

1. Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$.

Read in a, b, c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

papergrid
Date: / /

```
import java.util.Scanner;
import math.java.lang.Math;
class Quadratic
{
    public static void main (String ss[])
    {
        int a, b, c;
        float d, r1, r2;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter a, b and c");
        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();

        d = b*b - 4*a*c;
        if (d > 0)
        {
            System.out.println ("The roots are real and unequal");
            r1 = (-b + Math.sqrt (d)) / (2 * a);
            r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.printf ("Root 1 = %f Root 2 = %f", r1, r2);
        }
        else if (d == 0)
        {
            System.out.println ("The roots are real and equal");
            r1 = (-1 * b) / (2 * a);
            r2 = r1;
            System.out.printf ("r1 = %f r2 = %f", r1, r2);
        }
        else
        {
            System.out.println ("The discriminant " +
                "+ d + " is negative :: no real solution");
        }
    }
}
```

Command Prompt

```
C:\Users\HARSHITHA RM\Desktop\oop>java Quadratic
Enter a,b,c:
78 56 34
Roots are imaginary as d(-7472.0)is negative therefore no real solution

C:\Users\HARSHITHA RM\Desktop\oop>java Quadratic
Enter a,b,c:
23 48 76
Roots are imaginary as d(-4688.0)is negative therefore no real solution

C:\Users\HARSHITHA RM\Desktop\oop>java Quadratic
Enter a,b,c:
34 56 11
Roots are real and unequal
root 1 =-0.23 root 2=-1.42
C:\Users\HARSHITHA RM\Desktop\oop>java Quadratic
Enter a,b,c:
1 2 1
Roots are real and equal
root 1 =-1.00 root 2=-1.00
C:\Users\HARSHITHA RM\Desktop\oop>
```

2. Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```

import java.util.*;
class Student
{
    private int n, credit[], gp[];
    private String name, msn;
    double sum, sgpa, marks[];
    Student()
    {
        name = "";
        msn = "";
    }
    void accept()
    {
        Scanner sc = new Scanner (System.in);
        System.out.println("Enter the number of the subjects");
        n = sc.nextInt();
        credit = new int[n+1];
        marks = new double[n+1];
        System.out.println("Enter usn and name of the student");
        usn = sc.next();
        name = sc.next();
        System.out.println("Enter credit & marks in each subject");
        for (int i = 1; i <= n; i++)
        {
            System.out.println("Enter Subject " + i + " credit and marks");
            credit[i] = sc.nextInt();
            marks[i] = sc.nextDouble();
        }
    }
}

```

```
void calculate()
```

```
{  
    gp = new int [m+1];
```

```
    int cre = 0;
```

```
    for (int i=1; i<=m; i++)
```

```
        if (marks[i] >= 90)
```

```
            gp[i] = 10;
```

```
        else if (marks[i] >= 80 && marks[i] < 90)
```

```
            gp[i] = 9;
```

```
        else if (marks[i] >= 70 && marks[i] < 80)
```

```
            gp[i] = 8;
```

```
        else if (marks[i] >= 60 && marks[i] < 70)
```

```
            gp[i] = 7;
```

```
        else if (marks[i] >= 50 && marks[i] < 60)
```

```
            gp[i] = 5;
```

```
        else if (marks[i] < 40)
```

```
            gp[i] = 0;
```

```
    sum = (double)(credit[i] * gp[i]) + sum;
```

```
    cre = cre + credit[i];
```

```
}
```

```
sgpa = sum / cre;
```

```
}
```

```
void display()
```

```
System.out.println ("User:" + user + "Name:"  
                    + name);
```

```
for (int i=1; i<=m; i++)
```

```
System.out.println ("Marks :"  
                    + marks[i] + " Grade Points :" + gp[i]);
```

```
System.out.println ("SGPA = " + sgpa);
```

```
}
```

```
class Student main
```

```
{
```

```
public static void main (String args[])
```

```
{
```

```
Student s1 = new Student();
```

```
s1.accept();
```

```
s1.calculate();
```

```
s1.display();
```

```
}
```

```
}
```

```
enter the number of the subjects:  
5  
enter usn and name of the student  
abs123 Riya  
enter credit and marks in each subject  
enter subject 1 credit and marks  
3 98  
enter subject 2 credit and marks  
4 92  
enter subject 3 credit and marks  
4 78  
enter subject 4 credit and marks  
5 67  
enter subject 5 credit and marks  
3 90  
usn:abs123 name:Riya  
marks:98.0 grade points:10  
marks:92.0 grade points:10  
marks:78.0 grade points:8  
marks:67.0 grade points:7  
marks:90.0 grade points:10  
sgpa=8.79Press any key to continue . . .
```

3. Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

