Faculty : Mr. Tarek Mizan
Lab Instructor: Nazmul Alam Diptu
Email : nazmul.diptu@northsouth.edu
Class Timing: ST 1:00 PM – 2:30 PM (LIB-611)
Topic: TDD continue, File

Objective

1. TDD (Test Driven Development)
2. Create File, Write in a File, Read from a file

`CalculationUnit.java`

```
public class CalculationUnit {

    public int addition(int a, int b) {
        return a + b ;
    }
    public int subtract(int a, int b) {
        return a - b ;
    }
    public int multiply(int a, int b) {
        return a * b ;
    }
    public double divide(int a, int b) {
        try{
            return a / b ;
          }
          catch(ArithmeticException e){
             System.out.println ("You Shouldn't divide a number by zero");
          }
        return a / b ;
    }

}
```

`CalculationUnitTest.java`

```
class CalculationUnitTest {
    CalculationUnit c;

    @BeforeEach
    void init() {

        c = new CalculationUnit();
    }

    @Nested
    @DisplayName("Multiply")
    class MultiplyTest{

        @Test
        @DisplayName("Multiplying two + number")
        void testTowPosivive() {

            assertEquals(1, c.multiply(1, 1));
        }


        @Test
        @DisplayName("Multiplying two - number")
        void testTwoNegetive() {

            assertEquals(1, c.multiply(-1, -1));
        }

        @Test
        @DisplayName("Multiplying one positive with one negetive numbers")
        void testPosNege() {

            assertEquals(-1, c.multiply(1, -1));
        }
        @Test
        @DisplayName("Multiplying zero")
        void testOneZeor() {

            assertEquals(0, c.multiply(1, 0));
        }
    }


    @Test
    @DisplayName("Testing addition method for CalculationUnit")
    void testAddition() {
        //CalculationUnit c = new CalculationUnit();

        int a = 7;
        int b = 3;
        int expected = 10;
        int actual = c.addition(a, b);
        String msg =  a + " + " + b + " = " + (a+b) ;
        assertEquals(expected, actual, msg);
    }

    @Test
    @DisplayName("Testing subtruct method for CalculationUnit")
    void testSubtruction() {
        assertAll(
                ()-> assertEquals(0,c.subtruct(1, 1)),
                ()-> assertEquals(1,c.subtruct(2, 1)),
                ()-> assertEquals(-1,c.subtruct(1, 2))
                );
    }

    @Test
    void testDivide() {
        //CalculationUnit c = new CalculationUnit();
        assertThrows(ArithmeticException.class ,()-> c.divide(7, 0),
"Devide by zero should through Arithmetic Exception");
    }

}
```

CircleUnit.java

```
public class CircleUnit {

    public double diameter(double radious) {
        return 2 * radious;
    }
    public double circleArea(double radious) {
        return Math.PI * radious * radious;
    }

}
```

CircleUnitTest

```
class CircleUnitTest {

    CircleUnit c;
    TestInfo testInfo;
    TestReporter testReporter;
    double radious = 10;

    @BeforeEach
    void init(TestInfo testInfo, TestReporter testReporter) {

        c = new CircleUnit();
        this.testInfo = testInfo;
        this.testReporter = testReporter;
        testReporter.publishEntry(testInfo.getDisplayName());

    }

    @Test
    void testCircleDiameter(){

        double expected = 2 * radious;
        double actual = c.diameter(radious);
        // Lazy assert -> message created only if test fails.
        assertEquals(expected, actual,()-> "Diameter of a circle with 
radious " + radious + " should be " + expected);
    }

    @Test
    @Tag("Circle")
    void testCircleArea(){

        double expected = Math.PI * radious * radious;
        double actual = c.circleArea(radious);
        // Assert msg is created every time
        String msg =  "Area of a circle with radious " + radious + " should 
be " + expected;
        assertEquals(expected, actual, msg);
    }

    @Test
    @Disabled
    @DisplayName("Testing CircleCircumference method for CalculationUnit")
    @Tag("Circle")
    void testCircleCircumference() {

        double expected = 2 * Math.PI * radious;
        double actual = 0;
        //double actual = c.circleCircumference(radious);
        String msg =  "Circumperence of a circle with radious " + radious + 
" should be " + expected;
        assertEquals(expected, actual, msg);
    }
}
```

Faculty : Mr. Tarek Mizan
Lab Instructor: Nazmul Alam Diptu
Email : nazmul.diptu@northsouth.edu
Class Timing: ST 1:00 PM – 2:30 PM (LIB-611)
Topic: TDD continue, File

Tasks:

1. Ctreatte a java file called Rectangle.java. Inside that implements thse following methods :

- area(double length, double width)
- perimeter(double length, double width)
- didiagonal(double length, double width)

  Now create and implement Junit test for above methods.

1. Ctreatte a java file called EquilateralTriangle.java. Inside that implements thse following methods :

- area(double side)
- height(double side)

  Now create and implement Junit test for above methods.