VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming (23CS3PCOOJ)

Submitted by
Deepak S (1BM25CS422)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019

Aug-2025 to Jan-2026

B.M.S. College of Engineering.
Bull Temple Road, Bengaluru 560019
(Affiliated to Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Deepak S (1BM25CS422),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object-Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Dr. Seema Patil Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

Index

GitHub Link: *https://github.com/deepakscs25-pixel/23CS3POOJ*

Program 1

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

Code:

```
import java.util.Scanner;
import static java.lang.Math.sqrt;
class quadratic {
   public static void main (String args[])
   {

      Scanner n=new Scanner (System.in);
      double a,b,c;
      double d;
      System.out.println("Enter the value for a: ");
      a=n.nextInt();
      if(a==0) {
         System.out.println("Not a quadratic equation. Please enter a non-zero value for 'a'.");
         return;
      }

      System.out.println("Enter the value for b: ");
      b=n.nextInt();
      System.out.println("Enter the value for b:" );
      c=n.nextInt();

      d=b*b - 4*a*c;
      System.out.println("Output: "+d);
      if(d==0) {
         double r1= (-b)/(2*a);
         System.out.println("Roots are real and equal.");
         System.out.println("Root 1 and Root 2: "+r1);
      }
      else if(d>0) {
         double r1=((-b) + (Math.sqrt(d)))/(2*a);
         double r2=((-b) - (Math.sqrt(d)))/(2*a);
```

```
            System.out.println("Roots are Real and different.");
            System.out.println("Root 1:"+r1);
            System.out.println("Root 2:"+r2);

    else if (d<0) {
            System.out.println("Roots are imaginary.");
            double real=(-b)/(2*a);
            double imaginary=Math.sqrt(-d)/(2*a);
            System.out.println("Root 1:"+real);
            System.out.println("Root 2:"+imaginary);
        }
    }
}
```

Output:

```
PS D:\BMSCE\3rd sem Java\Lab Program 1> java quadratic
Enter the value for a: 0
Not a quadratic equation. Please enter a non-zero value for 'a'.
PS D:\BMSCE\3rd sem Java\Lab Program 1> java quadratic
Enter the value for a: 2
Enter the value for b: -3
Enter the value for c:4
Output: -23.0
Roots are imaginary.
Root 1:0.75
Root 2:1.1989578808281798
PS D:\BMSCE\3rd sem Java\Lab Program 1> java quadratic
Enter the value for a: 1
Enter the value for b: -2
Enter the value for c:1
Output: 0.0
Roots are real and equal.
Root 1 and Root 2: 1.0
PS D:\BMSCE\3rd sem Java\Lab Program 1> java quadratic
Enter the value for a: 4
Enter the value for b: 6
Enter the value for c:2
Output: 4.0
Roots are Real and different.
Root 1:-0.5
Root 2:-1.0
```

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.

Code:

```
import java.util.Scanner;

class Subject {
   int subjectMarks;
   int credits;
   int grade;
}
class Student{
   String name;
   String usn;
   double SGPA;
   Scanner s;
   Subject subject[];


Student(){
   subject =new Subject[8];
   for(int i=0;i<8;i++){
      subject[i] =new Subject();
   }
   s=new Scanner(System.in);
}

void getStudentDetails(){
   System.out.print("Enter name: ");
   this.name=s.nextLine();
   System.out.print("Enter usn: ");
   this.usn=s.nextLine();
}
```

```
void getMarks(){
   for(int i=0;i<8;i++){
      System.out.print("Enter marks for Subject:"+(i+1)+ ": ");
      subject[i].subjectMarks=s.nextInt();
      System.out.print("Enter credits for Subject"+(i+1)+": ");
      subject[i].credits=s.nextInt();

      if(subject[i].subjectMarks > 100){
         System.out.println("Error: Marks cannot exceed 100. Please start over.");
         return;
      }
      subject[i].grade=(subject[i].subjectMarks/10)+1;
      if(subject[i].grade == 11){
         subject[i].grade=10;
      }
      if(subject[i].grade <=4){
         subject[i].grade=0;
      }
   }
}

void computeSGPA(){
   double effectiveScore =0;
   int totalCredits = 0;

   for(int i =0;i<8;i++){
      effectiveScore += (subject[i].grade * subject[i].credits);
      totalCredits += subject[i].credits;
   }
   this.SGPA = effectiveScore / totalCredits;
}

void displayDetails(){
   System.out.println("\nStudent Details");
   System.out.println("Name: "+ this.name);
   System.out.println("USN: "+this.usn);
   System.out.printf("SGPA: %.2f\n",this.SGPA);
   }
}
```

```
public class SGPA_Calculator {
   public static void main(String[] args){
      System.out.println("SGPA Calculation Program");
      Student student1 = new Student();

      student1.getStudentDetails();
      student1.getMarks();
      student1.computeSGPA();
      student1.displayDetails();
   }
}
```