```

import java.util.*;
class Book {
    String name, author;
    int nOp;
    double price;
    book()
    {
        name = "";
        author = "";
        nOp = 0;
        price = 0.00;
    }
    void accept()
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter name, ");
        author, nOp, price");
        name = sc.next();
        author = sc.next();
        nOp = sc.nextInt();
        price = sc.nextDouble();
    }
    public String toString()
    {
        return ("name = " + name + "\n author = " +
        author + "\n number of pages = " + nOp + "\n
        price = " + price );
    }
}

```

```

class Book main
{
    public static void main (String args[])
    {
        int n;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter Number of objects:");
        n = sc.nextInt();
        Book b[] = new Book[n];
        for (int i=0; i<n; i++)
        {
            b[i] = new Book();
            b[i].accept();
        }
        for (int i=0; i<n; i++)
        {
            System.out.println ("Details of book " + (i+1) + ":");
            System.out.println (b[i]);
        }
    }
}

```

```
C:\Windows\system32\cmd.exe
```

```
enter number of objects:
```

```
3
```

```
enter name
```

```
black beauty
```

```
enter author
```

```
anna sewell
```

```
enter number of pages and price
```

```
130 250
```

```
enter name
```

```
heidi
```

```
enter author
```

```
johanna spyri
```

```
enter number of pages and price
```

```
234 432
```

```
enter name
```

```
merchant of venice
```

```
enter author
```

```
shakespeare
```

```
enter number of pages and price
```

```
450 1200
```

```
details of book 1:
```

```
name=black beauty
```

```
author=anna sewell
```

```
number of pages=130
```

```
price=250.0
```

```
enter author
shakespeare
enter number of pages and price
450 1200

details of book 1:
name=black beauty
author=anna sewell
number of pages=130
price=250.0

details of book 2:
name=heidi
author=johanna spyri
number of pages=234
price=432.0

details of book 3:
name=merchant of venice
author=shakespeare
number of pages=450
price=1200.0
Press any key to continue . . .
```

4. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

lab - 4

papergrid
Date: / /

```
import java.util.Scanner;
```

```
abstract class Shape
```

```
{
```

```
    int a, b;
```

```
    abstract void printArea(int a, int b);
```

```
}
```

```
class Rectangle extends Shape
```

```
{
```

```
    void printArea(int a, int b)
```

```
{
```

```
    System.out.println("area of Rectangle  
+ (a * b));
```

```
}
```

```
class Triangle extends Shape
```

```
{
```

```
    void printArea(int a, int b)
```

```
{
```

```
    System.out.println("area = " + (a * b))
```

```
}
```

```
class Circle extends Shape
```

```
{
```

```
    void printArea(int a, int b)
```

```
{
```

```
    System.out.printf("area = %f  
3.14 * a * a))
```

```
}
```

```
}
```

Date: / /

class ShapeMain

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);
System.out.println("Enter l & b");
int l = sc.nextInt();
int b = sc.nextInt();

Rectangle R = new Rectangle();
R.printArea(l, b);

System.out.println("enter base
and height of triangle");

l = sc.nextInt();

b = sc.nextInt();

TriangleT = new Triangle();
T.printArea(l, b);

System.out.println("Enter radius
of circle");

int r = sc.nextInt();

```
enter length and breadth of rectangle
2 4
area of rectangle=8
enter base and height of triangle
2 10
area of triangle=10.0
enter radius of circle
3
area of circle=28.26Press any key to continue . . .
```

5. Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their

requirements. Include the necessary methods in order to achieve the following tasks:

- Accept deposit from customer and update the balance.
- Display the balance.
- Compute and deposit interest
- Permit withdrawal and update the balance
- Check for the minimum balance, impose penalty if necessary and update the balance.

Lab-5

```
import java.util.Scanner;
class account
{
    String name, accnum, accType;
    boolean checkbook = true;
    double balance;
    account() {}

    void input()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter name, acc number");
        name = sc.nextLine();
        accnum = sc.nextLine();
        System.out.println("Enter balance");
        balance = sc.nextInt();
    }

    void display()
    {
        System.out.println("\n name=" + name + "\n acc number=" + accnum +
                           "\n balance=" + balance + "\n account type=" + accType);
    }
}
```

