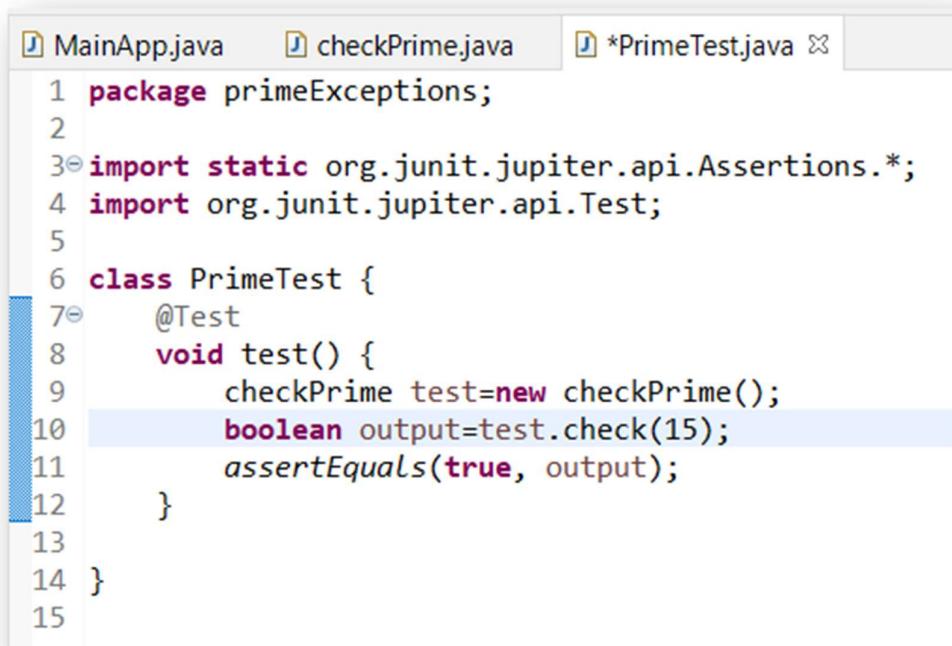


# Junit – Mockito – Logging Assignment

## 1. Junit

Junit Code Snippet:



The screenshot shows a Java code editor with three tabs at the top: 'MainApp.java', 'checkPrime.java', and '\*PrimeTest.java'. The code in the '\*PrimeTest.java' tab is as follows:

```
1 package primeExceptions;
2
3 import static org.junit.jupiter.api.Assertions.*;
4 import org.junit.jupiter.api.Test;
5
6 class PrimeTest {
7     @Test
8     void test() {
9         checkPrime test=new checkPrime();
10        boolean output=test.check(15);
11        assertEquals(true, output);
12    }
13
14 }
```

The line 'boolean output=test.check(15);' is highlighted with a light blue background.

Figure 1 Showing Junit Test Case

## Test Results

### 1. Pass Results (Passing 5 to be checked as Prime)

The screenshot shows the Eclipse IDE interface. On the left, the 'Package Explorer' and 'JUnit' view are visible. The 'JUnit' view displays the test results: 'Finished after 0.09 seconds', 'Runs: 1/1', 'Errors: 0', and 'Failures: 0'. A green progress bar indicates success. On the right, the code editor shows three files: MainApp.java, checkPrime.java, and PrimeTest.java. The PrimeTest.java code is as follows:

```
1 package primeExceptions;
2
3 import static org.junit.jupiter.api.Assertions.*;
4 import org.junit.jupiter.api.Test;
5
6 class PrimeTest {
7     @Test
8     void test() {
9         checkPrime test=new checkPrime();
10        boolean output=test.check(5);
11        assertEquals(true, output);
12    }
13
14 }
```

Figure 2 Showing Test Cases Passed

### 2. Fail Results (Passing 15 to be checked as Prime)

The screenshot shows the Eclipse IDE interface. The 'JUnit' view displays the test results: 'Finished after 0.101 seconds', 'Runs: 1/1', 'Errors: 0', and 'Failures: 1'. A red progress bar indicates failure. The 'Failure Trace' view at the bottom shows the error message: 'org.opentest4j.AssertionFailedError: expected: <true> but was: <false>'. The code editor on the right shows the same PrimeTest.java code as in Figure 2, but with a failure in the test case for the value 15.

```
1 package primeExceptions;
2
3 import static org.junit.jupiter.api.Assertions.*;
4 import org.junit.jupiter.api.Test;
5
6 class PrimeTest {
7     @Test
8     void test() {
9         checkPrime test=new checkPrime();
10        boolean output=test.check(15);
11        assertEquals(true, output);
12    }
13
14 }
```

