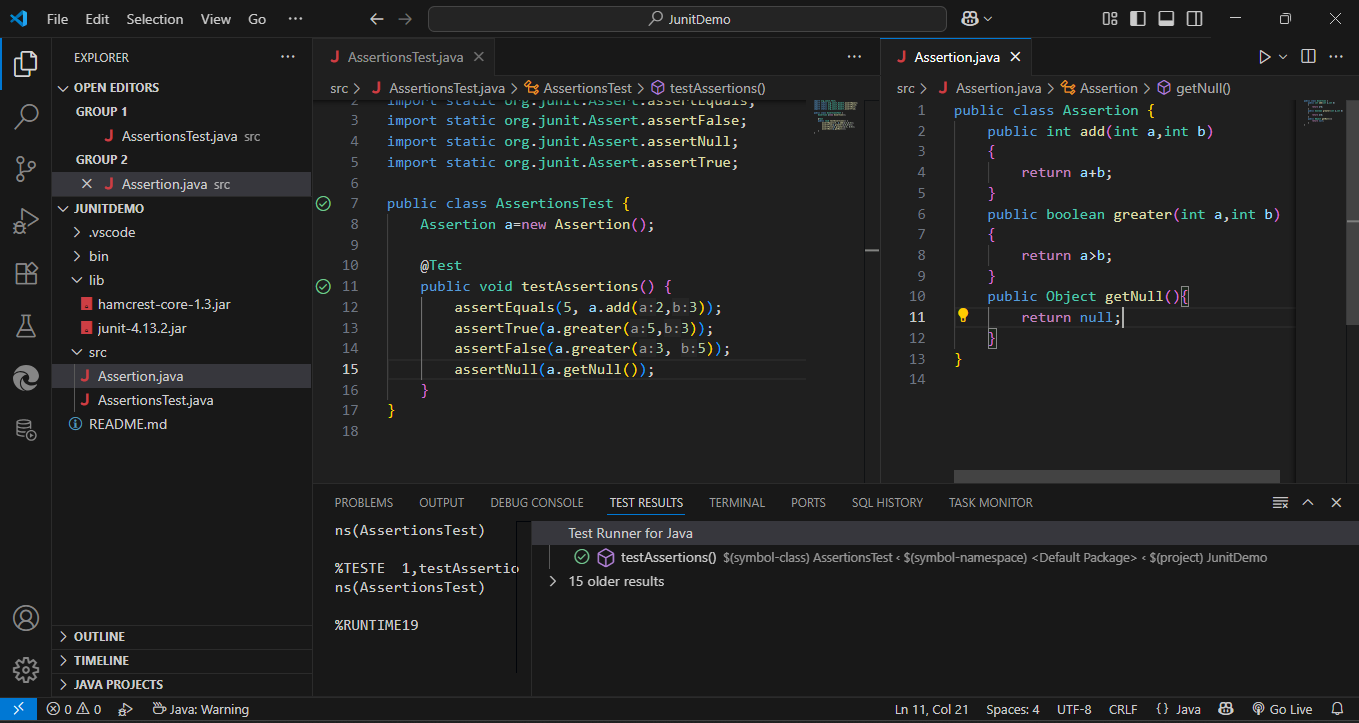
Refactor

**Test case4:** assert null

Test



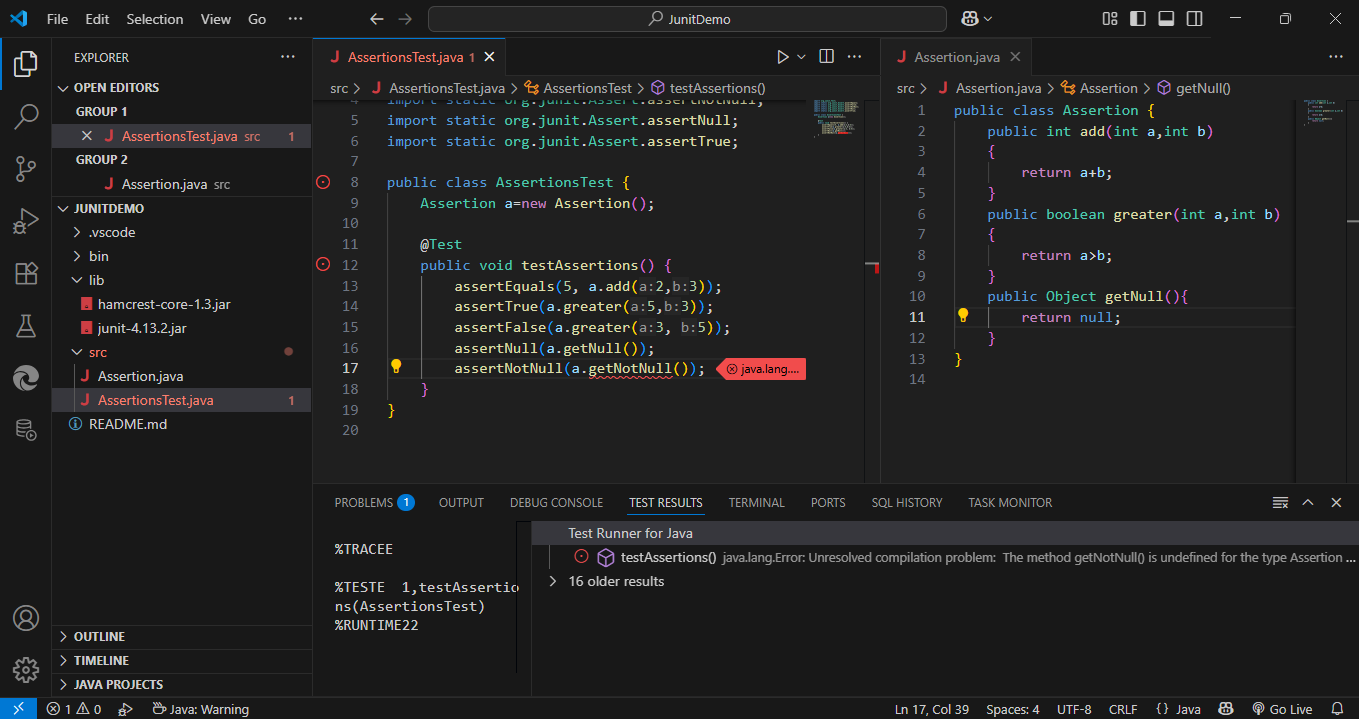
Write Code



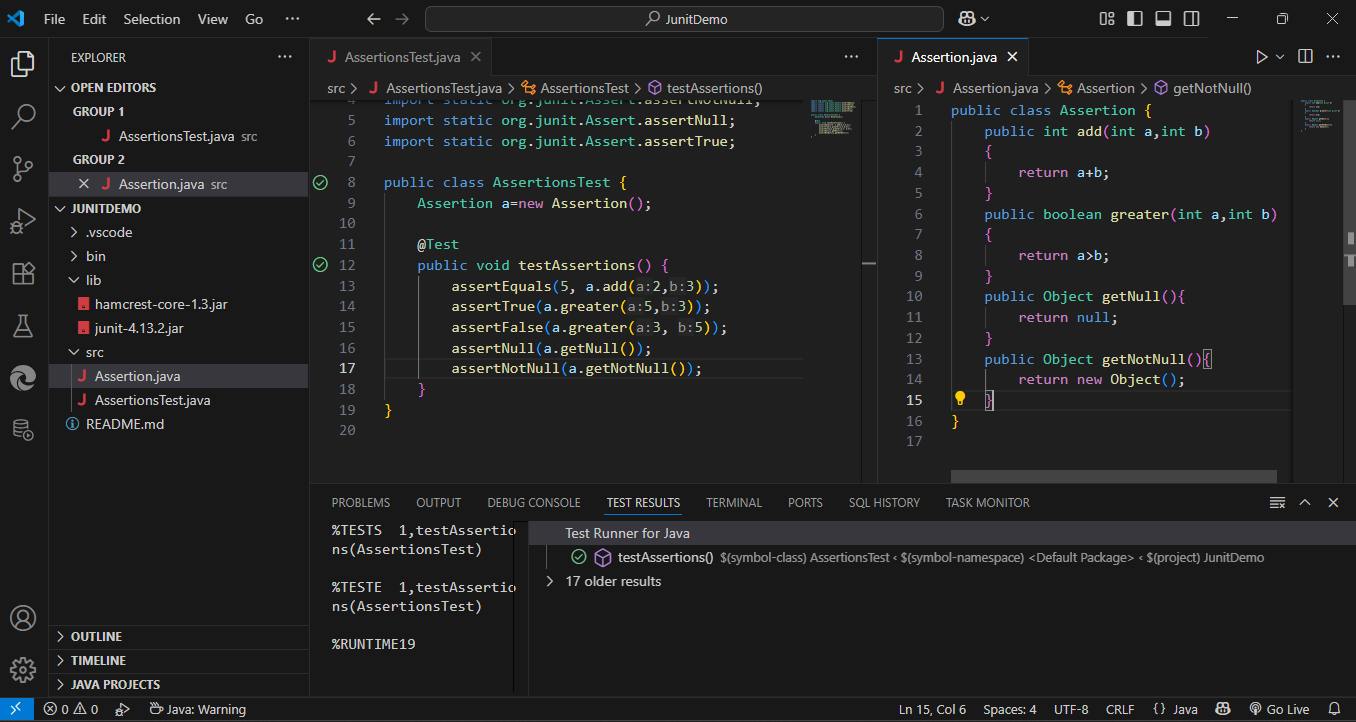
Refactor

**Test case5:** assert not null

Test



Write Code



Refactor

Assertion.java

public class Assertion {

    public int add(int a,int b)

    {

        return a+b;

    }

    public boolean greater(int a,int b)

    {

        return a>b;

    }

    public Object getNull(){

        return null;

    }

    public Object getNotNull(){

        return new Object();

    }

}

AssertionsTest.java

import org.junit.Test;

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;

public class AssertionsTest {