Output:

```
PS D:\BMSCE\3rd sem Java\Lab Program 2> java SGPA_Calculator
SGPA Calculation Program
Enter name: Deepak
Enter usn: 1BM25CS422-T
Enter marks for Subject:1: 95
Enter credits for Subject1: 3
Enter marks for Subject:2: 89
Enter credits for Subject2: 3
Enter marks for Subject:3: 92
Enter credits for Subject3: 4
Enter marks for Subject:4: 90
Enter credits for Subject4: 4
Enter marks for Subject:5: 88
Enter credits for Subject5: 4
Enter marks for Subject:6: 85
Enter credits for Subject6: 1
Enter marks for Subject:7: 99
Enter credits for Subject7: 1
Enter marks for Subject:8: 95
Enter credits for Subject8: 3

Student Details
Name: Deepak
USN: 1BM25CS422-T
SGPA: 9.65
```

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.

Code:

```
import java.util.Scanner;


class Books{
   String name;
   String author;
   int price;
   int numPages;

   Books(String name, String author, int price, int numPages){
      this.name=name;
      this.author=author;
      this.price=price;
      this.numPages=numPages;
   }

   public String getName(){
      return this.name;
   }

   public String getAuthor(){
      return this.author;
   }
   public int getNumPages(){
      return this.numPages;
   }

   public void setName(String nameString){
      this.name=nameString;
   }

   public void setAuthor(String authorString){
      this.author=authorString;
   }
```

```
    public void setPrice(int price){
        this.price=price;
    }

    public void setNumPages(int numPagesInt){
        this.numPages=numPagesInt;
    }

    @Override
     public String toString() {
        String details = "Book name: " + this.name + "\n" +
                         "Author name: " + this.author + "\n" +
                         "Price: " + this.price + "\n" +
                         "Number of pages: " + this.numPages + "\n";
        return details;
    }
}

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

        Books[] b=new Books[n];

        for(int i=0; i<n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");

            System.out.print("Enter name: ");
            String name=s.nextLine();

            System.out.print("Enter author: ");
            String author = s.nextLine();

            System.out.print("Enter price: ");
            int price=s.nextInt();

            System.out.print("Enter number of pages: ");
            int numPages=s.nextInt();
            s.nextLine();
```

```
            b[i] = new Books(name,author,price,numPages);
        }
        System.out.println("Book Details:");

        for(int i=0;i<n;i++){
            System.out.println("Book "+(i+1));
            System.out.println(b[i]);
        }
    }
}
```

Output:

```
PS D:\BMSCE\3rd sem Java> java library
Enter the number of books: 2

Enter details for Book 1:
Enter Book name: The Hobbit
Enter author: J.R.R. Tolkien
Enter price: 879
Enter number of pages: 310

Enter details for Book 2:
Enter Book name: Pride and Prejudice
Enter author: Jane Austen
Enter price: 675
Enter number of pages: 279
Book Details:
Book 1
Book name: The Hobbit
Author name: J.R.R. Tolkien
Price: 879
Number of pages: 310

Book 2
Book name: Pride and Prejudice
Author name: Jane Austen
Price: 675
Number of pages: 279
```

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.

