**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“Jnana Sangama”, Belgaum -590014, Karnataka.**



**PROJECT WORK REPORT**

**On**

# “OOJ LAB REPORT”

***Submitted by***

# C MOHAN BABU(2023BMS02523)

## Under the Guidance of

**SONIKA SHARMA**

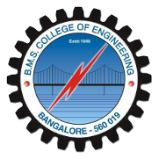
**ASSISTANT PROFESSOR**

## 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**

**Mar 2024**

**B. M. S. College of Engineering,**

**Bull Temple Road, Bangalore 560019**

(Affiliated To Visvesvaraya Technological University, Belgaum) **Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the project work entitled **“OOJ LAB REPORT”** carried out by  **C MOHAN BABU ( 2023BMS02523 )** who are bonafide students of **B. M.**

**S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visveswaraiah Technological University, Belgaum during the year 2024. The project report has been approved as it satisfies the academic requirements in respect of **JAVA LAB REPORT** work prescribed for the said degree.

Signature of the Guide Signature of the HOD

Swathi Sridharan Dr. Jyothy S Nayak

Assistant Professor Prof & Head of Dept of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

External Viva

Name of the Examiner Signature with date

**B.M.S. COLLEGE OF ENGINEERING**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



## DECLARATION

I, **C MOHAN BABU(2023BMS02523)** of 3rd Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this PROJECT entitled **"OOJ LAB REPORT"** has been carried out by me under the guidance of Sonika Sharma ma’am, Assistant Professor, Department of CSE, BMS College of Engineering, Bangalore during the academic semester Dec 2023 - Mar 2024.

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

Signature

# C MOHAN BABU(2023BMS02523)

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **TITLE** | **PAGE NO.** |
| 1 | **Week 1-**  **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 discriminant b2-4ac is negative, display a message stating that there are no real solutions. | 1-3 |
| 2 | **Week 2-**  **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. | 4-6 |
| 3 | **Week 3-**  **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 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 contains only the method printArea( ) that prints the area of the given shape. | 7-11 |
| 4 | **Week 4-**  **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: | 12-18 |
|  | 1. Accept deposit from customer and update the balance. 2. Display the balance. 3. Compute and deposit interest 4. Permit withdrawal and update the balance   Check for the minimum balance, impose penalty if necessary and update the balance. |  |
| 5 | Week 5 –  **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. | 19-21 |
| 6 | Week 6 –  **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.  **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. | 22-2 |
| 7 | Week 7 –  **Program 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. | 27-30 |

# WEEK 1

**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 discriminant b2-4ac is negative, display a message stating that there are no real solutions.

**CODE:**

import java.util.Scanner;

class Demo {

public static void main(String args[]) {

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

System.out.println("USN:2023BMS02523");

Scanner sc = new Scanner(System.in);

double a = sc.nextDouble();

double b = sc.nextDouble();

double c = sc.nextDouble();

double d = b \* b - 4 \* a \* c;

if (d < 0) {

System.out.println("No real solutions exist");

} else if (d == 0) {

double root = -b / (2 \* a);

System.out.println("Root is real and equal");

System.out.println("Root: " + root);

} else {

double root1 = (-b + Math.sqrt(d)) / (2 \* a);

double root2 = (-b - Math.sqrt(d)) / (2 \* a);

System.out.println("Roots are real and distinct");

System.out.println("Root 1: " + root1);

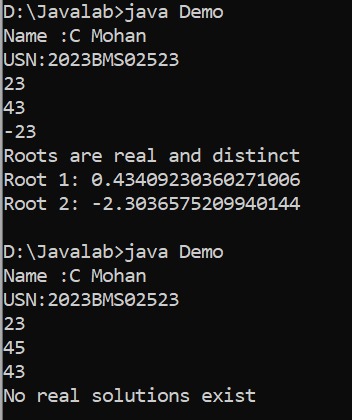
System.out.println("Root 2: " + root2);

}

}

}

**OUTPUT:**

****

**WEEK 2**

**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 Student {

private String usn;

private String name;

private int[] credits;

private int[] marks;

public Student() {

credits = new int[5];

marks = new int[5];

}

public void acceptDetails() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter USN: ");

usn = scanner.next();

System.out.print("Enter Name: ");

name = scanner.next();

System.out.println("Enter credits and marks for 5 subjects:");

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

System.out.print("Subject " + (i + 1) + " Credits: ");

credits[i] = scanner.nextInt();

System.out.print("Subject " + (i + 1) + " Marks: ");

marks[i] = scanner.nextInt();