Figure 3 Showing failed test case when 15 is being checked as prime

## 2. Mockito

### Mockito Code Snippet:



The screenshot shows a Java code editor with the following code snippet for a Mockito test:

```
1 package AccoliteUniversity.MockitoPrime;
2
3 import static org.mockito.Mockito.mock;
4 import static org.mockito.Mockito.when;
5
6 import org.junit.Before;
7 import org.junit.Test;
8 import org.mockito.Mockito.*;
9
10 import junit.framework.Assert;
11
12 public class CheckServiceTest {
13
14     CheckService chkObj;
15
16     @Before
17     public void setup() {
18         IsPrime i0bj= mock(IsPrime.class);
19         when(i0bj.check(5)).thenReturn(true);
20         chkObj=new CheckService();
21         chkObj.setObj(i0bj);
22     }
23
24     @Test
25     public void testCheckPrime() {
26         Assert.assertEquals(true, chkObj.checkPrime(5));
27     }
28
29 }
```

The code uses Mockito annotations (@Before, @Test) and methods (mock, when, assertEquals) to set up a mock object and verify its behavior. The code is part of a project named "MockitoPrime" with files like pom.xml, IsPrime.java, CheckService.java, and CheckServiceTest.java.

Figure 4 Showing Mockito Test Code

## Pass Test Case:

The screenshot shows the Eclipse IDE interface with the Package Explorer and JUnit view. The JUnit view displays a green bar indicating 'Runs: 1/1', 'Errors: 0', and 'Failures: 0'. Below this, it shows the class 'AccoliteUniversity.MockitoPrime.CheckServiceTest' running under 'JUnit 4' in 0.393 seconds. The code editor on the right contains the Java code for the test class, which includes imports for Mockito and JUnit, a setup method using @Before, and a test method 'testCheckPrime' that asserts the result of the 'checkPrime' method on a mock object. A red arrow points to the assert statement in the test method.

```
1 package AccoliteUniversity.MockitoPrime;
2
3 import static org.mockito.Mockito.mock;
4 import static org.mockito.Mockito.when;
5
6 import org.junit.Before;
7 import org.junit.Test;
8 import org.mockito.Mockito.*;
9
10 import junit.framework.Assert;
11
12 public class CheckServiceTest {
13
14     CheckService chkObj;
15
16     @Before
17     public void setup() {
18         ISpPrime iObj= mock(IspPrime.class);
19         when(iObj.check(5)).thenReturn(true);
20         chkObj=new CheckService();
21         chkObj.setObj(iObj);
22     }
23
24     @Test
25     public void testCheckPrime() {
26         Assert.assertEquals(true, chkObj.checkPrime(5));
27     }
}
```

Figure 5 Showing passed test case for checking 5 as prime

## Failed Test Case:

The screenshot shows the Eclipse IDE interface with the Package Explorer and JUnit view. The JUnit view displays a red bar indicating 'Runs: 1/1', 'Errors: 0', and 'Failures: 1'. Below this, it shows the class 'AccoliteUniversity.MockitoPrime.CheckServiceTest' running under 'JUnit 4' in 0.386 seconds. The code editor on the right contains the Java code for the test class, which includes imports for Mockito and JUnit, a setup method using @Before, and a test method 'testCheckPrime' that asserts the result of the 'checkPrime' method on a mock object. A red arrow points to the assert statement in the test method. The failure message in the Failure Trace indicates that the expected value was true but the actual value was false.

```
1 package AccoliteUniversity.MockitoPrime;
2
3 import static org.mockito.Mockito.mock;
4 import static org.mockito.Mockito.when;
5
6 import org.junit.Before;
7 import org.junit.Test;
8 import org.mockito.Mockito.*;
9
10 import junit.framework.Assert;
11
12 public class CheckServiceTest {
13
14     CheckService chkObj;
15
16     @Before
17     public void setup() {
18         ISpPrime iObj= mock(IspPrime.class);
19         when(iObj.check(15)).thenReturn(false);
20         chkObj=new CheckService();
21         chkObj.setObj(iObj);
22     }
23
24     @Test
25     public void testCheckPrime() {
26         Assert.assertEquals(true, chkObj.checkPrime(15));
27     }
}
```

Figure 6 Showing failed test case which classifies 15 as Prime