```
void display()
```

```
{  
    System.out.printf("balance = %.2f", balance);  
}
```

```
}
```

```
}
```

```
class current extends Account
```

```
{
```

```
    current()
```

```
{
```

```
    acctype = "current";  
}
```

```
    double minbal = 5000;
```

```
    void check()
```

```
{
```

```
    double penalty = 100;
```

```
    if (balance < minbal).
```

```
        balance = balance - penalty;
```

```
        System.out.println("penalty is  
imposed");  
        balance = balance + balance;
```

```
    else
```

```
{
```

```
        System.out.println("penalty not  
imposed");  
}
```

```
}
```

void deposit()

{

Scanner sc = new Scanner(System.in);
 System.out.println("Enter amount
 to deposit:");
 double amt = sc.nextDouble();
 balance = balance + amt;

}

class Savacc extends Account

{

Savacc()

{

acctype = "Savings";

}

double ci;

void calcompound (int n, int t)

{

ci = balance * (Math.pow((1+0.21*n), t));

balance = balance + ci;

System.out.println("Compound

Interest = %.2f"; ci);

System.out.println("In balance: %.2f
 balance");

}

```
void deposit()
```

{

```
    System.out.println("Enter amt");
```

```
    double depamt = sc.nextDouble();
```

```
    balance = balance + depamt;
```

```
    System.out.println("balance = %.2f",  
                       balance);
```

}

}

```
class AccountMain @
```

{

```
System.in.
```

```
Scanner sc = new Scanner();
```

```
System.out.println("Enter 1 for  
saving acc 2 for current account");
```

```
int acctype = sc.nextInt();
```

```
Savacc s = new Savacc();
```

```
currentacc c = new currentacc();
```

```
if (acctype == 1)
```

{

```
System.out.println("Enter your  
details");
```

```
s.input();
```

```
s.displayDetails();
```

```
System.out.println("n, I ");
```

```
int n = sc.nextInt();
```

```
int I = sc.nextInt();
```

```
s.calcompound(n, I);
```

```
int n1 = 1;
```

```
while (n1 == 1) {
    System.out.println ("1. enter
    2. deposit 3. withdrawal 4. exit");
    int ne = sc.nextInt();
    if (ne == 1)
        s.deposit();
    else if (ne == 2)
        System.out.println ("Enter amount");
        double amt = sc.nextDouble();
        s.withdrawal(amt);
        s.display();
    else
        system.exit(0);
}
else if (accType == 2)
    System.out.println ("Enter your
    details");
    c.input();
    c.displayDetails();
    c.check();
    c.deposit();
    c.display();
```

```
C:\WINDOWS\system32\cmd.exe
enter 1 for savings account 2 for current account
1
enter your details:
enter name ,acc number
harshitha 12345
enter balance
6000

name=harshitha
accnumber=12345
balance=6000.0
account type=savings
number of times interest to be compounded per unit t(n),time in years
2 3
compound interest:10629.37
balance:16629.37
enter 1.deposit 2.withdrawl 3.exit
2
enter the amount :
1000
balance=15629.37
enter 1.deposit 2.withdrawl 3.exit
1
enter amount to deposit:
100

balance:15729.37
enter 1.deposit 2.withdrawl 3.exit
3
Press any key to continue . . .
```

```
C:\WINDOWS\system32\cmd.exe
enter 1 for savings account 2 for current account
2
enter your details:
enter name ,acc number
harshitha 12345
enter balance
3000

