

OOJ LAB REPORT
VARSHA VIRUPAKSHA GUJJALA
1WA23CS034
3G

LAB PROGRAMS:

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 discriminate $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
import java.io.*;
import java.util.Scanner;

class qcal{
public static void main(String args[]){
int y=0;
Scanner sc=new Scanner(System.in);
System.out.println("General form of a quadratic equation is ax^2+bx+c=0");
do{
System.out.print("\nEnter value of a=");
int a=sc.nextInt();
System.out.print("Enter value of b=");
int b=sc.nextInt();
System.out.print("Enter value of c=");
int c=sc.nextInt();
float d=(float)(Math.pow(b,2)-4*a*c);

if(d<0){
System.out.println("There are no real solutions");}

else if(d==0){
System.out.println("it has one repeated root (2 equal roots):");
float r=-b/(2.0f*a);
System.out.println("x="+r);}

else{
System.out.println("it has two distinct roots:");
double r1=(-b+Math.sqrt(d))/(2*a);
System.out.println("x1="+r1);
double r2=(-b-Math.sqrt(d))/(2*a);
System.out.println("x2="+r2);
}
System.out.print("\nDo you want to calculate again? (yes=0 and no=1): ");
y = sc.nextInt();
}while(y==0);
}}
```

```
C:\Users\varsh\OneDrive\Desktop\java>javac qcal.java

C:\Users\varsh\OneDrive\Desktop\java>java qcal
General form of a quadratic equation is  $ax^2+bx+c=0$ 

Enter value of a=1
Enter value of b=3
Enter value of c=-10
it has two distinct roots:
x1=2.0
x2=-5.0

Do you want to calculate again? (yes=0 and no=1): 0

Enter value of a=1
Enter value of b=-5
Enter value of c=6
it has two distinct roots:
x1=3.0
x2=2.0

Do you want to calculate again? (yes=0 and no=1): 0

Enter value of a=2
Enter value of b=-8
Enter value of c=6
it has two distinct roots:
x1=3.0
x2=1.0

Do you want to calculate again? (yes=0 and no=1): 0

Enter value of a=1
Enter value of b=-4
Enter value of c=4
it has one repeated root (2 equal roots):
x=2.0

Do you want to calculate again? (yes=0 and no=1): 0

Enter value of a=1
Enter value of b=2
Enter value of c=5
There are no real solutions

Do you want to calculate again? (yes=0 and no=1): 1

C:\Users\varsh\OneDrive\Desktop\java>
```

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 & CGPA of a student.

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

```
class Subject {
```

```
int subM;
```

```
int cred;
```

```
int grade;
```

```
void setSubDet(int marks, int cred) {
```

```
this.subM = marks;
```

```
this.cred = cred;
```

```
if (subM >= 90) {
```

```
grade = 10; }
```

```
else if (subM >= 80) {
```

```
grade = 9; }
```

```
else if (subM >= 70) {
```

```
grade = 8; }
```

```
else if (subM >= 60) {
```

```
grade = 7; }
```

```
else if (subM >= 50) {
```

```
grade = 6; }
```

```
else if (subM >= 40) {
```

```
grade = 5; }
```

```
else {
```

```
grade = 0; }
```

```
}}
```

```
class Student {
```

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

```
Subject[] subjects = new Subject[8];
```

```
Student() {
```

```
for (int i = 0; i < subjects.length; i++) {
```

```
subjects[i] = new Subject(); }
```

```
}
```

```
void getMarks() {
```

```
for (int i = 0; i < subjects.length; i++) {
```

```
System.out.print("Enter marks for subject " + (i + 1) + ": ");
```

```
int marks = s.nextInt();
```

```
System.out.print("Enter credit for subject " + (i + 1) + ": ");
```

```
int cred = s.nextInt();
```

```
subjects[i].setSubDet(marks, cred); }
```

```
}
```

```

double calSGPA() {
double Score = 0;
int totalCred = 0;
double SGPA=0.0;
for (Subject subject : subjects) {
Score += (subject.grade * subject.cred);
totalCred += subject.cred; }

```

```

if (totalCred > 0) {
SGPA = Score / totalCred; }
else {
SGPA = 0; }
return SGPA;
}

```

