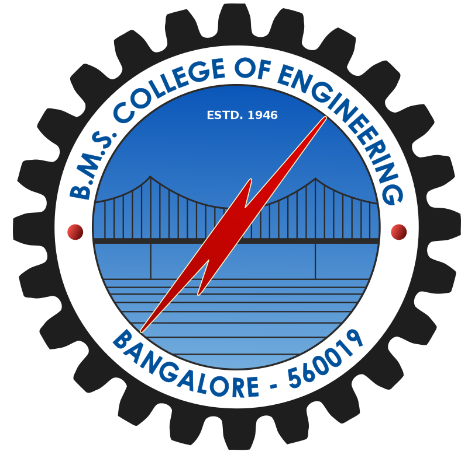
# B.M.S. COLLEGE OF ENGINEERING

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum) Bull Temple Road, Basavanagudi, Bengaluru – 560019



**LAB REPORT**

On

***Object Oriented Java Programming***

**(23CS3PCOOJ)**

Submitted By :

**C NEHA**

**1BM22CS074**

*In partial fulfilment of*

**BACHELOR OF ENGINEERING**

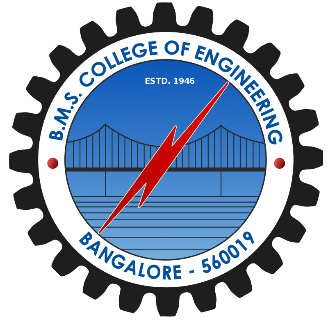
In

**COMPUTER SCIENCE AND ENGINEERING**

2023-24

# B.M.S. COLLEGE OF ENGINEERING

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum) Bull Temple Road, Basavanagudi, Bengaluru – 560019



**Department of**

**Computer Science & Engineering (CSE)**

CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Programming in Java (22CS3PCOOJ)” conducted by **C NEHA (1BM22CS074),** who is bonafide student at **B.M.S.College of Engineering**. It is in partial fulfilment for the award of **Bachelor of Engineering in Computer Science and Engineering** during the academic year 2023-24. The Lab report has been approved as it satisfies the academic requirements in respect of Object Oriented Programming in Java (22CS3PCOOJ) work prescribed for the said degree.

**C NEHA Shravya AR**

1BM22CS074 Assistant Professor

Department of CSE

BMSCE, Bengaluru-19

# Table of contents

|  |  |  |
| --- | --- | --- |
| **Sl.**  **No.** | **Program Title** | **Page No.** |
| 1 | Solution to Quadratic equation | 4-5 |
| 2 | Student details and Percentage calculation | 5-7 |
| 3 | Book Details | 8-9 |
| 4 | Calculating Area of different Shapes | 10-11 |
| 5 | Bank Details | 12-15 |
| 6 | CIE and SEE marks details(PACKAGES) | 15-19 |
| 7 | Exception Handling | 19-21 |
| 8 | Threads | 21-22 |

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 b 2 -4ac

is negative, display a message stating that there are no real solutions.

import java.util.Scanner;

import java.lang.Math;

class quadratic

{

public static void main(String agrs[])

 {

System.out.println("Name:C NEHA");

System.out.println("USN: 1BM22CS074");

    int a,b,c;

    System.out.println("enter the values of a,b,c respectively\n");

    Scanner s1= new Scanner(System.in);

    a = s1.nextInt();

    b = s1.nextInt();

    c = s1.nextInt();

    double d= b\*b - 4\*a\*c ;

    System.out.println("a = " + a +" b = " + b +" c = " + c);

    if(a==0) {System.out.println("not a quadratic equation");}

    else if( d>0)

    {

      System.out.println("the equation has two real and different solutions");

      double r1=(-b + Math.sqrt(d))/(2\*a);

      double r2=(-b - Math.sqrt(d))/(2\*a);

      System.out.println("r1 = " + r1);

      System.out.println("r2 = " + r2);

    }

   else if(d==0)

