

Lab Program 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.

Program:

```
→ import java.util.*;
import java.lang.*;

public class Main
{
    public static void main(String args[])
    {
        System.out.println("Enter a, b, c of the quadratic eqn:");
        Scanner scan = new Scanner(System.in);
        double a = scan.nextDouble();
        double b = scan.nextDouble();
        double c = scan.nextDouble();
        double d = (b * b) - (4 * a * c);
        System.out.println("D = " + d);
        if (d == 0)
        {
            double r1 = -b / (2 * a);
            System.out.println("The roots are real and equal.");
            System.out.println(r1);
        }
        else if (d > 0)
        {
            double r1 = (-b + Math.sqrt(d)) / (2 * a);
            double r2 = (-b - Math.sqrt(d)) / (2 * a);
            System.out.println("The roots are real & distinct");
            System.out.println(r1 + " and " + r2);
        }
        else
        {
            System.out.println("There are no real roots.");
        }
    }
}
```

Output:

```
Enter a,b,c:  
4 -4 1  
Both the roots are real and equal.  
The roots are: 0.5
```

```
Enter a,b,c:  
2 1 -1  
The roots are real and distinct.  
The roots are: 0.5 and -1.0
```

```
Enter a,b,c:  
2 5 5  
There are no real roots.
```

Lab Program 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.

Program:

```
import java.util.Scanner;  
class Student  
{  
    private String USN;  
    private String name;  
    private double SGPA=0;  
    private int totalCredits=0;  
    private double[] marks;  
    private int [] credits;  
    private int i;  
  
    void accept()  
    {  
        int n;  
        Scanner scan = new Scanner(System.in);  
        System.out.println("Enter USN of the student");  
        USN = scan.nextLine();  
        System.out.println("Enter Name of student");  
        name = scan.nextLine();  
        System.out.println("Enter number of subjects: ");  
        n = scan.nextInt();
```

```

credits = new int[n];
marks = new double[n];
System.out.println("Enter credits for details of the
subjects:");
for (int i=0; i<n; i++)
{
    System.out.println("Enter credits for subject " + (i+1));
    marks[i] = scan.nextInt();
    calculate();
}

```

```

void calculate()
{
    totalCredits = totalCredits + credits[i];
    if (marks[i] >= 90 && marks[i] <= 100)
        SGPA = SGPA + (10 * credits[i]);
    else if (marks[i] >= 80)
        SGPA = SGPA + (9 * credits[i]);
    else if (marks[i] >= 70)
        SGPA = SGPA + (8 * credits[i]);
    else if (marks[i] >= 60)
        SGPA = SGPA + (7 * credits[i]);
    else if (marks[i] >= 50)
        SGPA = SGPA + (6 * credits[i]);
    else if (marks[i] >= 40)
        SGPA = SGPA + (5 * credits[i]);
    else
        System.out.println("Failed in this subject.");
}

```

```

void Display()
{
    System.out.println("Details of the Student");
    System.out.println("USN: " + USN);
    System.out.println("Name: " + name);
    System.out.println("SGPA of student " + (SGPA / totalCredits));
}

```

```

class Main
{
    public static void main (String args[])
    {
        Student s1 = new Student();
        s1.accept();
        s1.Display();
    }
}

```

Output:

```
Enter USN of the student
12df
Enter Name of the student
abc
Enter no of subjects
3
Enter details of the subjects:
Enter credits for subject 1
5
Enter marks for subject 1
78
Enter credits for subject 2
4
Enter marks for subject 2
68
Enter credits for subject 3
3
Enter marks for subject 3
89
Details of the Student
USN: 12df
Name :abc
SGPA of Student 8.0
```

Lab Program - 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.

Program:

```
import java.util.Scanner;
class Book
{
    String name, author;
    double price;
    int num_pages;
```

```
public void set_book()
{
    Scanner scan = new Scanner(System.in);
    System.out.println("Enter book name: ");
    name = scan.next();
    System.out.println("Enter author name: ");
    author = scan.next();
    System.out.println("Enter book's price: ");
    price = scan.Double();
    System.out.println("Enter no. of pages: ");
    num_pages = scan.Int();
}
```

```
public String toString()
{
    return (name + " " + author + " " + price + " " + num_pages);
}
```

```
class Main
{
    public static void main [String [] args]
    {
        int n;
        System.out.println("Enter no. of objects: ");
        Scanner scan = new Scanner(System.in);
        n = scan.nextInt();
        Book o[] = new Book[n];
        for (int i=0; i<n; i++)
        {
            o[i] = new Book();
            o[i].set_book();
        }
        for (int i=0; i<n; i++)
        {
            System.out.println(o[i]);
        }
    }
}
```

Output:

```
Enter the number of objects:  
2  
Enter book name:  
abc  
Enter author's name:  
pqr  
Enter book's price:  
340.67  
Enter number of pages:  
500  
Enter book name:  
xyz  
Enter author's name:  
qwe  
Enter book's price:  
500  
Enter number of pages:  
901  
abc pqr 340.67 500  
xyz qwe 500.0 901
```

Lab Program - 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.