```

public class StudentDetails {
public static void main(String[] arg) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter number of semesters: ");
int numSems = sc.nextInt();
Student[] students = new Student[ numSems];
double c=0.0;
String usn,name;
System.out.print("Enter USN: ");
usn = sc.next();
System.out.print("Enter Name: ");
name = sc.next();
for (int i = 0; i < numSems; i++) {

System.out.println("Enter details for semester" + (i + 1));
students[i] = new Student();
students[i].getMarks();
double s=students[i].calSGPA();
c+=s;
}
c=c/numSems;
for (int i = 0; i < numSems; i++) {
System.out.println("USN: " + usn);
System.out.println("Name: " + name);
System.out.println("SGPA for sem "+ (i+1)+": " + students[i].calSGPA());}

System.out.println("CGPA: " +c);
}
}
}

```

Output:

```
C:\Users\varsh\OneDrive\Desktop\java>java Student$StudentDetails
Enter number of semesters: 3
Enter USN: sdfg
Enter Name: bg
Enter details for semester1
Enter marks for subject 1: 81
Enter credit for subject 1: 4
Enter marks for subject 2: 94
Enter credit for subject 2: 4
Enter marks for subject 3: 83
Enter credit for subject 3: 3
Enter marks for subject 4: 84
Enter credit for subject 4: 3
Enter marks for subject 5: 90
Enter credit for subject 5: 3
Enter marks for subject 6: 88
Enter credit for subject 6: 1
Enter marks for subject 7: 97
Enter credit for subject 7: 1
Enter marks for subject 8: 85
Enter credit for subject 8: 1
Enter details for semester2
Enter marks for subject 1: 92
Enter credit for subject 1: 4
Enter marks for subject 2: 97
Enter credit for subject 2: 4
Enter marks for subject 3: 92
Enter credit for subject 3: 3
Enter marks for subject 4: 90
Enter credit for subject 4: 3
Enter marks for subject 5: 86
Enter credit for subject 5: 3
Enter marks for subject 6: 82
Enter credit for subject 6: 1
Enter marks for subject 7: 98
Enter credit for subject 7: 1
Enter marks for subject 8: 93
Enter credit for subject 8: 1
Enter details for semester3
Enter marks for subject 1: 56
Enter credit for subject 1: 4
Enter marks for subject 2: 62
Enter credit for subject 2: 4
Enter marks for subject 3: 72
Enter credit for subject 3: 3
Enter marks for subject 4: 73
Enter credit for subject 4: 3
Enter marks for subject 5: 73
Enter credit for subject 5: 2
Enter marks for subject 6: 82
Enter credit for subject 6: 1
Enter marks for subject 7: 51
Enter credit for subject 7: 1
Enter marks for subject 8: 72
Enter credit for subject 8: 1
USN: sdfg
Name: bg
SGPA for sem 1:9.4
USN: sdfg
Name: bg
SGPA for sem 2:9.8
USN: sdfg
Name: bg
SGPA for sem 3:7.315789473684211
CGPA: 8.838596491228072
```

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. Display the complete details of the book. Develop a Java program to create n book objects.

NOTE: 1: Use normal display method

2: Use Override toString method

Eg: public String toString() {Write code to display data

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

```
class Book {
```

```
String name;
```

```
String author;
```

```
double price;
```

```
int num_pages;
```

```
public Book(String name, String author, double price, int num_pages) {
```

```
this.name = name;
```

```
this.author = author;
```

```
this.price = price;
```

```
this.num_pages = num_pages; }
```

```
public void setName(String name) {
```

```
this.name = name;}
```

```
public void setAuthor(String author) {
```

```
this.author = author; }
```

```
public void setPrice(double price) {
```

```
this.price = price; }
```

```
public void setNumPages(int num_pages) {
```

```
this.num_pages = num_pages; }
```

```
public String getName() {
```

```
return name; }
```

```
public String getAuthor() {
```

```
return author; }
```

```
public double getPrice() {
```

```
return price; }
```

```
public int getNumPages() {
```

```
return num_pages; }
```

```
public String toString() {
```

```
return "\nBook name: " + name + "\n" + "Author: " + author + "\n" + "Price: $" + price + "\n" + "Number of  
pages: " + num_pages ; }
```