}

}

public void displayDetails() {

System.out.println("\nStudent Details:");

System.out.println("USN: " + usn);

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

System.out.println("Credits and Marks:");

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

System.out.println("Subject " + (i + 1) + ": Credits - " + credits[i] + ", Marks - " + marks[i]);

}

}

public double calculateSGPA() {

double totalCredits = 0;

double weightedSum = 0;

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

totalCredits += credits[i];

weightedSum += calculateGradePoints(marks[i]) \* credits[i];

}

return weightedSum / totalCredits;

}

private int calculateGradePoints(int marks) {

if (marks >= 90) {

return 10;

} else if (marks >= 80) {

return 9;

} else if (marks >= 70) {

return 8;

} else if (marks >= 60) {

return 7;

} else if (marks >= 50) {

return 6;

} else if (marks >= 40) {

return 5;

} else {

return 0;

}

}

}

public class Main {

public static void main(String[] args) {

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

System.out.println("USN:2023BMS02523");

Student student = new Student();

student.acceptDetails();

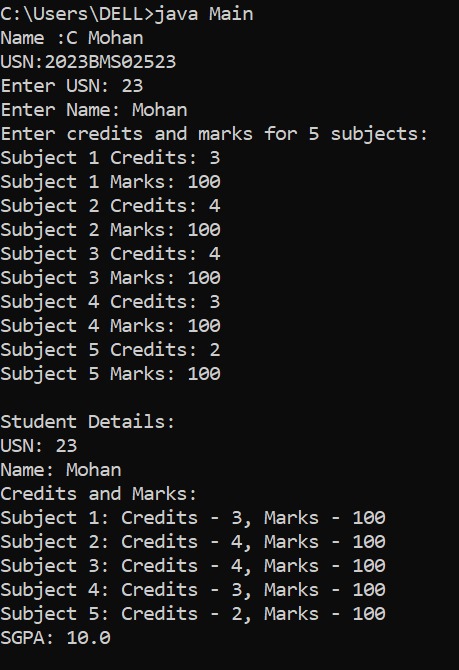
student.displayDetails();

System.out.println("SGPA: " + student.calculateSGPA());

}

}

**OUTPUT:**



# WEEK 3

**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.\*;

class Book{

String name;

String author;

int price;

int num\_pages;

public Book(String name, String author, int 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 String getName(){

return name;

}

public void setAuthor(String author){

this.author=author;

}

public String getAuthor(){

return author;

}

public void setPrice(int price){

this.price=price;

}

public int getPrice(){

return price;

}

public void setNum\_Page(int num\_pages){

this.num\_pages=num\_pages;

}

public int getNum\_Page(){

return num\_pages;

}

public String toString(){

return "Name of the book is "+name+"\n the author of the book is "+author+"\n the price of the book is "+price+"\n the total number of the pages of the book is"+num\_pages;

}

}

class Demo1{

public static void main(String args[]){

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

System.out.println("USN:2023BMS02523");

Scanner input=new Scanner(System.in);

int n=input.nextInt();

Book[] b=new Book[n];

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

System.out.println("Enter details for book "+(i+1)+":");

input.nextLine();

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

String name=input.nextLine();

System.out.println("Author :");

String author=input.nextLine();

System.out.println("price is:");

int price=input.nextInt();

System.out.println("number of pages in book is:");

int num\_page=input.nextInt();

b[i]=new Book(name,author,price,num\_page);

}

System.out.println("the book details are:");

for(Book book:b){

System.out.println(book.toString());

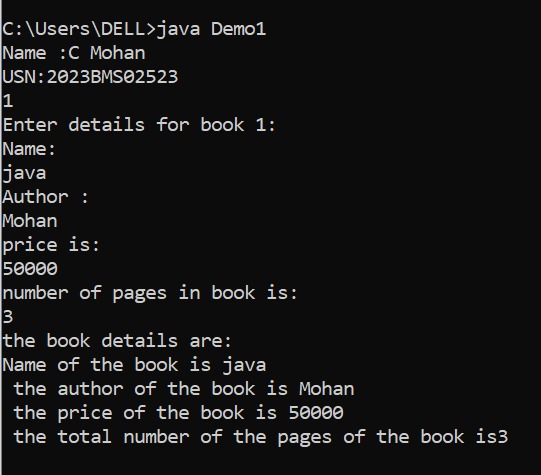
System.out.println();

}

}

}

**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 contains only the method printArea( ) that prints the area of the given shape.