Program:

```
import java.util.Scanner;  
abstract class shape{  
    double a,b;  
    abstract void printArea();  
}  
  
class rectangle extends shape{  
    void getdata(double x, double y){  
        a = x;  
        b = y; }
```

```
void printArea() {  
    System.out.println("Area of rectangle = " + (a * b)); }  
}
```

```
class Triangle extends Shape {  
    void getData (double x, double y) {  
        a = x;  
        b = y; }  
}
```

```
void printArea() {  
    double area = 0.5 * a * b;  
    System.out.println("Area of Triangle = " + area); }  
}
```

```
class Circle extends Shape {  
    void getData (double x) {  
        a = x; }  
}
```

```
void printArea() {  
    double area = 3.14 * a * a;  
    System.out.println("Area of circle = " + area); }  
}
```

```
public class Main {  
    public static void main (String [] args) {  
        Scanner scan = new Scanner (System.in);  
        Rectangle r = new Rectangle ();  
        Triangle t = new Triangle ();  
        Circle c = new Circle ();  
        System.out.println ("1. Rectangle \n 2. Triangle \n 3. Circle \nEnter  
        your choice: ");  
        ch = scan.nextInt();  
        switch (ch) {  
            case 1: System.out.println ("Enter length & breadth");  
                double l = scan.nextDouble();  
                double b = scan.nextDouble();  
                r.getData (l, b);  
                r.printArea ();  
                break;  
        }  
    }  
}
```

```
case 2: System.out.println("Enter base & height:");
        double b= scan.nextDouble();
        double h= scan.nextDouble();
        t.getdata(b,h);
        t.printArea(b,h);
        break;
```

```
case 3: System.out.println("Enter radius");
        double r1= scan.nextDouble();
        c.getdata(r1);
        c.printArea(r1);
        break
```

```
default: System.out.println("Invalid Input");
```

y
y

Output:

```
1. Rectangle
2. Triangle
3. Circle
Enter your choice: 2

Enter base and height:
4.5 6
Area of triangle = 13.5
```

```
1. Rectangle
2. Triangle
3. Circle
Enter your choice: 1

Enter length and breadth:
2 3
Area of rectangle = 6.0
```

```
1. Rectangle
2. Triangle
3. Circle
Enter your choice: 3

Enter radius:
3.6
Area of circle = 40.6944
```

Lab Program - 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

Program:

```
import java.util.Scanner;
import java.lang.Math;
class bank {
    String name;
    int acc_no;
    float bal, si;
    Scanner scan = new Scanner(System.in);
    void accept() {
        System.out.println("Enter name:");
        name = scan.next();
        System.out.println("Enter acc. no");
        acc_no = scan.nextInt();
        System.out.println("Enter balance");
        bal = scan.nextFloat(); }

    void display() {
        System.out.println("Name" + name + "Account no: " + acc_no + "\nBalance: " + bal); }
```

```
void deposit() {  
    System.out.println("Enter amount to be deposited: ");  
    int amt = scan.nextInt();  
    bal = bal + amt;  
    System.out.println("Available balance = " + bal); } }
```

```
class savings extends bank {  
    void cheque() {  
        System.out.println("In No cheque service"); } }
```

```
System.out.println("Minimum balance = Rs.1000.00\nEnter  
the amount to be withdrawn");  
amount = scan.nextFloat();  
if (amount > bal)  
    System.out.println("Balance is insufficient");  
else {  
    bal = bal - amount;  
    if (bal < 1000)  
        bal = bal - service_charge;  
    System.out.println("Service charge of Rs." +  
        service_charge + " is added.\nAvailable  
balance = " + bal); } }
```

```
else {  
    System.out.println(amount + " withdrawn\nAvailable  
balance = " + bal); } }
```

```
public class Main {  
    public static void main(String[] args) {  
        savings obj1 = new savings();  
        current obj2 = new current();  
        System.out.println("1: Savings In 2: Current In Enter your  
choice: ");  
        Scanner scan = new Scanner(System.in);  
        int ch = scan.nextInt();  
        switch (ch) {  
            case 1: obj1 = new savings();  
                    obj1.accept();  
                    obj1.display();
```

```
void simple_interest() {  
    System.out.println("Enter rate, no. of times interest  
    applied, time elapse");  
    int r = scan.nextInt();  
    int n = scan.nextInt();  
    int t = scan.nextInt();  
    si = bal * (1 + r/n);  
    System.out.println("Interest = Rs " + (Math.pow(si, n*t))); }  
}
```

```
void withdrawal() {  
    float amount;  
    System.out.println("No minimum balance required");  
    System.out.println("Enter the amount to withdraw");  
    amount = scan.nextFloat();  
    if (amount > bal)  
        System.out.println("Insufficient balance");  
    else {  
        bal = bal - amount;  
        System.out.println(amount + " withdrawn");  
        System.out.println("Available  
balance = " + bal); }  
}
```