```
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner ob = new Scanner(System.in);  
        System.out.print("\nEnter number of books:");  
        int n = ob.nextInt();  
        ob.nextLine();  
        Book[] books = new Book[n];  
        for(int i = 0; i < n; i++) {  
            System.out.print("\nEnter name of the book " + (i + 1) + ":");  
            String name = ob.nextLine();  
            System.out.print("Enter name of the author:");  
            String author = ob.nextLine();  
            System.out.print("Enter price of the book:");  
            double price = ob.nextDouble();  
            System.out.print("Enter number of pages:");  
            int num_pages = ob.nextInt();  
            ob.nextLine();  
            books[i] = new Book(name, author, price, num_pages);  
        }  
        System.out.print("\nBOOKS-----\n");  
        for (int i = 0; i < n; i++) {  
            System.out.println(books[i].toString());  
        }  
    }  
}
```

```
C:\Users\varsh\OneDrive\Desktop\java>javac Main.java
```

```
C:\Users\varsh\OneDrive\Desktop\java>java Main
```

```
Enter number of books:2
```

```
Enter name of the book 1:home
```

```
Enter name of the author:vg
```

```
Enter price of the book:20
```

```
Enter number of pages:150
```

```
Enter name of the book 2:skills
```

```
Enter name of the author:bg
```

```
Enter price of the book:30
```

```
Enter number of pages:255
```

```
BOOKS-----
```

```
Book name: home
```

```
Author: vg
```

```
Price: $20.0
```

```
Number of pages: 150
```

```
Book name: skills
```

```
Author: bg
```

```
Price: $30.0
```

```
Number of pages: 255
```

```
C:\Users\varsh\OneDrive\Desktop\java>
```


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.

```
import java.util.*;
abstract class Shape{
    double x,y;
    Shape(double x, double y){
        this.x=x;
        this.y=y;
    }
    abstract void printArea();
}
class Rectangle extends Shape{
    Rectangle(double l, double b){
        super(l,b);
    }
    void printArea(){
        double a=x*y;
        System.out.println("Area of rectangle is: "+ a+ " square units.");
    }
}
class Triangle extends Shape{
    Triangle(double b,double h){
        super(b,h);
    }
    void printArea(){
        double at=0.5*x*y;
        System.out.println("Area of triangle is: "+ at+ " square units.");
    }
}
class Circle extends Shape{
    Circle(double r,double e){
        super(r,0);
    }
    void printArea(){
        double ac=3.14*x*x;
        System.out.println("Area of circle is: "+ ac+ " square units");
    }
}
}
}
public class AREAS{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter sides of rectangle:\nlength:");
        int l=sc.nextInt();
        System.out.print("breadth:");
        int b=sc.nextInt();
        Rectangle rec=new Rectangle(l,b);
        rec.printArea();
        System.out.print("\nEnter sides of triangle:\nbase:");
        double B=sc.nextDouble();
```

```
System.out.print("height:");
double h=sc.nextDouble();
Triangle tri=new Triangle(B,h);
tri.printArea();
System.out.print("\nEnter radius of circle:");
double r=sc.nextDouble();
Circle c=new Circle(r,0);
c.printArea();
}
}
```

```
C:\Users\varsh\OneDrive\Desktop\java>javac AREAS.java
```

```
C:\Users\varsh\OneDrive\Desktop\java>java AREAS
```

```
Enter sides of rectangle:
```

```
length:12
```

```
breadth:2
```

```
Area of rectangle is: 24.0 square units.
```

```
Enter sides of triangle:
```

```
base:15
```

```
height:7
```

```
Area of triangle is: 52.5 square units.
```

```
Enter radius of circle:4.5
```

```
Area of circle is: 63.585 square units
```

```
C:\Users\varsh\OneDrive\Desktop\java>|
```

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 Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

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

```
class Acc{  
    private String cname;  
    private int acc_no;  
    protected double bal;
```

```
    public Acc(String cname, int acc_no, double bal){  
        this.cname = cname;  
        this.acc_no = acc_no;  
        this.bal = bal;  
    }
```

