Write a program that takes three double number from Java console representing, respectively, the three coefficients a, b. and c of a quadratic equation

```
import java.util.Scanner
public class Exercise 2 {
    public static void main (String [] strings)
{
    Scanner input = new Scanner(System.in);
    System.out.println("Input a: ");
    double a = input.nextDouble();
    System.out.println("Input b: ");
    double b = input.nextDouble();
    System.out.println("Input c:");
    double c = input.nextDouble();

double result = b*b - 4.0 *a*c;

if (result >0.0){
    double r1 = (-b + Math.pow(result, 0.5))/(2.0*a);
    double r2 = (-b-Math.pow(result,0.5))/(2.0*a);
    System.out.println("The roots are" + r1 + "and't
            r2);
```

```
} else if (result == 0.0) {
    double r1 = -b/(2.0*a);
    System.out.println("The root is " +r1);
} else {
    System.out.println("The equation has no real
    root");
  }
 }
}
```

Write Program to find domain name of URL starts with www and ends with .com and find the test cases and coverage of code.

```
import java.util.*
import java.lang.*
import java.net.*

public class GetDomainName
{
   public static void main(String args[])
   {
      String domain = getUrlDomainName("www.google.com");
      System.out.println(domain);
   }

   public static String getUrlDomainName(String url) throws URISyntaxException{

      URI uri = new URI(url);
      String domain = uri.getHost();
      return domain.startsWith("www.") ?
         domain.substring(4, domain.indexOf(".com")) :
            domain;
   }
}
```

```
@Test
void shouldFindDomainName(){
assert GetDomainName.getUrlDomainName('
        https://example.com/path1')
assert GetDomainName.getUrlDomain Name
        ("https://google.com");
}
```

Write program that reads two words representing passwords from the java console and output the number of characters in the smaller of the two.

```
import java.util.*;
import java.lang*;
import java.io.*;
public class FindPassword
{
    public static void main(String args[])
{
  // Scanner Keyboard = new Scanner(System.in)

    Console consl = System.console();
    if (consl == null) {
            System.out.println("No console available
            return;
    }
  // Read line
    String str = consl.readline("Enter Password 1:")
// print username
    String str2 = consl.readline("Enter Password 2");
```

```
int strLen = str.length();
int str2Len = str2.length();

String shorter = "";
if (strLen > str2Len)
    shorter = str2
else if (strLen < str2Len)
    shorter = str
else
    System.out.println("both have equal length");

System.out.println("shorter password is" +shorter)
}
```

Aryabhatta Engineering College
Ajmer

Date : ..... Page No .....

Name : ..... Class : ..... Group ..... Roll No. .....

Write a program for a calculator and find the test case and coverage and Def-use-graph.

```
Package com.testingdocs.calculator;

Public class calculator {
    //no-arg constructor
  public calculator () {
}
public int add (int a, int b)
 {
   return a+b;
 }
public int subtract (int a, int b)
{
   return a-b;
}
public log multiply (int a, int b)
{
   return a*b;
}
Public double devide (int a, int b) {
 double result;
if (b==0) {
```

```
throw new IllegalArgumentException ("Divisor cannot divided by
                                    zero")
}
else {
    result = Double.valueOf(a) / Double.valueOf(b);
  }
  return result;
}
}
```

Calculator Test Case

```
Package com.testingdocs.calculator.tests;

import com.testingdocs.calculator.Calculator;
import.org.junit.test;
import.org.junit.Assert;
import.org.junit.Before;

Public class CalculatorTest {
Private calculator objCalculatorTest;

@Before
```

```
Public void setUp (){
  objCalculatorTest = new Calculator ();
}

@ Test
 Public void testAdd (){
   int a = 15; int b = 20;
  int expectedResult = 35;
long result = objCalculatorUnderTest.add (a,b);
 Asserts.assertEquals (expectedResult, result);
}

@ Test
Public void testSubtract (){
  int a = 25; int b = 20;
 int expectedResult = 5;
 long result = obj calc Under Test.Substract (a,b);
Assert.assertEquals (Expected result, result);
                    (ExpectedResult, result);
 }
```

```
@ Test

Public void testDevide () {
 int a = 56;  int b = 10;
douple expectedResult = 5.6;
double result = objCalcUnderTest . divide (a,b);
Assert. assertEquals (expectedResult, result, 0.00005);
}

@ Test  (expected = Illegal ArgumentException . class)

Public void test dividedbyZero () {
  int a = 15 ; int b = 0;
   objCalcUnderTest. divide (a,b);
 }
}
```