# JUnit Week 3 Exercises

## Exercise 1: Setting Up JUnit

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

## Pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.example</groupId>

<artifactId>JUnitSetup</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

## SampleTest.java

import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class SampleTest {

@Test

public void sampleTestMethod() {

assertEquals(2, 1 + 1);

}

}

## Output

## 

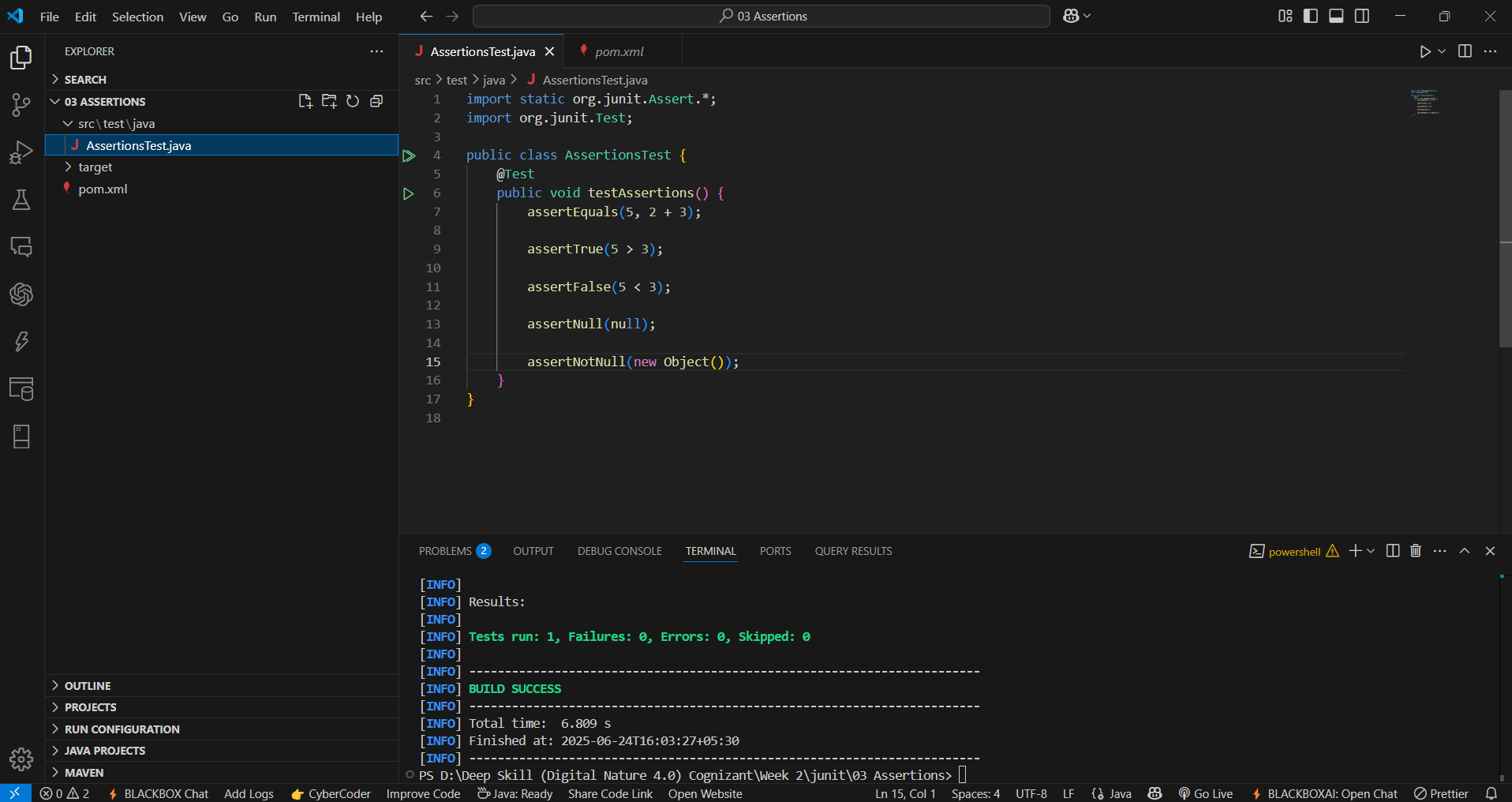
## Exercise 3: Assertions in JUnit

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

## AssertionsTest.java

import org.junit.Test;  
import static org.junit.Assert.\*;  
  
public class AssertionsTest {  
 @Test  
 public void testAssertions() {  
 assertEquals(5, 2 + 3);  
 assertTrue(5 > 3);  
 assertFalse(5 < 3);  
 assertNull(null);  
 assertNotNull(new Object());  
 }  
}

## Output



## Exercise 4: AAA Pattern, Setup and Teardown

Scenario:  
 You need to organize your tests using the AAA pattern and use setup and teardown methods.

## CalculatorTest.java

import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import static org.junit.Assert.\*;  
public class CalculatorTest {  
 private int result;  
 @Before  
 public void setUp() {  
 result = 0;  
 }  
 @After  
 public void tearDown() {  
 result = 0;  
 }  
 @Test  
 public void testAddition() {  
 int a = 2;  
 int b = 3;  
 result = a + b;  
 assertEquals(5, result);  
 }  
}

## Output

## 