    {

      System.out.println("the equation has real and equal solutions");

      double r1= -b/(2\*a);

      double r2= -b/(2\*a);

      System.out.println("r1 = " + r1);

      System.out.println("r2 = " + r2);

    }

   else if(d<0)

    {

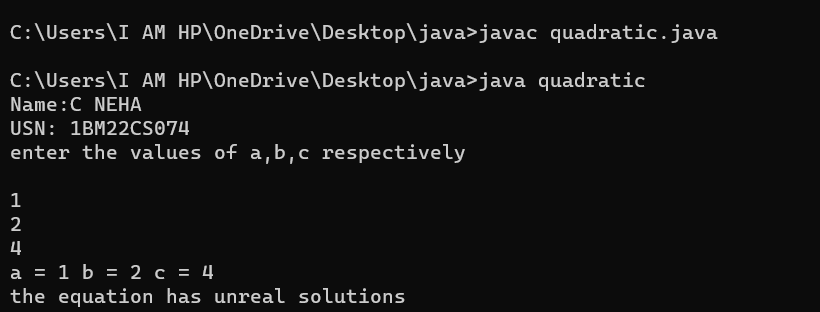
     System.out.println("the equation has unreal solutions");

    }

 }

}

**OUTPUT:**



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

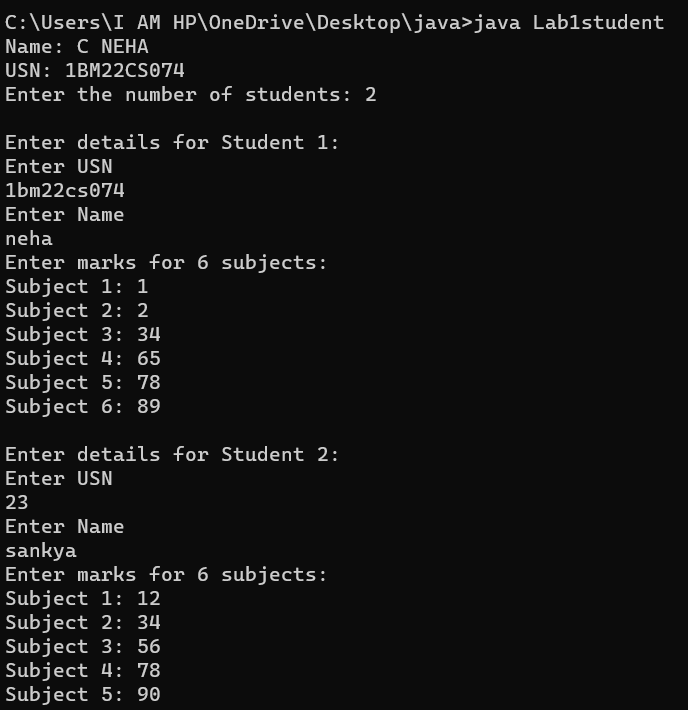
method to calculate percentage of a student.

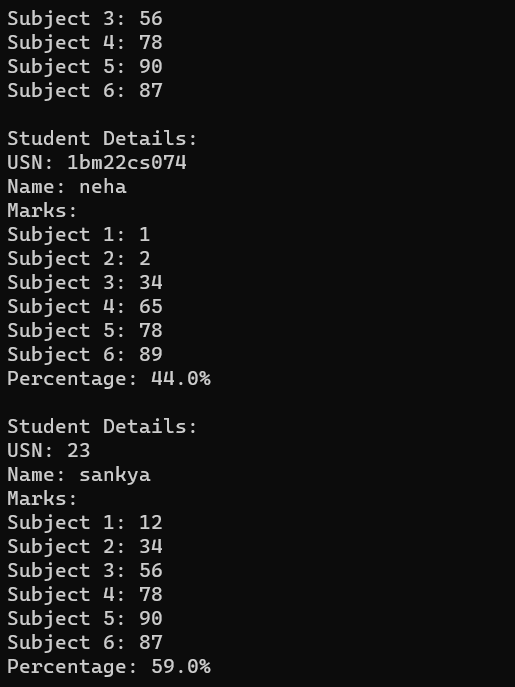
import java.util.Scanner;  
class Student{  
   String usn;  
   String name;  
   int marks[]= new int[6];  
     