name=harshitha
accnumber=12345
balance=3000.0
account type=current
penalty is imposed
balance=2900.0
enter amt to deposit:
1000
balance=3900.00Press any key to continue . . .
```

6. Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of

Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
package CIE;  
import java.util.*;  
public class Student  
{  
    public String user, name;  
    public int sem;  
  
    public void input()  
{  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Enter user, name,  
        sem");  
        user = sc.nextLine();  
        name = sc.nextLine();  
        sem = sc.nextInt();  
    }  
  
    public void display()  
{  
        System.out.println("user = " + user + "  
        name = " + name + " sem = " + sem);  
    }  
  
    public class Internals extends Student  
{  
        public int a[];  
        public void input()  
{  
            Scanner sc = new Scanner(System.in);  
            System.out.println("Enter CIE marks");  
            a = new int[5];  
        }  
    }  
}
```

```
for (int i=0; i<5; i++)  
    a[i] = sc.nextInt();  
}
```

```
package SEE;  
import CIE.*;  
import java.util.*;
```

```
public class external extends CIE.Student
```

```
{  
    public int a[];  
    public void input()
```

```
    Scanner sc = new Scanner (System.in);  
    System.out.println ("Enter five  
    marks");
```

```
    a = new int [5];  
    for (int i=0; i<5; i++)  
        a[i] = sc.nextInt();
```

```
}
```

```
import CIE.*;  
import SEE.*;  
import java.util.*;
```

```
class Total
```

```
{
```

```
    public static void main (String [] args)
```

```
Scanner sc = new Scanner(System.in);
System.out.println("enter the number
of students");
int n = sc.nextInt();
CIE.internals in[] = new CIE.internals[n];
SEE.externals ex[] = new SEE.externals[n];
int total;
```

```
for(int j=0; j<n; j++)
{
    System.out.println("Enter student
details");
    in[j] = new CIE.internals();
    ex[j] = new SEE.externals();
    in[j].input();
    ex[j].input();
}
```

6
for (int j=0; j<n; j++)

```
{  
    in[j].display();  
    for (int k=0; k<5; k++)  
        s.o.println(in[j].a[k] + (ex[j].a[k]/2));  
}
```

```
C:\> C:\WINDOWS\system32\cmd.exe
enter number of students
2
enter1student details:
enter usn,name,sem
1bm19cs060 harshitha 3
enter cie marks(5 subjects) out of 50
49 45 48 50 49
enter see marks(5 subjects) out of 100
90 95
94 93 98
enter2student details:
enter usn,name,sem
1bm19cs050 riya 3
enter cie marks(5 subjects) out of 50
40 45 49 47 40
enter see marks(5 subjects) out of 100
90 91 95 90 98

usn=1bm19cs060  name=harshitha  sem=3

student 1 total marks:
94
92
95
96
98
usn=1bm19cs050  name=riya  sem=3

student 2 total marks:
85
90
96
92
89
Press any key to continue . . .
```

7. Write a program to demonstrate generics with multiple object parameters.

LAB

```
class Gen <T, G1, A>
```

{

T ob1;

G1 ob2;

A ob3;

Gen (T ob1, G1 ob2, A ob3)

{

ob1 = ob1;

ob2 = ob2;

ob3 = ob3;

}

T getOb1() {

} return ob1;

}

G1 getOb2() {

} return ob2;

}

A getOb3() { return ob3 }

void showType() {

System.out.println ("Type of T is " +

ob1.getClass().getName() + " & Type of

G1 is " + ob2.getClass().getName() + "

" & Type of A is " + ob3.getClass().getName());

}

}

```
class Grendemo  
{
```

```
    public static void main (String args[])
```

```
    {  
        Grendemo obj = new Grendemo (100, "harshitha", 'a');  
        obj.showType();
```

```
        int v = obj.getOb1();
```

```
        String str = obj.getOb2();
```

```
        char v3 = obj.getOb3();
```

```
        System.out.println ("Value 1 :" + v +  
                           "\n Value 2 :" + str + "\n Value 3 :" +  
                           v3);
```

```
}
```

```
}
```

```
C:\WINDOWS\system32\cmd.exe
Type of T is java.lang.Integer
Type of G is java.lang.String
Type of A is java.lang.Character
value: 100
value: harshitha
value: a
Press any key to continue . . .
```

8. Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception Wrong Age() when the input age=father’s age.

LAB.

papergrid
Date: / /

```
import java.util.*;  
class wrongage extends Exception  
{  
    private int detail;  
    wrongage (int a)  
    {  
        detail = a;  
    }
```

```
public String toString()  
{  
    return "wrongage [" + detail + "]";  
}
```

```
class father
```

```
{  
    int age;  
    father (int a)  
    {  
        age = a;  
    }
```

```
void checkage () throws wrongage  
{  
    if (age <= 0)  
        throw new wrongage (age);  
}
```

```
class son extends father
{
```

```
    int ages;
    son (int a, int b)
    {
        super (b);
        ages = a;
```

```
void check () throws wrong age
{
```

```
    if (ages == 0)
        throw new wrong age (ages);
```

```
    if (age <= ages)
        throw new wrong age (age);
```

```
System.out.println ("The given
ages are correct");
```

```
}
```

```
class agemain1
{
```

```
public static void main (String
args [])
{
```

```
try {
```

```
Scanner sc = new Scanner (System.in);
```

```
System.out.println ("Enter father's
age: ");
System.out.println ("Enter son's age: ");
```

```
int a1 = sc.nextInt ();
```

```
int a2 = sc.nextInt ();
```

```
son s = new son (a2, a1);
```

```
s.check (a1);
```

```
s.check ();
```

```
}
```

```
C:\WINDOWS\system32\cmd.exe
enter fathers and sons age:
10 20
Caught wrongage[10]
Press any key to continue . . .
```

9. Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

class Thread1 implements Runnable,

```
{  
    Thread t;  
    String a;  
    int b;
```

Thread1 (String s, int n)

```
{  
    a = s;  
    b = n;
```

```
t = new Thread (this, "NThread");  
System.out.println ("CT: " + t);  
t.start();
```

public void run()

```
{  
    try
```

```
    {  
        for (int n = 5; n > 0; n--)
```

```
        System.out.println (a);  
        Thread.sleep (b);
```

```
}
```

} catch (InterruptedException e)

```
    System.out.println ("Child Thread  
    Interrupted");
```

```
} System.out.println ("Child Thread  
    Quitting");
```

```
class Thread2
{
    public static void main (String args[])
    {
        Thread1 n1 = new Thread1 (""
            BMS college of Engineering", 10000);
        Thread1 n2 = new Thread1 ("(SE", 2000),
        System.out.println ("Back in main");
    }
}
```

```
C:\WINDOWS\system32\cmd.exe
CT:Thread[NThread,5,main]
CT:Thread[NThread,5,main]
BMS college of engineering
CSE
Back in main
CSE
CSE
CSE
CSE
BMS college of engineering
Child Thread quitting
BMS college of engineering
BMS college of engineering
BMS college of engineering
Child Thread quitting
Press any key to continue . . .
```

10. Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the

program would throw an Arithmetic Exception Display the exception in a message dialog box.

```
Date: / /  
import java.awt.*;  
import java.awt.event.*;  
  
public class divnum extends Frame  
    implements ActionListener  
{  
    String msg = "", res = "";  
    Button division;  
    JTextField num1p, num2p, result;  
  
    public divnum()  
    {  
        setLayout(new FlowLayout());  
        division = new Button("divide");  
        Label num1 = new Label("NUM 1:",  
            Label.RIGHT);  
        Label num2 = new Label("NUM 2:", label.  
            RIGHT);  
        Label result = new Label("Result:",  
            Label.RIGHT);  
        num1p = new JTextField('2');  
        num2p = new JTextField('2');  
        resul = new JTextField('2');  
  
        add(num1);  
        add(num1p);  
        add(num2);  
        add(num2p);  
        add(division);  
        add(result);  
        add(resul);  
        num1p.addActionListener(this);  
        num2p.addActionListener(this);
```

```
division.addActionListener(this);
result.addActionListener(this);
addWindowListener(new WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }
});
public void actionPerformed(ActionEvent ae) {
    String str = ae.getActionCommand();
    if (str.equals("divide")) {
        dividenum();
    }
}
void dividenum() {
    int n1, n2, n = 0;
    try {
        n1 = Integer.parseInt(num1.getText());
        n2 = Integer.parseInt(num2.getText());
        n = n1 / n2;
        result.setText(String.valueOf(n));
    }
}
```

```
catch (NumberFormatException ne) {
    msg = "Number format exception";
    dia d = new dia (this, "Exception");
    d.setVisible (true);
}

catch (ArithmaticException a) {
    msg = "Arithmatic Exception";
    dia d = new dia (this, "Exception");
    d.setVisible (true);
}
```

```
public static void main (String args) {
    divnum appwin = new divnum();
    appwin.setSize (new Dimension(250, 150));
    appwin.setTitle ("Division");
    appwin.setVisible (true);
}
```

```
class dia extends Dialog implements ActionListener {
    divnum bld;
    dia (Frame parent, String title) {
        super (parent, title, false);
        bld = (divnum) parent;
        setLayout (new FlowLayout ());
        setSize (300, 200);
        add (new Label (bld.msg));
        Button b;
```

```
add(lb = new Button("OK"));
    b. add ActionListener(this);
```

{

```
public void actionPerformed(
    ActionEvent ae)
```

{

```
    dispose();
```

{

{