Code:

```
import java.util.Scanner;

abstract class Shape {
   double dim1;
   double dim2;

   public Shape(double dim1,double dim2){
      this.dim1=dim1;
      this.dim2=dim2;
   }
   public abstract void printArea();
}

class Rectangle extends Shape{
   public Rectangle(double length,double width){
      super(length,width);
   }
   public void printArea(){
      double area=dim1*dim2;
      System.out.println("Area of Rectangle="+area);
   }
}
class Triangle extends Shape{
   public Triangle(double base,double height){
      super(base,height);
   }
   public void printArea(){
      double area= 0.5* dim1* dim2;
      System.out.println("Area of Triangle="+area);
   }
}
```

```
class Circle extends Shape{
   public Circle(double radius){
      super(radius,0);
   }
   public void printArea(){
      double area = 3.14*dim1*dim1;
      System.out.println("Area of circle="+area);
   }
}

public class ShapeDemo{
   public static void main(String[] args) {
      Scanner s=new Scanner(System.in);
      System.out.println("Enter Dimension of the rectangle");
      System.out.print("Enter Length:");
      double recLen=s.nextDouble();
      System.out.print("Enter Breadth:");
      double recWidt=s.nextDouble();
      Shape rect=new Rectangle(recLen,recWidt);


      System.out.println("\nEnter Dimension of the triangle(Base and Height):");
      System.out.print("Enter Base:");
      double triBase=s.nextDouble();
      System.out.print("Enter Height:");
      double triHeight=s.nextDouble();
      Shape tri=new Triangle(triBase,triHeight);

      System.out.println("\nEnter Dimension of the Circle(radius):");
      System.out.print("Enter Radius:");
      double r=s.nextDouble();
      Shape circle=new Circle(r);

      rect.printArea();
      tri.printArea();
      circle.printArea();

   }
}
```

Output:

```
PS D:\BMSCE\3rd sem Java\Lab Program 4> javac ShapeDemo.java
PS D:\BMSCE\3rd sem Java\Lab Program 4> java ShapeDemo
Enter Dimension of the rectangle
Enter Length:25
Enter Breadth:30

Enter Dimension of the triangle(Base and Height):
Enter Base:12
Enter Height:40

Enter Dimension of the Circle(radius):
Enter Radius:20
Area of Rectangle=750.0
Area of Triangle=240.0
Area of circle=1256.0
```

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.