   void Details()  
   {  
Scanner s=new Scanner(System.in);  
System.out.println("Enter USN");  
usn=s.next();  
System.out.println("Enter Name");  
name=s.next();  
System.out.println("Enter marks for 6 subjects:");  
for(int i = 0; i < 6; i++)  
{  
System.out.print("Subject " + (i + 1) + ": ");  
marks[i]=s.nextInt();  
}  
   }  
   double percentage()  
   {  
int total=0;  
for(int i=0;i<6;i++)  
{  
total+=marks[i];  
}  
double p=total/6;  
return p;  
   }  
   void display()  
   {  
System.out.println("\nStudent Details:");  
        System.out.println("USN: " + usn);  
        System.out.println("Name: " + name);  
        System.out.println("Marks:");  
        for(int i = 0; i < 6; i++)  
        {  
System.out.println("Subject " + (i + 1) + ": " + marks[i]);  
        }  
        System.out.println("Percentage: "+ percentage() + "%");  
   }  
}  
class Lab1student  
{  
public static void main(String args[]){  
Scanner s = new Scanner(System.in);

System.out.println("Name: C NEHA");

System.out.println("USN: 1BM22CS074");  
System.out.print("Enter the number of students: ");  
int n = s.nextInt();  
Student[] students = new Student[n];  
for (int i = 0; i < n; i++)  
{  
students[i] = new Student();  
System.out.println("\nEnter details for Student " + (i + 1) + ":");  
students[i].Details();  
}  
for (Student student : students)  
{  
student.display();  
}  
}  
}

**OUTPUT:**





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.

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.numPages = numPages;

        this.price = price;

    }

    public String toString() {

        String name, Author, price, numPages;

        name = "Book name: " + this.Name + "\n";

        Author = "Author name: " + this.Author + "\n";

        numPages = "Number of pages: " + this.numPages + "\n";

        price = "Price: " + this.price + "\n";

        return name + Author + numPages + price;

    }

}

class Main {

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        int n;

        String Name;

        String Author;

        int price;

        int numPages;

System.out.println("Name:C NEHA");

System.out.println("USN: 1BM22CS074");

        System.out.println("Enter the number of books");

        n = s.nextInt();

        Books b[] = new Books[n];

        for (int i = 0; i < n; i++) {

            System.out.println("book");

            System.out.println("Enter name of book");

            Name = s.next();

            System.out.println("Enter name of Author");

            Author = s.next();

            System.out.println("Enter price");

            price = s.nextInt();

            System.out.println("Enter numPages");

            numPages = s.nextInt();

            b[i] = new Books(Name, Author, price, numPages);

        }

        for (int i = 0; i < n; i++) {

            System.out.println("Books " + (i + 1) + "\n");

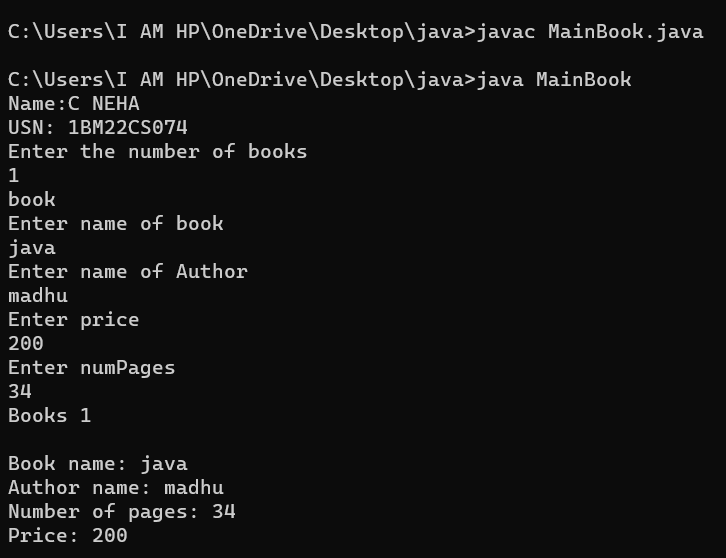
            System.out.println(b[i].toString());

