**JUNIT ASSIGNMENTS:**

**1st code:**

public class MinMax {

public static int[] findMaxMin(int[] inputArr)

{

int[] minMaxValue = new int[2];

int max = inputArr[0];

int min = inputArr[0];

for(int i = 1 ; i < inputArr.length ; i++)

{

if(inputArr[i] > max)

max=inputArr[i];

if(inputArr[i] < min)

min=inputArr[i];

}

minMaxValue[0] = min;

minMaxValue[1] = max;

return minMaxValue; //returning array

}

}

MinMaxFinderTest.java

import static org.junit.jupiter.api.Assertions.\*;

import java.util.Arrays;

import org.junit.jupiter.api.Test;

class MinMaxTest {

int[] result = new int[2];

@Test

void 1first1() {

result = MinMax.findMaxMin(new int[] {1,3,56,26,32,755,0,4535,42,21});

int[] expectedResult = {0,4535};

assertEquals(Arrays.toString(expectedResult), Arrays.toString(result));

}

@Test

void second2() {

result = MinMax.findMaxMin(new int[] {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0});

int[] expectedResult = {0,0};

assertEquals(Arrays.toString(expectedResult), Arrays.toString(result));

}

@Test

void third3() {

result = MinMax.findMaxMin(new int[] {1,3,4,5,6,723,563,121231,545,2,56,6});

int[] expectedResult = {1,121231};

assertEquals(Arrays.toString(expectedResult), Arrays.toString(result));

}

@Test

void fourth4() {

result = MinMax.findMaxMin(new int[] {0,324,234,23,521,55,555,55666,555,77});

int[] expectedResult = {0,55666};

assertEquals(Arrays.toString(expectedResult), Arrays.toString(result));

}

@Test

void fifth5() {

result = MinMax.findMaxMin(new int[] {333,33,333,333,333,3333,3333333,333,33});

int[] expectedResult = {33,3333333};

assertEquals(Arrays.toString(expectedResult), Arrays.toString(result));

}

}

**2nd code:**

package MinMaxFromArray;

public class MinMax {

private int[] minMax = new int[2];

public int[] getMinMax() {

return minMax;

}

public void setMinMax(int min , int max) {

this.minMax[0] = min;

this.minMax[1] = max;

}

}

FindMinMax.java

package MinMaxFromArray;

public class FindMinMax {

public static MinMax maxMinInArray ( int[] inputArray )

{ MinMax obj = new MinMax(); //created a object to store min max from input array

int min = inputArray[0]; //min

int max = inputArray[0]; //max

for(int i = 1 ; i < inputArray.length ; i++)

{ if( inputArray[i] > max ) //for max value

max=inputArray[i];

if( inputArray[i] < min ) //for min value

min=inputArray[i];

}

obj.setMinMax(min,max); // stored min max of array in the object

return obj; //returning the object

}

}

FindMinMaxTest.java

package MinMaxFromArray;

import static org.junit.jupiter.api.Assertions.\*;

import java.util.Arrays;

import org.junit.jupiter.api.Test;

class FindMinMaxTest {

MinMax testObject;

@Test

void test1() {

testObject = FindMinMax.maxMinInArray(new int[] {1,3,56,26,32,755,0,4535,42,21});

int[] expected = {0,4535};

int[] actual = testObject.getMinMax();

assertEquals(Arrays.toString(expected),Arrays.toString(actual));

}

@Test

void test2() {

testObject = FindMinMax.maxMinInArray(new int[] {12,46,78,123,7,2325,3232,7644,211235});

int[] expected = {7,211235};

int[] actual = testObject.getMinMax();

assertEquals(Arrays.toString(expected),Arrays.toString(actual));

}

@Test

void test3() {

testObject = FindMinMax.maxMinInArray(new int[] {14,62,632,6344,776,2345,45232,4331});

int[] expected = {14,45232};

int[] actual = testObject.getMinMax();

assertEquals(Arrays.toString(expected),Arrays.toString(actual));

}

@Test

void test4() {

testObject = FindMinMax.maxMinInArray(new int[] {1});

int[] expected = {1,1};

int[] actual = testObject.getMinMax();

assertEquals(Arrays.toString(expected),Arrays.toString(actual)); }

@Test

void test5() {

testObject = FindMinMax.maxMinInArray(new int[] {1234,12});

int[] expected = {12,1234};

int[] actual = testObject.getMinMax();

assertEquals(Arrays.toString(expected),Arrays.toString(actual));

}

}

**3rd code:**

import org.junit.jupiter.api.Assertions;

public class BankTest {

public int take()

{

Bank b=new Bank();

b.Withdraw(5500);

Assertions.assertEquals(4500,4500);

}

}

import javax.net.ssl.ExtendedSSLSession;

class InsufficientFundExceeption extends Exception{

InsufficientFundExceeption(String msg)

{

super(msg);

}

}

public class Bank {

public int Withdraw(int a)

{

int with=a;

int bal=10000;

if(bal<a)

{

String msg;

throw new InsufficientFundExceeption(msg);

}

else

{

int c;

c=bal-with;

return c;

}

}

}

**4th code:**

import org.junit.jupiter.api.\*;

public class Calculator {

Calc ca;

@BeforeEach

void init()

{

System.out.println("Instialize the class object");

// Calc ca= new Calc();

}

@AfterEach

void clean()

{

System.out.println("Clean the memory");

}

@Test

public void Badd()

{

ca.add(2,5);

Assertions.assertEquals(7,7);

System.out.println("Testcase pass for addition");

}

@Test

public void Csub()

{

ca.sub(6,5);

Assertions.assertEquals(1,1);

System.out.println("Testcase pass for subtraction");

}

@Test

public void Dmul()

{

ca.mul(5,6);

Assertions.assertEquals(30,30);

System.out.println("Testcase pass for multiplication");

}

@Test

public void Ediv()

{

ca.div(3,2);

Assertions.assertEquals(1,1);

System.out.println("Testcase pass for Division");

}

}

public class Calc {

public int add(int a1 ,int b1)

{ int add1;

add1=a1+b1;

return add1;

}

public int sub(int a2, int b2)

{

int sub1;

sub1=a2-b2;

return sub1;

}

public int mul(int a3 , int b3)

{

int mul1;

mul1=a3\*b3;

return mul1;

}

public int div(int a4,int b4)

{

int div1;

div1=a4/b4;

return div1;

}

}