Code:

```
import java.util.Scanner;

class Account {
   String name;
   int accno;
   String type;
   double balance;

   Account(String n, int a, String t, double b) {
      name = n;
      accno = a;
      type = t;
      balance = b;
   }

   void deposit(double amt) {
      balance += amt;
      System.out.println("Deposited: " + amt + ". Updated balance: " + balance);
   }

   void display() {
      System.out.println("Customer name: " + name);
      System.out.println("Account number: " + accno);
      System.out.println("Type of Account: " + type);
      System.out.println("Account Balance: " + balance);
   }
}

class SavAccount extends Account {
   SavAccount(String n, int a, double b) {
      super(n, a, "saving", b);
   }
```

```
  void interest() {
        double rate = 0.04; // 4% interest
        double intr = balance * rate;
        balance += intr;
        System.out.println("Interest added: " + intr + ". Updated balance: " + balance);
    }

    void withdraw(double amt) {
        if (amt <= balance) {
            balance -= amt;
            System.out.println("Withdrawn: " + amt + ". Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}


class CurAccount extends Account {
    CurAccount(String n, int a, double b) {
        super(n, a, "current", b);
    }

    void withdraw(double amt) {
        if (amt <= balance) {
            balance -= amt;
            System.out.println("Withdrawn: " + amt + ". Updated balance: " + balance);
            checkMinBalance();
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    void checkMinBalance() {
        double min = 2000;
        if (balance < min) {
            balance -= 100;
            System.out.println("Service charge imposed! Updated balance: " + balance);
        }
    }
}
```

```
class Bank {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        System.out.print("Enter customer name: ");
        String name = s.nextLine();

        System.out.print("Enter account number: ");
        int accno = s.nextInt();

        System.out.print("Enter account type (saving/current): ");
        String type = s.next();

        System.out.print("Enter initial balance: ");
        double bal = s.nextDouble();

        if (type.equalsIgnoreCase("saving")) {
            SavAccount sa = new SavAccount(name, accno, bal);
            menu(sa, s);
        } else {
            CurAccount ca = new CurAccount(name, accno, bal);
            menu(ca, s);
        }
    }

    static void menu(Account acc, Scanner s) {
        int choice;
        do {
            System.out.println("\n------MENU------");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Compute Interest for Savings Account");
            System.out.println("4. Display Account Details");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = s.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter the deposit amount: ");
                    double d = s.nextDouble();
                    acc.deposit(d);
                    break;
```

```
            case 2:
                System.out.print("Enter the withdrawal amount: ");
                double w = s.nextDouble();
                if (acc instanceof SavAccount)
                    ((SavAccount) acc).withdraw(w);
                else if (acc instanceof CurAccount)
                    ((CurAccount) acc).withdraw(w);
                break;
            case 3:
                if (acc instanceof SavAccount)
                    ((SavAccount) acc).interest();
                else
                    System.out.println("Interest not available for current account.");
                break;
            case 4:
                acc.display();
                break;
            case 5:
                System.out.println("Exiting...");
                break;
            default:
                System.out.println("Invalid choice.");
        }
    } while (choice != 5);
  }
}
```

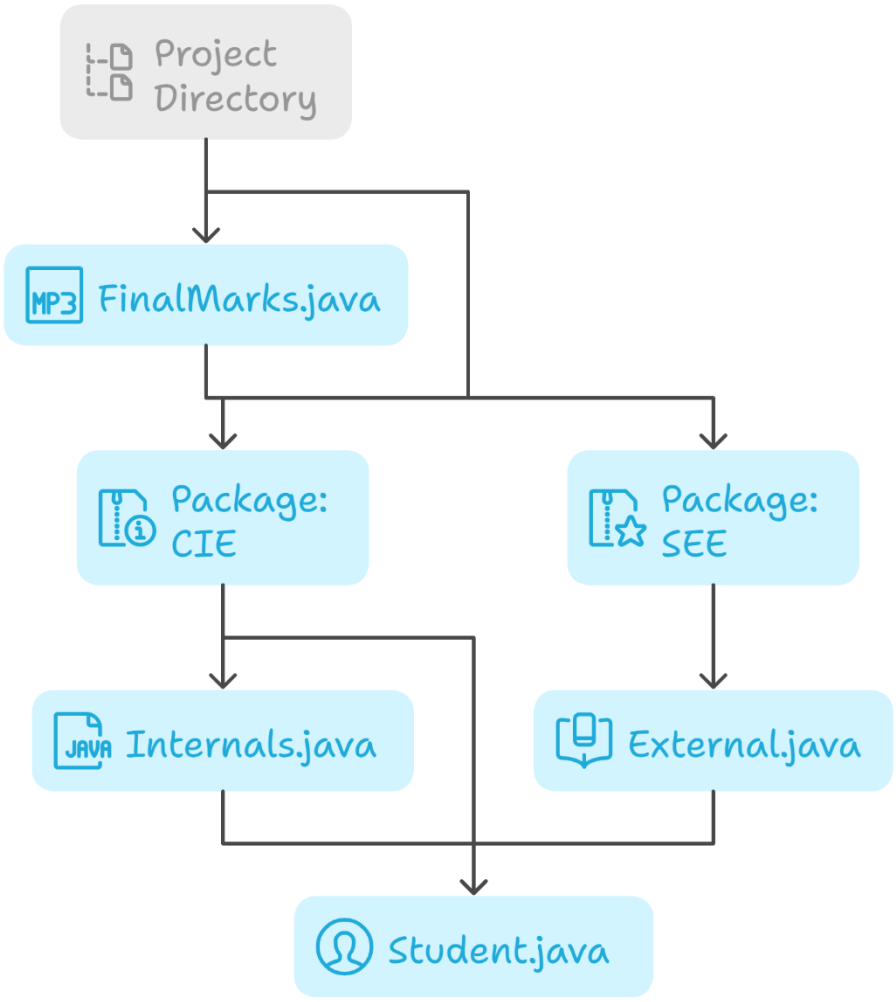
Output:

```
PS D:\BMSCE\3rd sem Java\Lab Program 5> java Bank
Enter customer name: Chandu
Enter account number: 98765432
Enter account type (saving/current): saving
Enter initial balance: 10000

------MENU------
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 2
Enter the withdrawal amount: 50000
Insufficient balance.

------MENU------
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 3
Interest added: 400.0. Updated balance: 10400.0

------MENU------
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 2
Enter the withdrawal amount: 5000
Withdrawn: 5000.0. Updated balance: 5400.0
```

```
PS D:\BMSCE\3rd sem Java\Lab Program 5> java Bank
Enter customer name: Deepak
Enter account number: 123456789
Enter account type (saving/current): current
Enter initial balance: 50000

------MENU------
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 3
Interest not available for current account.

------MENU------
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 2
Enter the withdrawal amount: 4000
Withdrawn: 4000.0. Updated balance: 46000.0

------MENU------
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 4
Customer name: Deepak
Account number: 123456789
Type of Account: current
Account Balance: 46000.0
```