        }

    }

}

OUTPUT:



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.

abstract class Shape {

protected int dimension1;

protected int dimension2;

public Shape(int dimension1, int dimension2) {

this.dimension1 = dimension1;

this.dimension2 = dimension2;

}

public abstract void printArea();

}

class Rectangle extends Shape {

public Rectangle(int length, int width) {

super(length, width);

}

public void printArea() {

int area = dimension1 \* dimension2;

System.out.println("Area of Rectangle: " + area);

}

}

class Triangle extends Shape {

public Triangle(int base, int height) {

super(base, height);

}

public void printArea() {

double area = 0.5 \* dimension1 \* dimension2;

System.out.println("Area of Triangle: "+ area);

}

}

class Circle extends Shape {

public Circle(int radius) {

super(radius, 0);

}

public void printArea() {

double area = 3.14\* dimension1 \* dimension1;

System.out.println("Area of Circle: " + area);

}

}

public class Main1 {

public static void main(String[] args) {

System.out.println("Name: C NEHA");

System.out.println("USN: 1BM22CS074");

Rectangle rectangle = new Rectangle(4, 5);

rectangle.printArea();

Triangle triangle = new Triangle(3, 6);

triangle.printArea();

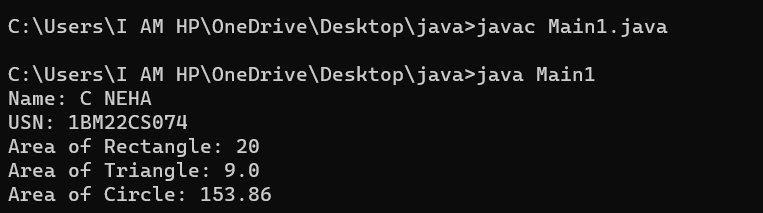
Circle circle = new Circle(7);

circle.printArea();

}

}

OUTPUT:



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 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;

abstract class Account {

protected String customerName;

protected int accountNumber;

protected double balance;

protected String accountType;

public Account(String customerName, int accountNumber, String

accountType) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

}

public abstract void deposit(double amount);

public abstract int withdraw(double amount);

public abstract void displayBalance();

}

class SavAcct extends Account {

private double interestRate;

public SavAcct(String customerName, int accountNumber, double

interestRate) {

super(customerName, accountNumber, "Savings");

this.interestRate = interestRate;

}

public void deposit(double amount) {

balance += amount;

computeInterest();

System.out.println("Deposit successful!");

}

public int withdraw(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawal successful!");

return 1;

} else {

System.out.println("Insufficient funds!");

return 0;

}

}

public void displayBalance() {

System.out.println("Account Type: " + accountType);

System.out.println("Customer: " + customerName);

System.out.println("Account Number: " + accountNumber);

System.out.println("Balance: " + balance);

}

private void computeInterest() {

double interest = balance \* (interestRate / 100);

balance += interest;

System.out.println("Interest deposited: " + interest);

}

}

class CurAcct extends Account {

private double minimumBalance;

private double serviceCharge;

public CurAcct(String customerName, int accountNumber, double

minimumBalance, double serviceCharge) {

super(customerName, accountNumber, "Current");

this.minimumBalance = minimumBalance;

this.serviceCharge = serviceCharge;

}

public void deposit(double amount) {

balance += amount;

System.out.println("Deposit successful!");

}

public int withdraw(double amount) {

if (balance - amount >= minimumBalance) {

balance -= amount;