**CODE:**

abstract class Shape {

protected int width;

protected int height;

public Shape(int width, int height) {

this.width = width;

this.height = height;

}

public abstract void printArea();

}

class Rectangle extends Shape {

public Rectangle(int width, int height) {

super(width, height);

}

@Override

public void printArea() {

System.out.println("Rectangle Area: " + (width \* height));

}

}

class Triangle extends Shape {

public Triangle(int width, int height) {

super(width, height);

}

@Override

public void printArea() {

System.out.println("Triangle Area: " + (0.5 \* width \* height));

}

}

class Circle extends Shape {

public Circle(int width, int height) {

super(width, height);

}

@Override

public void printArea() {

System.out.println("Circle Area: " + (Math.PI \* width \* width));

}

}

public class Main {

public static void main(String[] args) {

Shape rectangle = new Rectangle(5, 10);

Shape triangle = new Triangle(5, 10);

Shape circle = new Circle(5, 0);

rectangle.printArea();

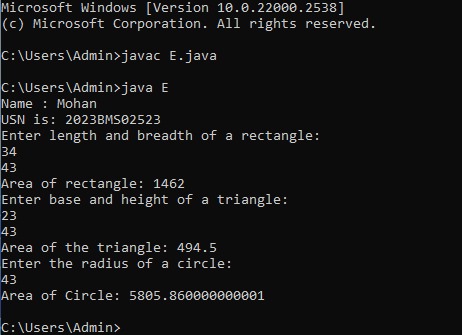
triangle.printArea();

circle.printArea();

}

}

**OUTPUT:**

****

# WEEK 4

**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:

1. Accept deposit from customer and update the balance.
2. Display the balance.
3. Compute and deposit interest
4. Permit withdrawal and update the balance

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

**CODE:**

import java.util.Scanner;

class Account {

String customerName;

int accountNumber;

String accountType;

double balance;

public Account(String customerName, int accountNumber, String accountType, double balance) {

this.customerName = customerName;

this.accountNumber = accountNumber;

this.accountType = accountType;

this.balance = balance;

}

public void displayBalance() {

System.out.println("Balance for Account Number " + accountNumber + ": $" + balance);

}

public void deposit(double amount) {

balance += amount;

System.out.println("Deposit of $" + amount + " successful.");

}

public void withdrawal(double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println("Withdrawal of $" + amount + " successful.");

} else {

System.out.println("Insufficient funds for withdrawal.");

}

}

}

class CurrAcct extends Account {

double minBalance;

double serviceCharge;

public CurrAcct(String customerName, int accountNumber, double balance) {

super(customerName, accountNumber, "Current", balance);

this.minBalance = 1000;

this.serviceCharge = 50;

}

@Override

public void withdrawal(double amount) {

if (balance - amount >= minBalance) {

super.withdrawal(amount);

} else {

System.out.println("Insufficient funds to maintain minimum balance. Service charge of $" + serviceCharge + " imposed.");

balance -= serviceCharge;

}

}

}

class SavAcct extends Account {

double interestRate;

public SavAcct(String customerName, int accountNumber, double balance) {

super(customerName, accountNumber, "Savings", balance);

this.interestRate = 0.05; // 5% interest rate for savings account

}

public void computeInterest() {

double interest = balance \* interestRate;

deposit(interest);

System.out.println("Interest of $" + interest + " computed and deposited.");

}

}

public class Bank {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

System.out.println("USN:2023BMS02523");

CurrAcct currentAccount = new CurrAcct("John Doe", 12345, 1500);

currentAccount.displayBalance();

currentAccount.withdrawal(2000);

currentAccount.displayBalance();

SavAcct savingsAccount = new SavAcct("Jane Doe", 67890, 5000);

savingsAccount.displayBalance();

savingsAccount.computeInterest();

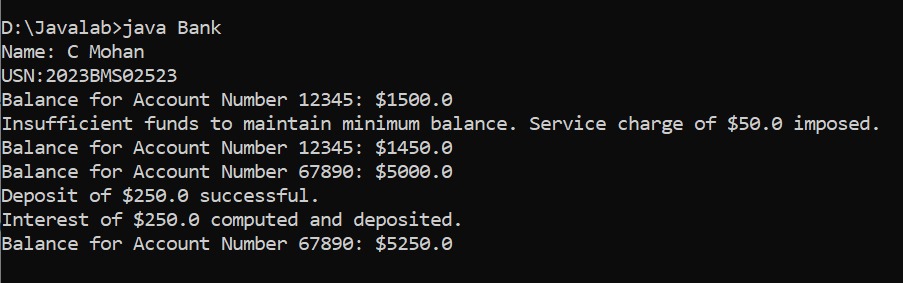
savingsAccount.displayBalance();

scanner.close();