Program 6:

Create a package CIE which has two classes- Student and Internals. The class Student has members like USN, name, sem. The class Internals derived from Student 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.



Code:

```
package CIE;

import java.util.Scanner;

public class Internals extends Student {
   protected int[] internalMarks = new int[5];

   public void inputInternalMarks() {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter Internal Marks (out of 50) for 5 subjects:");
      for (int i = 0; i < 5; i++) {
         System.out.print("Subject " + (i + 1) + ": ");
         internalMarks[i] = s.nextInt();
      }
   }

   public void displayInternalMarks() {
      System.out.println("Internal Marks:");
      for (int i = 0; i < 5; i++) {
         System.out.println("Subject " + (i + 1) + ": " + internalMarks[i]);
      }
   }
}

package SEE;

import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
   protected int[] seeMarks = new int[5];
   protected double[] finalMarks = new double[5];

   public Externals() {
      for (int i = 0; i < 5; i++) {
         seeMarks[i] = 0;
         finalMarks[i] = 0.0;
      }
   }

   public void inputSEEmarks() {
```

```
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE Marks (out of 100) for 5 subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            seeMarks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = internalMarks[i] + (seeMarks[i] / 2.0);
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        System.out.println("Final Marks in 5 Subjects:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Subject " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}

import SEE.*;
import java.util.*;

public class Final {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter number of students: ");
        int n = sc.nextInt();

        Externals[] students = new Externals[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Externals();
            System.out.println("\nEnter Details for Student " + (i + 1) + "");
            students[i].inputStudentDetails();
            students[i].inputInternalMarks();
            students[i].inputSEEmarks();
            students[i].calculateFinalMarks();
        }
```

```
        System.out.println("\n========== FINAL RESULTS ==========");
        for (int i = 0; i < n; i++) {
            students[i].displayFinalMarks();
        }
    }
}
```