System.out.println("Withdrawal successful!");

return 1;

} else {

System.out.println("Insufficient funds! Service charge of " +

serviceCharge + " imposed.");

balance -= serviceCharge;

return 0;

}

}

public void displayBalance() {

System.out.println("Account Type: " + accountType);

System.out.println("Customer: " + customerName);

System.out.println("Account Number: " + accountNumber);

System.out.println("Balance: " + balance);

}

}

public class lab {

public static void main(String[] args) {

System.out.println("\n\n");

System.out.println("Name: C NEHA \nUSN: 1BM22CS074\n");

Scanner scanner = new Scanner(System.in);

System.out.println("Enter customer name: ");

String customerName = scanner.nextLine();

System.out.println("Enter account number: ");

int accountNumber = scanner.nextInt();

System.out.println("Choose account type:");

System.out.println("1. Savings");

System.out.println("2. Current");

int accountTypeChoice = scanner.nextInt();

if (accountTypeChoice == 1) {

System.out.println("Enter interest rate for Savings account:");

double interestRate = scanner.nextDouble();

SavAcct savingsAccount = new SavAcct(customerName, accountNumber,

interestRate);

operateAccount(savingsAccount);

} else if (accountTypeChoice == 2) {

System.out.println("Enter minimum balance for Current account:");

double minBalance = scanner.nextDouble();

System.out.println("Enter service charge for Current account:");

double serviceCharge = scanner.nextDouble();

CurAcct currentAccount = new CurAcct(customerName, accountNumber,

minBalance, serviceCharge);

operateAccount(currentAccount);

} else {

System.out.println("Invalid account type choice!");

}

scanner.close();

System.out.println("\n\n");

}

private static void operateAccount(Account account) {

Scanner scanner = new Scanner(System.in);

int choice;

do {

System.out.println("\nMenu:");

System.out.println("1. Deposit");

System.out.println("2. Withdraw");

System.out.println("3. Display Balance");

System.out.println("4. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.print("Enter amount to deposit: ");

double depositAmount = scanner.nextDouble();

account.deposit(depositAmount);

break;

case 2:

System.out.print("Enter amount to withdraw: ");

double withdrawAmount = scanner.nextDouble();

account.withdraw(withdrawAmount);

break;

case 3:

account.displayBalance();

break;

case 4:System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice!");

}

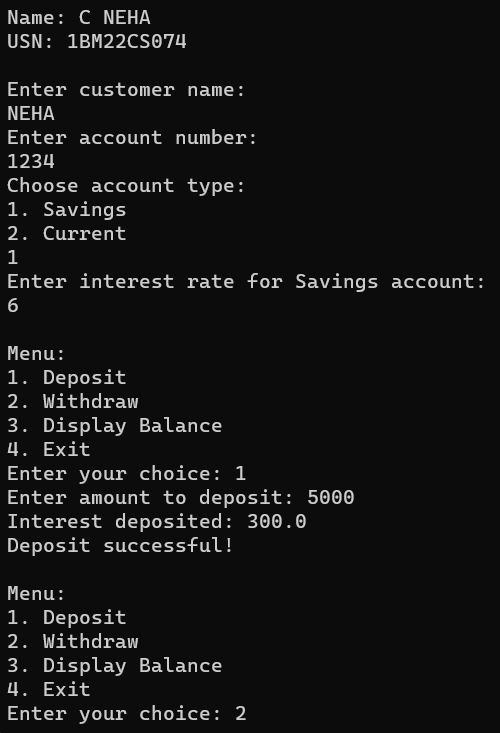
} while (choice != 4);