3
class current extends bank {
 float service_charge = 100;

```
void cheque() {  
    System.out.println("Cheque service not available"); }  
}
```

```
void withdrawal() {  
    float amount;
```

```
obj1.cheque();
obj1.deposit();
obj1.simple-interest();
obj1.withdrawal();
break;
```

```
case 2: obj2 = new current();
obj2.accept();
obj2.display();
obj2.cheque();
obj2.deposit();
obj2.withdrawal();
break;
```

```
default: System.out.println("Invalid Input");
```

3.

Output:

```
1. Savings
2. Current
Enter your choice: 2

Enter the name of the account holder:
abc
Enter account number:
123
Enter account balance:
2000

Details
Name: abc
Account number: 123
Balance: 2000.0

Cheque service available

Enter the amount to be deposited:
500
Available balance= 2500.0
Minimun balance = Rs 1000.00
Enter the amount to be withdrawn: 1600
Service charge of Rs 100.0 is added.
Available balance= 800.0
```

```
1. Savings
2. Current
Enter your choice: 1

Enter the name of the account holder:
abc
Enter account number:
123
Enter account balance:
2000

Details
Name: abc
Account number: 123
Balance: 2000.0

No cheque service

Enter the amount to be deposited:
500
Available balance= 2500.0

Enter Rate of interest:
8
Enter the number of times interest applied per time period
3
Enter the time elapse:
4
Simple interest = Rs 3.167635202407837E46
No minimum balance required
Enter the amount to be withdrawn
1000
1000.0 withdrawn
Available balance= 1500.0
```

Lab Program - 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.

Program:

LAB - 6

Page _____

Student.java

```
package CIE;  
import java.util.Scanner;  
public class Student {  
    Scanner scan = new Scanner(System.in);  
    public String usn, name;  
    public int sem;  
  
    public void accept() {  
        System.out.println("Enter usn: ");  
        usn = scan.nextLine();  
        System.out.println("Enter name: ");  
        name = scan.nextLine();  
        System.out.println("Enter sem: ");  
        sem = scan.nextInt();  
    }  
}
```

```
public void display() {  
    System.out.println("Name: " + name);  
    System.out.println("USN: " + usn);  
    System.out.println("Sem: " + sem);  
}
```

Internals.java

```
package CIE;  
import java.util.Scanner;  
public class Internals extends CIE.Student {  
    Scanner scan = new Scanner(System.in);  
}
```

```
public int ciem[] = new int[5];
```

```
public void accept(){
```

```
for(int i=0; i<5; i++) {
```

```
System.out.println("Enter cie marks of  
subject "+(i+1));
```

```
ciem[i] = scan.nextInt();
```

}

}

Externals.java

```
package SEE;
```

```
import CIE.*;
```

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

```
public class Externals extends CIE.Student {
```

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

```
public int secm[] = new int[5];
```

```
public void accept(){
```

```
for(int i=0; i<5; i++) {
```

```
System.out.println("Enter sec marks of subject"  
+(i+1));
```

```
secm[i] = scan.nextInt();
```