}

}

**OUTPUT:**



# WEEK 6

**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.

**CODE:**

package CIE;

import java.util.\*;

public class Student

{

public int sem;

public String usn;

public String name;

public void accept()

{

Scanner scan = new Scanner(System.in);

System.out.println("Enter USN, Name, Sem:\n");

usn=scan.nextLine();

name=scan.nextLine();

sem=scan.nextInt();

}

}

package CIE;

public class Internals

{

public int internal[]=new int[5];

}

package SEE;

import CIE.Student;

public class External extends Student

{

public int external[]=new int[5];

}

import java.util.\*;

import SEE.\*;

import CIE.\*;

public class FinalMarks

{

public static void main(String args[])

{

System.out.println("Naem:Mohan");

System.out.println("USN:2023BMS02523");

int fm[]=new int[5];

Scanner sc= new Scanner(System.in);

System.out.println("Enter n: ");

int n=sc.nextInt();

SEE.External st[]=new SEE.External[n];

CIE.Internals s[]=new CIE.Internals[n];

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

{

st[i]=new SEE.External();

s[i]=new CIE.Internals();

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

st[i].accept();

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

{

System.out.println("Enter internal and external of sub "+(j+1));

s[i].internal[j]=sc.nextInt();

st[i].external[j]=sc.nextInt();

fm[j]=s[i].internal[j]+st[i].external[j];

}

System.out.println("Final marks of "+st[i].name);

for(int k=0; k<5; k++)

{

System.out.println("Course "+(k+1)+" = "+fm[k]);

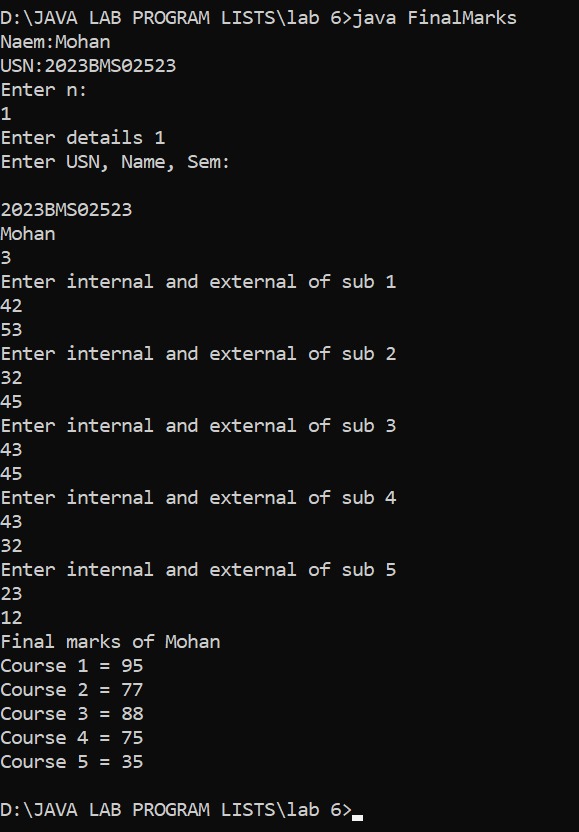
}

}

}

}

**Output:**



**Week 6:**

**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:**

class WrongAge extends Exception{

public WrongAge(String message){

super(message);

}

}

class Father{

int age;

public Father(int age) throws WrongAge{

if(age<0){

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

}

this.age=age;

}

}

class Son extends Father{

int sonAge;

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

super(fatherAge);

if(sonAge>=fatherAge){

throw new WrongAge("Son's age should be less than father's age");

}

this.sonAge=sonAge;

}

}

class Main2{

public static void main(String args[]) {

Father father=null;

try{

father=new Father(-3);

}

catch(WrongAge e){

System.out.println("Father age is wrong "+e.getMessage());

}

try{

father=new Father(40);

Son son=new Son(father.age,45);

}

catch(WrongAge e){

System.out.println("given age is wrong "+e.getMessage());

}

try{

father=new Father(40);

Son son=new Son(father.age,5);

}

catch(WrongAge e){

System.out.println("Age is coorect");