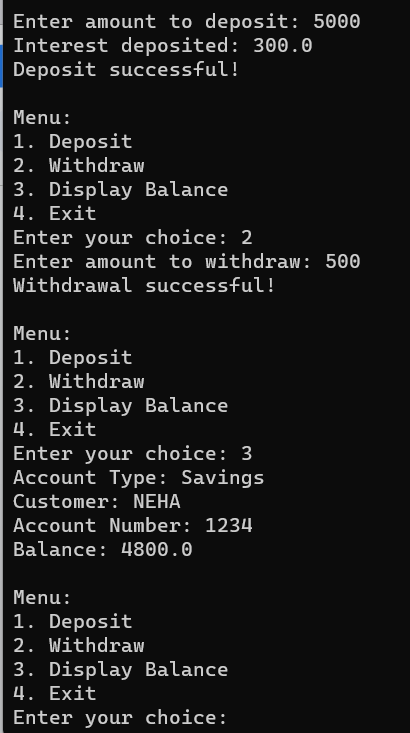
scanner.close();

}

}

OUTPUT:





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.

package CIE;

public class Student

{

String name;

String usn;

int sem;

public Student(String name,String usn,int sem)

{

this.name=name;

this.usn=usn;

this.sem=sem;

}

}

package CIE;

public class Internals extends CIE.Student

{

public int [] InternalMarks;

public Internals(String name,String usn,int sem,int []InternalMarks)

{

super(name,usn,sem);

this.InternalMarks=InternalMarks;

}

}

package SEE;

public class Externals extends CIE.Student

{

public int[] SeeMarks;

public Externals(String name,String usn,int sem,int []SeeMarks)

{

super(name,usn,sem);

this.SeeMarks=SeeMarks;

}

}

//Main File

import CIE.Internals;

import SEE.Externals;

import java.util.Scanner;

public class FinalMarks {

public static void main(String[] args) {

System.out.println("\n\n");

System.out.println("Name: Brijesh S G \nUSN: 1BM22CS073\n");

Scanner s1 = new Scanner(System.in);

System.out.println("\nEnter the number of students");

int n = s1.nextInt();

String[] name = new String[n];

String[] usn = new String[n];

int[] sem = new int[n];

int[][] InternalMarks = new int[n][5];

int[][] SeeMarks = new int[n][5];

for (int i = 0; i < n; i++) {

System.out.println("Enter details of Student" + (i + 1) + ":");

System.out.print("\nName:");

name[i] = s1.next();

s1.nextLine();

System.out.print("\nUSN:");

usn[i] = s1.next();

s1.nextLine();

System.out.print("\nSem:");

sem[i] = s1.nextInt();

System.out.println("Enter internal marks for 5 courses:");

for (int j = 0; j < 5; j++) {

System.out.println("\nCourse" + (j + 1) + ":");

InternalMarks[i][j] = s1.nextInt();

}

System.out.println("Enter See marks for 5 courses:");

for (int j = 0; j < 5; j++) {

System.out.println("\nCourse" + (j + 1) + ":");

SeeMarks[i][j] = s1.nextInt();

}

}

int[][] FinalMarks = new int[n][5];

for (int i = 0; i < n; i++) {

Internals I1 = new Internals(name[i], usn[i], sem[i],

InternalMarks[i]);

Externals E1 = new Externals(name[i], usn[i], sem[i],

SeeMarks[i]);

for (int j = 0; j < 5; j++) {

FinalMarks[i][j] = I1.InternalMarks[i] + E1.SeeMarks[j];

} System.out.println("\n\nFinal Marks for "

+

n

+

"

Student in 5

courses:");

for (int k = 0; k < n; k++) {

System.out.println(name[k] + ":");

System.out.println("CIE marks:" + ": ");

for (int j = 0; j < 5; j++) {

System.out.println("Course"+(j+1)+":

"+InternalMarks[k][j]);

} System.out.println("SEE marks:" +

": ");

for (int j = 0; j < 5; j++) {

System.out.println("Course"+(j+1)+": "+SeeMarks[k][j]);