    Assertion a=new Assertion();

    @Test

    public void testAssertions() {

        assertEquals(5, a.add(2,3));

        assertTrue(a.greater(5,3));

        assertFalse(a.greater(3, 5));

        assertNull(a.getNull());

        assertNotNull(a.getNotNull());

    }

}

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

**Scenario:**

You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup and teardown methods.

1. Write tests using the AAA pattern.

import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class MessageTest {

    private Message msg;

    @Test

    public void testGreet() {

        // Arrange

        String name = "John";

        // Act

        String result = msg.greet(name);

        // Assert

        assertEquals("Hello, John", result);

    }

}

1. Use @Before and @After annotations for setup and teardown methods.

import static org.junit.Assert.assertEquals;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

public class MessageTest {

    private Message msg;

    @Before

    public void setUp() {

        msg = new Message(); // setup before each test

    }

    @After

    public void tearDown() {

        msg = null; // cleanup after each test

    }

  @Test

    public void testGreet() {

        String name = "John"; // Arrange

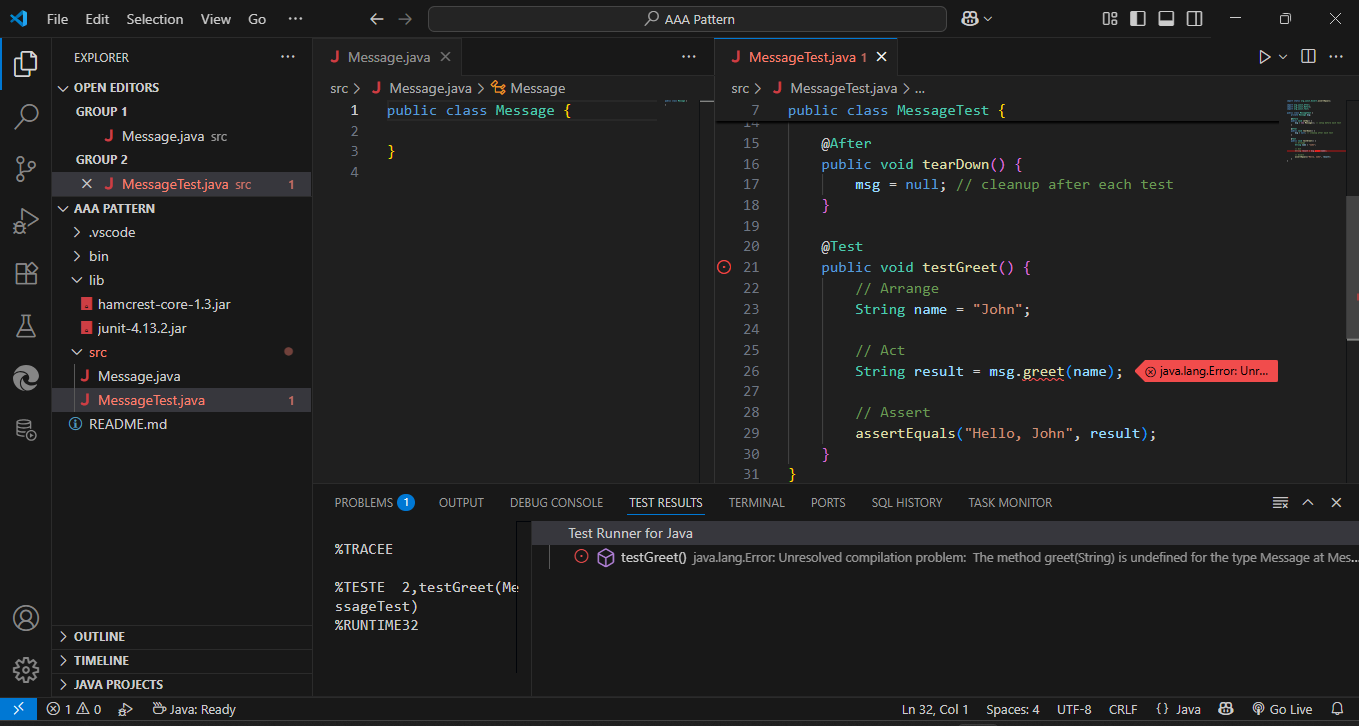
        String result = msg.greet(name); // Act

        assertEquals("Hello, John", result); // Assert

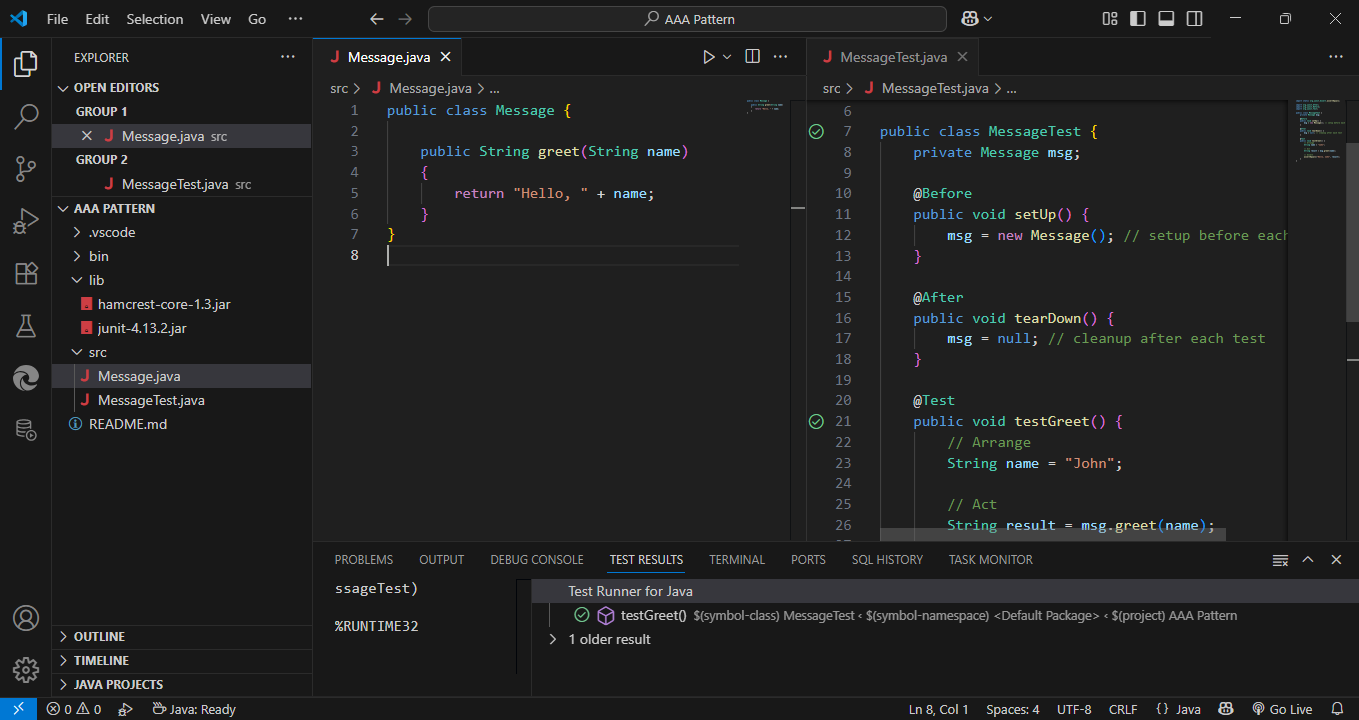
    }

}

Test



Write Code



Refactor

Message.java

public class Message {

    public String greet(String name)

    {

        return "Hello, " + name;

    }

}

MessageTest.java

import static org.junit.Assert.assertEquals;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

public class MessageTest {

    private Message msg;

    @Before

    public void setUp() {

        msg = new Message(); // setup before each test

    }

    @After

    public void tearDown() {

        msg = null; // cleanup after each test

    }

    @Test

    public void testGreet() {

        // Arrange

        String name = "John";

        // Act

        String result = msg.greet(name);

        // Assert

        assertEquals("Hello, John", result);

    }

}