**Assignment Day 12**

**Duration 90 Min**

**3.  Write a code for solving quadratic equations, given the coefficients’ a, b, c for a standard quadratic equation of type**

The roots would be of type

Please raise exceptions for imaginary values.

Build a test case for the same & raise exceptions for imaginary values.

**package** com.verizon.day3.calci.tests;

**import** java.util.Scanner;

**public** **class** QuadraticEquation {

**public** **double**[] roots(**double** a,**double** b,**double** c) **throws** Exception

{

**double** root1, root2, root,d;

Scanner s = **new** Scanner(System.***in***);

System.***out***.println("Given quadratic equation:ax^2 + bx + c");

System.***out***.print("Enter a:");

a = s.nextInt();

System.***out***.print("Enter b:");

b = s.nextInt();

System.***out***.print("Enter c:");

c = s.nextInt();

System.***out***.println("Given quadratic equation:"+a+"x^2 + "+b+"x + "+c);

d = (b \* b) - (4 \* a \* c);

**if**(d > 0)

{

System.***out***.println("Roots are real and unequal");

root1 = ( - b + Math.*sqrt*(d))/(2\*a);

root2 = (-b - Math.*sqrt*(d))/(2\*a);

System.***out***.println("First root is:"+root1);

System.***out***.println("Second root is:"+root2);

}

**else** **if**(d == 0)

{

System.***out***.println("Roots are real and equal");

root = (-b)/(2\*a);

System.***out***.println("Roots:"+root+root);

}

**else**

{

**throw** **new** Exception("Roots are imaginary");

}

s.close();

**return** **null**;

}

}

package com.verizon.day3.employee.dao.test;

import static org.junit.Assert.\*;

import org.junit.AfterClass;

import org.junit.BeforeClass;

import org.junit.Ignore;

import org.junit.Test;

import com.verizon.day3.calci.Calculator;

import com.verizon.day3.calci.tests.QuadraticEquation;

public class QuadraticEquationTest {

static QuadraticEquation q;

@BeforeClass

public static void setUpBeforeClass() throws Exception {

q = new QuadraticEquation();

}

@AfterClass

public static void tearDownAfterClass() throws Exception {

q = null;

}

@Ignore

@Test

public void testRoots() throws Exception {

double expected[] = { -2.0, -2.0 };

double result[] = q.roots(1.0, 4.0, 4.0);

assertEquals(expected, result);

}

@Test(expected = java.lang.Exception.class)

public void testNegativeAdd() throws Exception {

q.roots(-10.0, 5.0, 5.0);

}

}