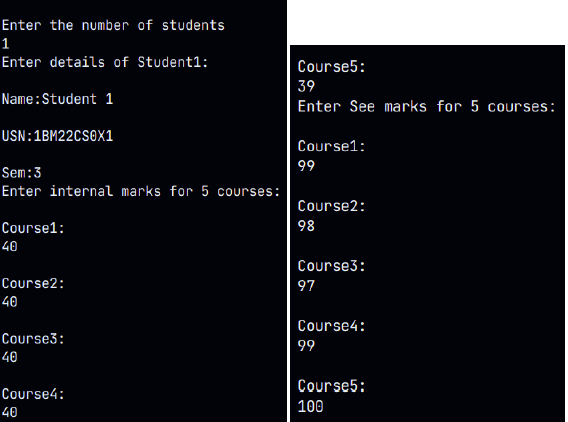
} System.out.println("\n\n");

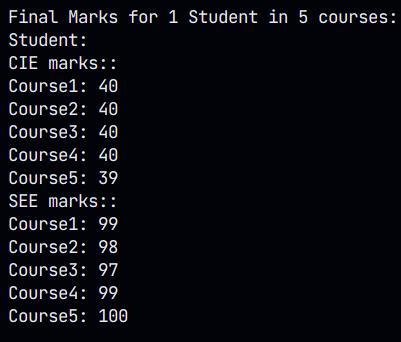
} s1.close();

}

}}

OUTPUT:





**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.

import java.util.Scanner;

class WrongAgeException extends Exception {

public WrongAgeException(String message)

super(message);

}

}

class Father { private int age;

public Father(int age) throws WrongAgeException {

if (age < 0) {

throw new WrongAgeException("Age cannot be negative");

}

this.age = age;

}

public int getAge() {

return age;

}

}

class Son extends Father {

private int sonAge;

public Son(int fatherAge, int sonAge) throws WrongAgeException {

super(fatherAge);

if (sonAge >= fatherAge) {

throw new WrongAgeException("Son's age should be less than Father's age");

}

this.sonAge = sonAge;

}

public int getSonAge() {

return sonAge;

}

}

class ExceptionHandlingDemo {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

System.out.println(" NAME: C Neha");

System.out.println("USN:1BM22CS074");

System.out.print("Enter Father's Age: ");

int fatherAge = scanner.nextInt();

Father father = new Father(fatherAge);

System.out.println("Father's Age: " + father.getAge());

System.out.print("Enter Son's Age: ");

int sonAge = scanner.nextInt();

Son son = new Son(father.getAge(), sonAge);

System.out.println("Son's Age: " + son.getSonAge());

} catch (WrongAgeException e) {

System.out.println("Exception caught: " + e.getMessage());

} finally {

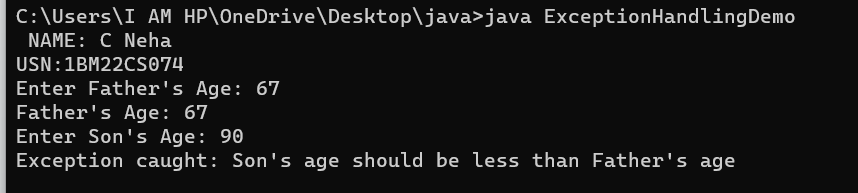
scanner.close();

}

}

}

OUTPUT:



**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.

public class Thread1 extends Thread{

public void run(){

while(true)

{

try{

System.out.println("B.M.S college of Engineering");

Thread1.sleep(10000);

}

catch(InterruptedException e)

{

System.out.println("Thread 1 exception");

}

}

}

}

public class Thread2 extends Thread{

public void run(){

while(true){

try{

System.out.println("CSE");

Thread1.sleep(2000);

}

catch(InterruptedException e)

{

System.out.println("Thread 2 exception");

}

}

}

}

public class Threads{

public static void main(String args[])

{

System.out.println(" NAME: C Neha");

System.out.println("USN:1BM22CS074");

Thread1 t1=new Thread1();

Thread2 t2=new Thread2();

t1.start();

t2.start();

}

}

OUTPUT:

