

Forman Christian College (A Chartered University)



Estd. 1864

Software Quality Assurance - CSCS 351

Section A

Assignment1

Muhammad Abdullah Bin Tariq

22-10307

Test Cases:

Below are attachments of test cases used in this assignment:

Test Case1:

```
public void testcase1(){
    System.out.println(x: "Test case 1");
    assertEquals(5, add(num1: 1, num2: 2));
}
```

```
import org.junit.test;

public class MyTestcase1 {

    public static int add(int num1, int num2) {
        return num1 + num2;
    }

    @Test
    public void testcase1(){
        System.out.println(x: "Test case 1");
        assertEquals(5, add(num1: 3, num2: 2));
    }
}
```

TestCase2:

```
public static int sub(int num1, int num2) {
    return num1 - num2;
}

@Test
public void testcase2(){
    system.out.println(x: "Test case 2");
    assertEquals(1, sub(num1: 3, num2: 2));
}
```

```
assertEquals(6, mul(num1: 1, num2: 2));
```

Test Case 3:

The screenshot shows an IDE with a test results pane on the left and a code editor on the right. The test results pane lists several test cases, with 'MyTestcase3' and 'testcase3()' both marked as passed (green checkmarks) with a duration of 9.0ms. The code editor displays the implementation of 'testcase3()', which prints 'Test case 3' and asserts that the product of 3 and 2 is 6.

```
8      return num1 * num2;
9  }
10
11  @Test
12  public void testcase3(){
13      System.out.println(x: "Test case 3");
14
15      assertEquals(6, mul(num1: 3, num2: 2));
16  }
17
18
19 }
```

Test Case 4:

The screenshot shows an IDE with a test results pane on the left and a code editor on the right. The test results pane shows a hierarchy of test cases, with 'MyTestcase4' and 'testcase4()' both marked as passed (green checkmarks) with a duration of 7.0ms. The code editor displays the implementation of 'testcase4()', which prints 'Test case 4' and asserts that the result of dividing 6 by 2 is 3.

```
3  import org.junit.Test;
4
5  public class MyTestcase4 {
6      public static int div(int num1, int num2) {
7          return num1 / num2;
8      }
9
10
11  @Test
12  public void testcase4(){
13      System.out.println(x: "Test case 4");
14
15      assertEquals(3, div(num1: 6, num2: 2));
16  }
17 }
```

Test Case 5:

The screenshot displays an IDE with a test runner on the left and a code editor on the right. The test runner shows a tree of test cases, with 'MyTestcase5' selected and highlighted. The code editor shows the implementation of 'MyTestcase5', which includes a static 'pow' method and a 'testcase5' test method.

```
import org.junit.Test;

public class MyTestcase5 {
    public static int pow(int num1, int num2) {
        if (num1 == 0 || num2 == 0)
            return 1;
        else
            return num1 * pow(num1, num2 - 1);
    }

    @Test
    public void testcase5(){
        System.out.println(x: "Test case 5");

        assertEquals(9, pow(num1: 3, num2: 2));
    }
}
```

Test Suite:

The screenshot displays an IDE with a test runner on the left and a code editor on the right. The test runner shows a tree of test suites, with 'MyTestsuite' selected and highlighted. The code editor shows the implementation of 'MyTestsuite', which includes annotations for running with a suite and a list of suite classes, followed by the class definition.

```
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    MyTestcase1.class,
    MyTestcase2.class,
    MyTestcase3.class,
    MyTestcase4.class,
    MyTestcase5.class})

public class MyTestsuite {}
```

Program Code:

TestCase1.java

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

import org.junit.Test;

public class MyTestcase1 {

    public static int add(int num1, int num2) {
        return num1 + num2;
    }

    @Test
    public void testcase1(){
        System.out.println("Test case 1");

        assertEquals(5, add(3, 2));
    }

}
```

TestCase2.java

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

import org.junit.Test;

public class MyTestcase2 {

    public static int sub(int num1, int num2) {
        return num1 - num2;
    }

    @Test
    public void testcase2(){
        System.out.println("Test case 2");

        assertEquals(1, sub(3, 2));
    }

}
```

```
}
```

TestCase3.java

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

import org.junit.Test;

public class MyTestcase3 {

    public static int mul(int num1, int num2) {
        return num1 * num2;
    }

    @Test
    public void testcase3(){
        System.out.println("Test case 3");

        assertEquals(6, mul(3, 2));
    }

}
```

TestCase4.java

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

import org.junit.Test;

public class MyTestcase4 {
    public static int div(int num1, int num2) {
        return num1 / num2;
    }

    @Test
    public void testcase4(){
        System.out.println("Test case 4");

        assertEquals(3, div(6, 2));
    }

}
```

```
}
```

TestCase5.java

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

import org.junit.Test;

public class MyTestcase5 {
    public static int pow(int num1, int num2) {
        if (num1 == 0 || num2 == 0)
            return 1;
        else
            return num1 * pow(num1, num2 - 1);
    }

    @Test
    public void testcase5(){
        System.out.println("Test case 5");

        assertEquals(9, pow(3, 2));
    }
}
```

MyTestSuite.java

```
import org.junit.runner.RunWith;
import org.junit.runners.Suite;

@RunWith(Suite.class)
@Suite.SuiteClasses({
    MyTestcase1.class,
    MyTestcase2.class,
    MyTestcase3.class,
    MyTestcase4.class,
    MyTestcase5.class})

public class MyTestsuite {}
```