```
    public double getBal(){  
        return bal;  
    }
```

```
    public void dep(double amt){  
        if (amt > 0){  
            bal+= amt;  
            System.out.println("Deposited: " + amt); }  
        else{  
            System.out.println("Deposit amount must be positive.");  
        }  
    }  
    public void wd(double amt)  
    {  
        if(amt<=getBal()){  
            bal-=amt;  
            System.out.println("withdrew:"+amt + " balance is:"+ bal); }  
        else  
            System.out.println("Insufficient funds!!");  
    }  
    public void displayBal(){  
        System.out.println("Current Balance: " + bal); }  
    }
```

```
class SavAcc extends Acc{  
    private double interestRate;
```

```

public SavAcc(String cName, int accNum, double initialBal, double interestRate){
    super(cName, accNum, initialBal);
    this.interestRate = interestRate;
}

public void computeAndDepositInterest(){
    double interest = getBal() * interestRate / 100;
    dep(interest);
}
}

class CurAcc extends Acc{
    private double minBal;
    private double serviceCharge;
    public CurAcc(String cName, int accNum, double initialBal, double minBal, double serviceCharge) {
        super(cName, accNum, initialBal);
        this.minBal = minBal;
        this.serviceCharge = serviceCharge;
    }
    public void checkMinBal(){
        if (getBal() < minBal) {
            System.out.println("Balance is below minimum");
            bal-=serviceCharge;
            System.out.println("Deducted service charge:" +serviceCharge);
            System.out.println("Balance after deduction is:"+bal);
        }
    }
}

public class Bank1{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.println("enter customer name:");
        String name=sc.nextLine();
        System.out.println("enter account number:");
        int acc_no=sc.nextInt();
        System.out.println("enter initial balance:");
        double bal=sc.nextDouble();
        System.out.println("enter minimum balance:");
        double min_bal=sc.nextDouble();
        System.out.println("enter interest rate:");
        double interest_rate=sc.nextDouble();
        System.out.println("enter service charge:");
        double service_charge=sc.nextDouble();
        System.out.println("Enter choice:\n 1.Current acc\n 2.Savings acc");
        int ch=sc.nextInt();
        System.out.println("Customer name is:"+ name+"\nAccount number:"+acc_no+"\nVaraha V G-1WA23CS034");

        switch(ch){
            case(1):
                System.out.println("account is current type");
                CurAcc ca = new CurAcc(name,acc_no,bal,min_bal,service_charge);
                do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");
                    int c=sc.nextInt();

```

```

ca.checkMinBal();
if(c==1){
System.out.println("enter amount to be deposited:");
double amt=sc.nextDouble();
ca.dep(amt); }
else if(c==2){
System.out.println("enter amount to withdraw:");
double amt=sc.nextDouble();
ca.wd(amt); }
else if(c==3){
ca.displayBal(); }
else
System.exit(0);
}while(true);

case(2):
System.out.println("account is savings type");
SavAcc sa=new SavAcc(name,acc_no,bal,interest_rate);
do{ System.out.println("enter choice:\n 1.deposit\n 2.withdraw\n 3.display balance");
int c1=sc.nextInt();
if(c1==1){
System.out.println("enter amount to be deposited:");
double amt=sc.nextDouble();
sa.dep(amt); }
else if(c1==2){
System.out.println("enter amount to withdraw:");
double amt=sc.nextDouble();
sa.wd(amt); }
else if(c1==3){
sa.computeAndDepositInterest();
sa.displayBal(); }
else{
System.exit(0);
}
}while(true);
}
}
}

```

OUTPUT 1:

```
C:\Users\varsh\OneDrive\Desktop\java>java Bank1
```

```
enter customer name:
```

```
vg
```

```
enter account number:
```

```
1122
```

```
enter initial balance:
```

```
10000
```

```
enter minimum balance:
```

```
500
```

```
enter interest rate:
```

```
5
```

```
enter service charge:
```

```
20
```

```
Enter choice:
```

```
1.Current acc
```

```
2.Savings acc
```

```
1
```

```
Customer name is:vg
```

```
Account number:1122
```

```
Varaha V G-1WA23CS034
```

```
account is current type
```

```
enter choice:
```

```
1.deposit
```

```
2.withdraw
```

```
3.display balance
```

```
1
```

```
enter amount to be deposited:
```

```
1000
```

```
Deposited: 1000.0
```

```
enter choice:
```

```
1.deposit
```

```
2.withdraw
```

```
3.display balance
```

```
2
```

```
enter amount to withdraw:
```

```
10501
```

```
withdrew:10501.0 balance is:499.0
```

```
enter choice:
```

```
1.deposit
```

```
2.withdraw
```

```
3.display balance
```

```
3
```

```
Balance is below minimum
```

```
Deducted service charge:20.0
```

```
Balance after deduction is:479.0
```

```
Current Balance: 479.0
```

```
enter choice:
```

```
1.deposit
```

```
2.withdraw
```

```
3.display balance
```

```
4
```

```
Balance is below minimum
```

```
Deducted service charge:20.0
```

```
Balance after deduction is:459.0
```

```
C:\Users\varsh\OneDrive\Desktop\java>|
```

OUTPUT 2:

```
C:\Users\varsh\OneDrive\Desktop\java>java Bank1
enter customer name:
sr
enter account number:
3344
enter initial balance:
10000
enter minimum balance:
2000
enter interest rate:
4
enter service charge:
50
Enter choice:
  1.Current acc
  2.Savings acc
2
Customer name is:sr
Account number:3344
Varaha V G-1WA23CS034
account is savings type
enter choice:
  1.deposit
  2.withdraw
  3.display balance
1
enter amount to be deposited:
5000
Deposited: 5000.0
enter choice:
  1.deposit
  2.withdraw
  3.display balance
2
enter amount to withdraw:
9000
withdrew:9000.0 balance is:6000.0
enter choice:
  1.deposit
  2.withdraw
  3.display balance
3
Deposited: 240.0
Current Balance: 6240.0
enter choice:
  1.deposit
  2.withdraw
  3.display balance
2
enter amount to withdraw:
4245
withdrew:4245.0 balance is:1995.0
enter choice:
  1.deposit
  2.withdraw
  3.display balance
3
Deposited: 79.8
Current Balance: 2074.8
enter choice:
  1.deposit
  2.withdraw
  3.display balance
4
C:\Users\varsh\OneDrive\Desktop\java>
```

6. Create a package CIE which has two classes- Student and Internals. The class Student 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.Scanner;

public class Student{
    String usn;
    String name;
    int sem;

    public void StudentDet(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = sc.nextLine();
        System.out.print("Enter Name: ");
        name = sc.nextLine();
        System.out.print("Enter Semester: ");
        sem = sc.nextInt();
    }

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

```
package CIE;
import java.util.Scanner;

public class Internals{
    public int[] internalM = new int[5];

    public void CIEmarks(){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Internal Marks for 5 subjects:");
        for (int i=0; i<5; i++){
            System.out.print("Subject " + (i+1) + ": ");
            internalM[i] = s.nextInt();
        }
    }
}
```

```
package SEE;
import CIE.Internals;
import CIE.Student;
import java.util.Scanner;

public class Externals extends Student{
    int[] seeM = new int[5];
}
```



```

int[] finalM = new int[5];
public Internals internals=new Internals();

public void SEEmarks(){
Scanner sc = new Scanner(System.in);
System.out.println("Enter SEE Marks for 5 subjects:");
for (int i=0; i<5; i++){
System.out.print("Subject "+(i+1)+" ": );
seeM[i] = sc.nextInt(); }
}

```

```

public void calcFinalM(){
for (int i=0; i<5; i++){
finalM[i] = internals.internalM[i] + seeM[i]; }
}

```

```

public void displayFinalM(){
displaySD();
System.out.println("Final Marks for 5 subjects:");
for (int i=0; i<5; i++){
System.out.println("Subject "+(i+1)+" ": "+finalM[i]); }
}
}

```

```

import SEE.Externals;
import CIE.Student;
import CIE.Internals;
import java.util.Scanner;

```

```

class Main1{
public static void main(String[] args){
Scanner s = new Scanner(System.in);
System.out.print("Enter number of students: ");
int n = s.nextInt();

```

```

Externals[] students = new Externals[n];

```

```

for (int i=0; i<n; i++){
System.out.println("\nEnter details for student " + (i+1) + ":");
students[i] = new Externals();
students[i].StudentDet();
students[i].internals.CIEMarks();
students[i].SEEmarks();
students[i].calcFinalM();
}

```

```

displaySD();
System.out.println("\nFinal Marks of Students:");
for (int i=0; i<n; i++){
System.out.println("\nStudent " + (i+1) + ":");
students[i].displayFinalM(); }
System.out.println("\n");
}}

```

```
C:\Users\varsh\OneDrive\Desktop\java>javac Main1.java
```

```
C:\Users\varsh\OneDrive\Desktop\java>java Main1
```

```
Enter number of students: 3
```

```
Enter details for student 1:
```

```
Enter USN: 1WA23CS034
```

```
Enter Name: VARSHA VG
```

```
Enter Semester: 1
```

```
Enter Internal Marks for 5 subjects:
```

```
Subject 1: 33
```

```
Subject 2: 38
```

```
Subject 3: 35
```

```
Subject 4: 32
```

```
Subject 5: 36
```

```
Enter SEE Marks for 5 subjects:
```

```
Subject 1: 90
```

```
Subject 2: 81
```

```
Subject 3: 88
```

```
Subject 4: 89
```

```
Subject 5: 89
```

```
Enter details for student 2:
```

```
Enter USN: 1BM23CS123
```

```
Enter Name: CD
```

```
Enter Semester: 1
```

```
Enter Internal Marks for 5 subjects:
```

```
Subject 1: 35
```

```
Subject 2: 36
```

```
Subject 3: 39
```

```
Subject 4: 40
```

```
Subject 5: 40
```

```
Enter SEE Marks for 5 subjects:
```

```
Subject 1: 96
```

```
Subject 2: 93
```

```
Subject 3: 98
```

```
Subject 4: 98
```

```
Subject 5: 97
```

```
Enter details for student 3:
```

```
Enter USN: 1WA23CS077
```

```
Enter Name: ZAYN
```

```
Enter Semester: 1
```

```
Enter Internal Marks for 5 subjects:
```

```
Subject 1: 30
```

```
Subject 2: 32
```

```
Subject 3: 29
```

```
Subject 4: 32
```

```
Subject 5: 36
```

```
Enter SEE Marks for 5 subjects:
```

```
Subject 1: 88
```

```
Subject 2: 87
```

```
Subject 3: 90
```

```
Subject 4: 85
```

```
Subject 5: 88
```

Final Marks of Students:

Student 1:

USN: 1WA23CS034

Name: VARSHA VG

Semester: 1

Final Marks for 5 subjects:

Subject 1: 123

Subject 2: 119

Subject 3: 123

Subject 4: 121

Subject 5: 125

Student 2:

USN: 1BM23CS123

Name: CD

Semester: 1

Final Marks for 5 subjects:

Subject 1: 131

Subject 2: 129

Subject 3: 137

Subject 4: 138

Subject 5: 137

Student 3:

USN: 1WA23CS077

Name: ZAYN

Semester: 1

Final Marks for 5 subjects:

Subject 1: 118

Subject 2: 119

Subject 3: 119

Subject 4: 117

Subject 5: 124

C:\Users\varsh\OneDrive\Desktop\java>

7. 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 WrongAge() when the input age<0. In Son class, implement a constructor that uses both father and son's age and throws an exception if son's age is >=father's age.

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

```
class WrongAgeException extends Exception{  
    public WrongAgeException(String message){  
        super(message);  
    }  
}
```

```
class Father{  
    int age;  
    public Father(int age) throws WrongAgeException{  
        if(age<0){  
            throw new WrongAgeException("Father's age cannot be negative."); }  
        this.age = age;  
    }  
}
```

```
class Son extends Father{  
    int sAge;  
    public Son(int fAge, int sAge) throws WrongAgeException {  
        super(fAge);  
        if(sAge<0){  
            throw new WrongAgeException("Son's age cannot be negative.");  
        }  
        if(sAge>=fAge){  
            throw new WrongAgeException("Son's age cannot be greater than or equal to Father's age.");  
        }  
        this.sAge=sAge;  
    }  
}
```

```
public class Ages{  
    public static void main(String args[]){  
        Scanner scanner = new Scanner(System.in);  
  
        try{  
            System.out.print("Enter age of Father: ");  
            int fAge = scanner.nextInt();  
            System.out.print("Enter age of Son: ");  
            int sAge = scanner.nextInt();  
            Father father = new Father(fAge);  
            Son son = new Son(fAge, sAge);  
            System.out.println("Father's age: " + father.age);  
            System.out.println("Son's age: " + son.sAge);  
        }  
        catch (WrongAgeException e){  
            System.out.println("Exception: " + e.getMessage());  
        }  
    }  
}
```

```
catch (Exception e){
System.out.println("Invalid input. Please enter numeric values.");
}

finally{
System.out.println("Done");
}
}
}
```

```
C:\Users\varsh\OneDrive\Desktop\java>javac Ages.java

C:\Users\varsh\OneDrive\Desktop\java>java Ages
Enter age of Father: 20
Enter age of Son: 21
Exception: Son's age cannot be greater than or equal to Father's age.
Done

C:\Users\varsh\OneDrive\Desktop\java>java Ages
Enter age of Father: 34
Enter age of Son: 33
Father's age: 34
Son's age: 33
Done

C:\Users\varsh\OneDrive\Desktop\java>java Ages
Enter age of Father: -34
Enter age of Son: -54
Exception: Father's age cannot be negative.
Done

C:\Users\varsh\OneDrive\Desktop\java>java Ages
Enter age of Father: 32
Enter age of Son: -1
Exception: Son's age cannot be negative.
Done
```

8. 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 CollegeThread extends Thread{
public void run(){
try{
while(true){
System.out.println("BMS College of Engineering");
Thread.sleep(10000); }
}
catch(InterruptedException e){
System.out.println("CollegeThread interrupted."); }
}
}
class CSEThread extends Thread{
public void run(){
try{
while(true){
System.out.println("CSE");
Thread.sleep(2000); }
}
catch (InterruptedException e){
System.out.println("CSEThread interrupted."); }
}
}

public class Threads{
public static void main(String[] args){
CollegeThread collegeThread = new CollegeThread();
CSEThread cseThread = new CSEThread();
collegeThread.start();
cseThread.start();
}
}
```

```
C:\Users\varsh\OneDrive\Desktop\java>javac Threads.java
```

```
C:\Users\varsh\OneDrive\Desktop\java>java Threads
```

```
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
|
```

9. 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. (Open ended program)

```
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class DivisionApp{
    public static void main(String[] args){
        JFrame frame = new JFrame("Integer Division App");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 2, 5, 5));

        JLabel labelNum1 = new JLabel("Num1:");
        JTextField textNum1 = new JTextField();
        JLabel labelNum2 = new JLabel("Num2:");
        JTextField textNum2 = new JTextField();

        JLabel labelResult = new JLabel("Result:");
        JTextField textResult = new JTextField();
        textResult.setEditable(false);

        JButton divideButton = new JButton("Divide");

        panel.add(labelNum1);
        panel.add(textNum1);
        panel.add(labelNum2);
        panel.add(textNum2);
        panel.add(labelResult);
        panel.add(textResult);
        panel.add(new JLabel());
        panel.add(divideButton);

        frame.add(panel);

        divideButton.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e){
                try{
                    int num1 = Integer.parseInt(textNum1.getText());
                    int num2 = Integer.parseInt(textNum2.getText());

                    int result = num1 / num2;
                    textResult.setText(String.valueOf(result));
                }
                catch (NumberFormatException ex){
```

```

JOptionPane.showMessageDialog(frame, "Please enter valid integers in both fields.", "Number Format Error",
JOptionPane.ERROR_MESSAGE);
}
catch (ArithmeticException ex){
JOptionPane.showMessageDialog(frame, "Division by zero is not allowed.", "Arithmetic Error",
JOptionPane.ERROR_MESSAGE);
}
}
});

frame.setVisible(true);
}
}

```