Output:

```
PS D:\BMSCE\3rd sem Java\Lab Program 6> java Final
>>
Enter number of students: 1

--- Enter Details for Student 1 ---
Enter USN: 1BM25CS422
Enter Name: Deepak
Enter Semester: 3
Enter Internal Marks (out of 50) for 5 subjects:
Subject 1: 45
Subject 2: 50
Subject 3: 35
Subject 4: 40
Subject 5: 35
Enter SEE Marks (out of 100) for 5 subjects:
Subject 1: 90
Subject 2: 95
Subject 3: 89
Subject 4: 100
Subject 5: 95

========== FINAL RESULTS ==========
USN: 1BM25CS422
Name: Deepak
Semester: 3
Final Marks in 5 Subjects:
Subject 1: 90.0
Subject 2: 97.5
Subject 3: 79.5
Subject 4: 90.0
Subject 5: 82.5
-----------------------------------
```

Program 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 cases both father and son’s age and throws an exception if son’s age is >=father’s age.

Code:

```
import java.util.Scanner;

class WrongAge extends Exception{
   public WrongAge(String msg){
      super(msg);
   }
}

class InputScanner {
   Scanner s=new Scanner(System.in);
}

class Father extends InputScanner{
   int FatherAge;
   Father() throws WrongAge{
      System.out.print("Enter Father's Age:");
      FatherAge= s.nextInt();

      if (FatherAge<=0){
      throw new WrongAge("Age cannot Be Zero or Negative");
      }
   }


   void displayFather(){
      System.out.print("Fathers Age is "+FatherAge);
   }
}

class Son extends Father{
   int SonAge;

   Son() throws WrongAge{
      System.out.print("Enter Son's Age:");
```

```
        SonAge=s.nextInt();

        if(SonAge<=0){
           throw new WrongAge("Son's Age can not be Negative or Zero");
        }
        else if(SonAge>=FatherAge){
           throw new WrongAge("Son's Age cannot be greater than Fathers Age");
        }
     }

     void display(){
        super.displayFather();
        System.out.println("\nSon's Age is:"+SonAge);
     }
}

public class main{
   public static void main(String[] args) {
      try {

         Son obj= new Son();
         obj.display();
      }
      catch(WrongAge e){
         System.out.println("Exception: "+ e.getMessage());
      }
   }
}
```

Output:

```
PS C:\Users\Admin\Desktop\DEEPAK\Lab Program 7> java main
Enter Father's Age:0
Exception: Age cannot Be Zero or Negative
PS C:\Users\Admin\Desktop\DEEPAK\Lab Program 7> java main
Enter Father's Age:45
Enter Son's Age:0
Exception: Son's Age can not be Negative or Zero
PS C:\Users\Admin\Desktop\DEEPAK\Lab Program 7> java main
Enter Father's Age:45
Enter Son's Age:54
Exception: Son's Age cannot be greater than Fathers Age
PS C:\Users\Admin\Desktop\DEEPAK\Lab Program 7> java main
Enter Father's Age:45
Enter Son's Age:19
Fathers Age is 45
Son's Age is 19
PS C:\Users\Admin\Desktop\DEEPAK\Lab Program 7>
```

Program 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.

Code:

```
class BMSPrinter extends Thread{
   public void run(){
      try {
        for(int i=0;i<3;i++){
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000);
         }
      }
      catch (InterruptedException e) {
         System.out.println("Thread Interrupted");
      }
   }
}


class CSEPrinter extends Thread{
   public void run(){
      try {
         for(int i=0;i<10;i++){
            System.out.println("CSE");
            Thread.sleep(2000);
         }
      }
      catch (InterruptedException e) {
         System.out.println("Thread Interrupted");
      }
   }
}




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

```
        BMSPrinter t1=new BMSPrinter();
        CSEPrinter t2=new CSEPrinter();

        t1.start();
        t2.start();

    }
}
```

Output:

```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
PS C:\Users\Admin\Desktop\DEEPAK\Lab Program 8>
```