3

3

TotalMarks.java

```

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

class TotalMarks {
    public static void main (String [] args) {
        int i, j, n;
        Scanner scan = new Scanner (System.in);
        int total [] = new int [5];
        System.out.println ("Enter no. of stds : ");
        n = scan.nextInt();
        CIE.Student s [] = new CIE.Student [n];
        CIE.Internals ci [] = new CIE.Internal [n];
        SEE.Externals xe [] = new SEE.Externals [n];
        for (i=0; i<n; i++) {
            System.out.println ("Enter student detail - ");
            s[i] = new CIE.Student ();
            s[i].accept ();
            ci[i] = new CIE.Internals ();
            ci[i].accept ();
            xe[i] = new SEE.Externals ();
            xe[i].accept ();
        }
        for (i=0; i<n; i++) {
            System.out.println ("Details of student " +(i+1));
            s[i].display ();
            for (j=0; j<5; j++) {
                total[j] = ci[i].clem[j] + xe[i].seem[j];
            }
        }
    }
}

```

System.out.println("Total marks in subject "+
 (j+1)+": "+total[j]);

Output:

```
Enter the number of students:  

2  

Enter student 1 details-  

Enter usn:  

1bm19cs001  

Enter name:  

aakanksha  

Enter sem:  

3  

Enter cie marks of subject 1  

45  

Enter cie marks of subject 2  

34  

Enter cie marks of subject 3  

41  

Enter cie marks of subject 4  

30  

Enter cie marks of subject 5  

44  

Enter see marks of subject 1  

45  

Enter see marks of subject 2  

34  

Enter see marks of subject 3  

41  

Enter see marks of subject 4  

30  

Enter see marks of subject 5  

44  

Enter student 2 details-  

Enter usn:  

1bm19cs000  

Enter name:  

javaa  

Enter sem:  

5  

Enter cie marks of subject 1  

44  

Enter cie marks of subject 2
```

```
Enter cie marks of subject 2  

43  

Enter cie marks of subject 3  

42  

Enter cie marks of subject 4  

41  

Enter cie marks of subject 5  

40  

Enter see marks of subject 1  

44  

Enter see marks of subject 2  

43  

Enter see marks of subject 3  

42  

Enter see marks of subject 4  

41  

Enter see marks of subject 5  

40  

Details of student 1  

Name: aakanksha  

USN: 1bm19cs001  

Sem: 3  

Total marks in subject 1: 90  

Total marks in subject 2: 68  

Total marks in subject 3: 82  

Total marks in subject 4: 60  

Total marks in subject 5: 88  

  

Details of student 2  

Name: javaa  

USN: 1bm19cs000  

Sem: 5  

Total marks in subject 1: 88  

Total marks in subject 2: 86  

Total marks in subject 3: 84  

Total marks in subject 4: 82  

Total marks in subject 5: 80
```

Lab Program - 7

Write a program to demonstrate generics with multiple object parameters.

Program:

```
import java.util.Scanner;
class multiple<T,V> {
    T ob1;
    V ob2;

    multiple(T o1, V o2) {
        ob1 = o1;
        ob2 = o2;
    }

    void showTypes() {
        System.out.println("Type : " + ob1.getClass().getName());
        "In Type : " + ob2.getClass().getName();
    }

    T getob1() {
        return ob1;
    }

    V getob2() {
        return ob2;
    }
}

class Main {
    public static void main(String args[]) {
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter string, integer : ");
        String s = scan.nextLine();
        int i = scan.nextInt();

        System.out.println("Enter string, double : ");
        String st = scan.next();
        double d = scan.nextDouble();

        multiple<String, Integer> tgObj = new multiple<String, Integer>(s, i);
        tgObj.showTypes();
        String str = tgObj.getob1();
        System.out.println("Value : " + str);
        int v = tgObj.getob2();
        System.out.println("Value : " + v);
    }
}
```

```
multiple <String, Double> tObj = new multiple  
<String, Double>(st, d);  
tObj.showTypes();  
String strn = tObj.getob1();  
double vd = tObj.getob2();  
System.out.println("Value :" + strn + "\nValue :" + vd);
```

g

Output:

```
Enter any string:  
wed  
Enter any integer:  
23  
  
Enter any string:  
tue  
Enter any double:  
34.022  
Type of first: java.lang.String  
Type of second: java.lang.Integer  
value of set 1: wed  
value of set 1: 23  
  
Type of first: java.lang.String  
Type of second: java.lang.Double  
value of set 2: tue  
value of set 2: 34.022
```

Lab Program - 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.