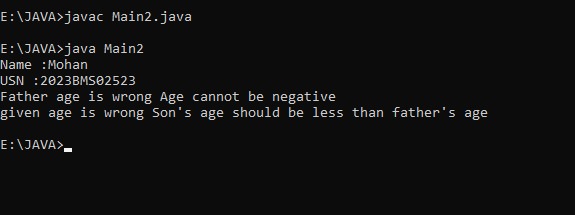
}

System.out.println("Excpetion not caught"+father.age);

}

}

**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.

class A extends Thread{

public void run(){

for(int i=1;i<=10;i++){

System.out.println("Hi");

try{

Thread.sleep(20);

}

catch(InterruptedException e){

e.printStackTrace();

}

}

}

}

class B extends Thread{

public void run(){

for(int i=1; i<=10; i++){

System.out.println("Mohan");

try{

Thread.sleep(20);

}

catch(InterruptedException e){

e.printStackTrace();

}

}

}

}

class Z extends Thread{

public void run(){

for(int i=1; i<=10; i++){

System.out.println("Mohan");

try{

Thread.sleep(20);

}

catch(InterruptedException e){

e.printStackTrace();

}

}

}

public void printName(){

System.out.println("Thread Name is"+Thread.currentThread().getName());

Thread.currentThread().setPriority(Thread.MIN\_PRIORITY);

System.out.println("A"+getPriority());

}

}

class ThreadStart{

public static void main(String args[]){

A a=new A();

B b=new B();

Z z=new Z();

z.start();

b.setPriority(Thread.MIN\_PRIORITY);

System.out.println("Priroity is"+b.getPriority());

a.start();

try{

Thread.sleep(12);

}

catch(InterruptedException e){

e.printStackTrace();

}

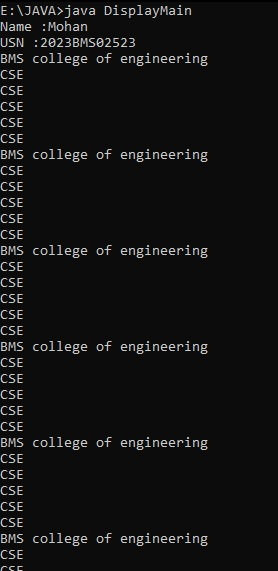
b.start();

z.printName();

}

}

**Output:**



**Week 7 –**

**Program 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.

**Code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class SwingDemo{

SwingDemo(){

JFrame jfrm=new JFrame("Divider");

jfrm.setSize(270,180);

jfrm.setLayout(new FlowLayout());

jfrm.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JLabel jlab=new JLabel("Enter the divider and divident");

JTextField ajtf=new JTextField(8);

JTextField bjtf=new JTextField(20);

JButton button=new JButton("Calculate");

JLabel err=new JLabel();

JLabel alab=new JLabel();

JLabel blab=new JLabel();

JLabel anslab=new JLabel();

jfrm.add(err);

jfrm.add(jlab);

jfrm.add(ajtf);

jfrm.add(bjtf);

jfrm.add(button);

jfrm.add(alab);

jfrm.add(blab);

jfrm.add(anslab);

ActionListener l=new ActionListener(){

public void actionPerformed(ActionEvent evt){

System.out.println("Action event from a team");

}

};

ajtf.addActionListener(l);

bjtf.addActionListener(l);

button.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent evt){

try{

int a=Integer.parseInt(ajtf.getText());

int b=Integer.parseInt(bjtf.getText());

int ans=a/b;

alab.setText("\nA="+a);

blab.setText("\nB="+b);

anslab.setText("\nAns="+ans);

}

catch(NumberFormatException e){

alab.setText("");

blab.setText("");

anslab.setText("");

err.setText("Enter Only Integers!");

}

catch(ArithmeticException e){

alab.setText("");

blab.setText("");

anslab.setText("");

err.setText("B should be NON zero!");

}

}

});

jfrm.setVisible(true);

}

public static void main(String args[]){

System.out.println("Name:Mohan");

System.out.println("USN:2023BMS02523");

SwingUtilities.invokeLater(new Runnable(){

public void run(){

new SwingDemo();

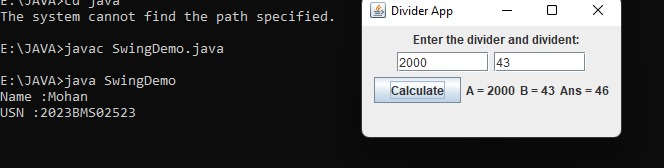
}

});

}

}

**Output:**

****