Program:

```
import java.util.Scanner;
class fatherAgeException extends Exception {
    public String toString() {
        return ("Wrong Age! Father's age is less than 0");
    }
}

class sonAgeException extends Exception {
    int a, b;
    sonAgeException (int sage, int fage) {
        a = sage;
        b = fage;
    }
    public String toString() {
        if (a == b)
            return ("Son's age = father's age");
        else
            return ("Son's age > father's age");
    }
}

class Father {
    public int age1;
    Scanner scan = new Scanner(System.in);
    Father() {
        System.out.println("Enter father's age: ");
        age1 = scan.nextInt();
    }
    void ex1() throws fatherAgeException {
        if (age1 < 0)
            throw new fatherAgeException();
    }
}
```

```

class Son extends Father {
    public int age2;
    Son() {
        System.out.println("Enter son's age:");
        age2 = scan.nextInt();
    }
    void ex2() throws SonAgeException {
        if (age2 >= age1)
            throw new SonAgeException(age2, age1);
    }
}

class Main {
    public static void main(String[] args) {
        Son s = new Son();
        try {
            s.ex1();
        } catch (FatherAgeException e) {
            System.out.println(e);
        }
        try {
            s.ex2();
        } catch (SonAgeException e) {
            System.out.println(e);
        }
    }
}

```

Output:

```

Enter father's age: -4
Enter son's age: 3
Wrong Age!! Father's age is less than 0
Wrong Age!! Son's age is more than father's age

...Program finished with exit code 0

```

Lab Program - 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.

Program:

```
class bms implements Runnable
{
    Thread t1;
    String name, ab;

    bms(String name)
    {
        ab = name;
        t1 = new Thread(this, ab);
        t1.start();
    }

    public void run()
    {
        try
        {
            for(int i=5; i>0; i--)
            {
                if(ab == "BMS")
                {
                    System.out.println("BMS College of Engineering");
                    Thread.sleep(10000);
                }
                else
                {
                    System.out.println("CSE");
                    Thread.sleep(2000);
                }
            }
        }
        catch(InterruptedException e)
        {
            if(ab == "BMS")
                System.out.println("BMS interrupted");
            else
                System.out.println("CSE interrupted");
            System.out.println("Exiting : " + t1);
        }
    }

    public class Main
    {
        public static void main(String args[])
        {
            new bms("BMS");
            new bms("CSE");
        }
    }
}
```

Output:

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
Exiting :Thread[CSE,5,main]
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
Exiting :Thread[BMS,5,main]

...Program finished with exit code 0
Press ENTER to exit console.□
```

Lab Program - 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 ArithmeticException. Display the exception in a message dialog box.

Program:

```
import java.awt.*;
import java.awt.event.*;

public class lab extends Frame implements ActionListener {
    JTextField Num1, Num2;
    JButton divide = new JButton("DIVIDE");
    float result = 0;
    String msg = "";
    Dialog d;
    public lab() {
        setLayout(new FlowLayout());
        Num1 = new JTextField(10);
        Num2 = new JTextField(10);
        Label result = new Label("Num1:", Label.RIGHT);
        Label msg = new Label("Num 2:", Label.RIGHT);
        add(new Label("Num1:"));
        add(Num1);
        add(new Label("Num 2:"));
        add(Num2);
        add(divide);
        divide.addActionListener(this);
        addWindowListener(new WinAdapter());
    }
}
```

```
3
public void actionPerformed(ActionEvent ae) {
    if (ae.getSource() == divide) {
        try {
            int n1 = Integer.parseInt(Num1.getText());
            int n2 = Integer.parseInt(Num2.getText());
            result = n1/n2;
        }
    }
}
```

```
        catch (NumberFormatException) {
            dial d1= new dial ("Number Format Exception");
            d1.setVisible (true);
        }
        catch (ArithmaticException exception) {
            dial d2= new dial ("Arithmatic Exception");
            d2.setVisible (true);
        }
        repaint ();
    }
    public void paint (Graphics g){
        g.drawString ("Result : " + String.valueOf(result), 20, 100);
    }
}
```

```
public static void main (String [] args) {
    lab p= new lab();
    p.setSize (new Dimension (350, 300));
    p.setTitle ("Divide");
    p.setVisible (true);
}
```

```
class dial extends Dialog implements ActionListener {
    dial (String str){
        super (new Frame(), str, true);
        setLayout (new FlowLayout ());
        setSize (300, 200);
        addWindowListener (new WinAdapter ());
    }
}
```

```
public void actionPerformed (ActionEvent ae) {
    setVisible (true);
}
```

```
public void paint (Graphics g){
    g.drawString ("Error!", 10, 70);
}
```

```
class WinAdapter extends WindowAdapter {
    public void windowClosing (WindowEvent we){
        System.exit (0);
    }
}
```

Output:

