

**B.M.S COLLEGE OF ENGINEERING BENGALURU**  
Autonomous Institute, Affiliated to VTU



# **LAB REPORT**

**23CS3PCOOJ**

## **OBJECT ORIENTED JAVA PROGRAMMING**

Bachelor of Engineering  
in  
Computer Science and Engineering

*Submitted by:*

**ARUGUNTA HAMSIKA**

**1BM22CS054**

Department of Computer Science and Engineering  
B.M.S College of Engineering  
Bull Temple Road, Basavanagudi, Bangalore 560 019  
2022-2023

**B.M.S COLLEGE OF ENGINEERING**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



***DECLARATION***

I, ARUGUNTA HAMSIKA (1BM22CS054) student of 2<sup>nd</sup> Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this LAB REPORT for OBJECT ORIENTED JAVA PROGRAMMING has been carried out in Department of CSE, BMS College of Engineering, Bangalore during the academic semester december - march 2024. I also declare that to the best of our knowledge and belief, the lab report is not from part of any other report by any other students.

**Signature of the Candidate**

Arugunta Hamsika (1BM22CS054)

**BMS COLLEGE OF ENGINEERING**  
**DEPARTMENT OF COMPUTER SCIENCE AND**  
**ENGINEERING**



***CERTIFICATE***

This is to certify that the lab report for OBJECT ORIENTED JAVA PROGRAMMING has been carried out by **ARUGUNTA HAMSIKA (1BM22CS054)** during the academic year 2023-2024.

Signature of the Faculty in Charge

## Table of Contents

Sl No	Title	Pageno
1	Quadratic equation	1-2
2	SGPA calculation	3-5
3	Book objects	6-9
4	Area of given shape	10-11
5	bank account	12-20
6	packages	21-26
7	Father&son age exception	27-29
8	Threads	30-32
9	divider app	33-36

## WEEK 1

Program 1: Develop a Java program that prints all real solutions to the quadratic equation  $ax^2+bx+c=0$ . Read in a, b, c and use the quadratic formula. If the discriminant  $b^2-4ac$  is negative, display a message stating that there are no real solutions

### Code:

```
import java.util.Scanner;

public class QuadraticEquation {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the coefficients of the quadratic equation (a, b, c):");
        double a = scanner.nextDouble();
        double b = scanner.nextDouble();
        double c = scanner.nextDouble();

        double discriminant = b * b - 4 * a * c;

        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("Two distinct real roots exist:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("One real root exists:");
            System.out.println("Root: " + root);
        } else {
            System.out.println("No real solutions exist for the given quadratic equation.");
        }

        System.out.println("Name: Arugunta Hamsika");
        System.out.println("ID: 1BM22CS054");
    }
}
```

}

Output:

```
Enter the coefficients of the quadratic equation (a, b, c):  
1  
6  
1  
Two distinct real roots exist:  
Root 1: -0.1715728752538097  
Root 2: -5.82842712474619  
Name: Arugunta Hamsika  
ID: 1BM22CS054
```

```
Enter the coefficients of the quadratic equation (a, b, c):  
21  
18  
15  
No real solutions exist for the given quadratic equation.  
Name: Arugunta Hamsika  
ID: 1BM22CS054
```

```
Enter the coefficients of the quadratic equation (a, b, c):  
1  
2  
1  
One real root exists:  
Root: -1.0  
Name: Arugunta Hamsika  
ID: 1BM22CS054
```

## 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 {
```

```
    String USN;
```

```
    String name;
```

```
    int marks[] = new int[6];
```

```
    float percentage = 0;
```

```
    void getd(int i) {
```

```
        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 Student" + i + " Marks");
```

```
        for (int j = 0; j < 6; j++) {
```

```
            System.out.println("Enter Marks of Subject" + j + ":");
```

```
            marks[j] = s.nextInt();
```

```
            percentage += marks[j];
```

```
        }
```

```
    }
```

```

void calculatePercentage(int i) {
    percentage = (percentage / 6);
    System.out.println("Percentage of student" + i + "=" + percentage + "%");
}
}

class studentMain {
    public static void main(String[] args) {
        System.out.println("ARUGUNTA HAMSIKA");
        System.out.println("1BM22CS054");
        System.out.println("Enter the number of Students");
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        student s[] = new student[n];
        for (int i = 0; i < n; i++) {
            s[i] = new student();
            s[i].getd(i);
        }
        for (int i = 0; i < n; i++) {
            s[i].calculatePercentage(i);
        }
    }
}

```



Output:

```
Enter name: shreya
Enter USN: 4567

Enter no. of subjects: 3
Enter marks and credits:
Marks for subject 1:
90
Credits for subject 1:
4
Marks for subject 2:
89
Credits for subject 2:
3
Marks for subject 3:
98
Credits for subject 3:
1
Name : shreya
USN : 4567
Subject 1 :      Marks= 90      Credits= 4
Subject 2 :      Marks= 89      Credits= 3
Subject 3 :      Marks= 98      Credits= 1

SGPA : 9.625
Name: Arugunta Hamsika C
USN: 1BM22CS054
```

## 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 books {
```

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

```
    String name, author;
```

```
    int price, num_pages;
```

```
    books() {
```

```
    }
```

```
    books(String name, String author, int price, int num_pages) {
```

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

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

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

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

```
    }
```

```
    void getdata() {
```

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

```
        name = sc.nextLine();
```

```

        System.out.println("Enter the name of the author");

        author = sc.nextLine();

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

        price = sc.nextInt();

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

        num_pages = sc.nextInt();

    }

    public String toString() {

        return ("Name : " + name + "\nAuthor : " + author + "\nPrice : " + price + "\nNumber of
pages : " + num_pages);

    }

}

class bookdetails {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        books b1 = new books("Mehta", "neil", 56, 154);

        System.out.println("Constructor values : \n" + b1);

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

        int n = sc.nextInt();

        books s[] = new books[n];

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

            s[i] = new books();

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

            s[i].getdata();

```

```
    }  
    for (int i = 0; i < n; i++) {  
        System.out.println("\nDetails of the book" + (i + 1));  
        System.out.println(s[i]);  
    }  
    System.out.println("Arugunta hamsika");  
    System.out.println("1BM22CS054");  
}  
}
```

Output:

```
Constructor values :
Name : Mehta
Author : neil
Price : 56
Number of pages : 154
Enter the number of object of books
1
Enter the details of 1book
Enter the name of the book
it starts with us
Enter the name of the author
collen hoover
Enter the price
458
Enter the number of pages
300

Details of the book1
Name : it starts with us
Author : collen hoover
Price : 458
Number of pages : 300
Arugunta Hamsika
1BM22CS054
```

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:

```
import java.util.Scanner;

abstract class Shape {
    int x, y;

    abstract void area();

    public static void main(String args[]) {
        Shape obj1 = new Circle();
        obj1.area();
        Shape obj2 = new Rectangle();
        obj2.area();
        Shape obj3 = new Triangle();
        obj3.area();
        System.out.println("Arugunta Hamsika");
        System.out.println("1BM22CS054");
    }
}

class Circle extends Shape {
    Circle() {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the radius of the circle");
        x = sc.nextInt();
        y = x;
    }

    void area() {
        System.out.println("area of circle is " + 3.14 * x * y);
    }
}

class Rectangle extends Shape {
    Rectangle() {
```

```

        Scanner sc = new Scanner(System.in);
        System.out.println("enter the length and breadth of the rectangle");
        x = sc.nextInt();
        y = sc.nextInt();
    }

    void area() {
        System.out.println("area of rectangle is " + x * y);
    }
}

class Triangle extends Shape {
    Triangle() {
        Scanner sc = new Scanner(System.in);
        System.out.println("enter the base and hieght of the triangle");
        x = sc.nextInt();
        y = sc.nextInt();
    }

    void area() {
        System.out.println("area of triangle is " + 0.5 * x * y);
    }
}

```

Output:

```

enter the radius of the circle
1
area of circle is 3.14
enter the length and breadth of the rectangle
1
2
area of rectangle is 2
enter the base and hieght of the triangle
2
3
area of triangle is 3.0
Arugunta Hamsika
1BM22CS054

```

## 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:

- 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.

### Code:

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

```
class Account {
```

```
    String customerName;
```

```
    int accountNumber;
```

```
    String accountType;
```

```
    double balance;
```

```
    Account(String name, int number, String type, double initialBalance) {
```

```
        customerName = name;
```

```
        accountNumber = number;
```

```
        accountType = type;
```



```

    balance = initialBalance;
}

void deposit(double amount) {
    if (amount > 0) {
        balance += amount;

        System.out.println("Deposit of INR " + amount + " successful");
    } else {
        System.out.println("Invalid deposit amount. Please enter a positive value.");
    }
}

void displayBalance() {
    System.out.println("Account Number: " + accountNumber);
    System.out.println("Customer Name: " + customerName);
    System.out.println("Account Type: " + accountType);
    System.out.println("Balance: INR " + balance);
}

void withdraw(double amount) {
    if (balance >= amount) {
        balance -= amount;

        System.out.println("Withdrawal of INR " + amount + " successful");
    } else {
        System.out.println("Insufficient funds");
    }
}

```

```

void computeInterest() {
}

void checkMinimumBalance(double minBalance, double serviceCharge) {
}
}

class SavAcct extends Account {
    double interestRate = 0.05;

    SavAcct(String name, int number, String type, double initialBalance) {
        super(name, number, type, initialBalance);
    }

    void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of INR " + interest + " added to the account");
    }
}

class CurAcct extends Account {
    double minBalance = 1000;
    double serviceCharge = 50;

    CurAcct(String name, int number, String type, double initialBalance) {
        super(name, number, type, initialBalance);
    }
}

```

```

void checkMinimumBalance(double minBalance, double serviceCharge) {
    if (balance < minBalance) {
        System.out.println("Service charge of INR " + serviceCharge + " imposed");
        balance -= serviceCharge;
    }
}

}

}

public class Bank {
    public static void main(String[] args) {
        System.out.println("ARUGUNTA HAMSIKA");
        System.out.println("1BM22CS054");
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of users: ");
        int numUsers = scanner.nextInt();

        Account[] accounts = new Account[numUsers];
        for (int i = 0; i < numUsers; i++) {
            System.out.println("\nUser " + (i + 1));
            System.out.print("Enter customer name: ");
            scanner.nextLine();
            String name = scanner.nextLine();
            System.out.print("Enter account number: ");
            int accNumber = scanner.nextInt();
            System.out.print("Enter initial deposit amount: INR ");
            double initialDeposit = scanner.nextDouble();

```

```

System.out.print("Enter account type (Savings/Current): ");
scanner.nextLine();

String accType = scanner.nextLine();

if (accType.equalsIgnoreCase("Savings")) {
    accounts[i] = new SavAcct(name, accNumber, accType, initialDeposit);
} else if (accType.equalsIgnoreCase("Current")) {
    accounts[i] = new CurAcct(name, accNumber, accType, initialDeposit);
} else {
    System.out.println("Invalid account type entered. Defaulting to Account.");
    accounts[i] = new Account(name, accNumber, "Account", initialDeposit);
}
}

boolean exit = false;

while (!exit) {
    System.out.println("\nChoose an option:");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    System.out.println("4. Compute Interest (Savings only)");
    System.out.println("5. Exit");
    System.out.print("Enter your choice: ");
    while (!scanner.hasNextInt()) {
        System.out.println("Invalid input. Please enter a number.");
        scanner.next();
    }
}

```

```
int choice = scanner.nextInt();
switch (choice) {
    case 1:
        System.out.print("Enter account number: ");
        int accNum = scanner.nextInt();
        System.out.print("Enter deposit amount: INR ");
        double depositAmount = scanner.nextDouble();
        for (Account acc : accounts) {
            if (acc.accountNumber == accNum) {
                acc.deposit(depositAmount);
            }
        }
        break;
    case 2:
        System.out.print("Enter account number: ");
        accNum = scanner.nextInt();
        System.out.print("Enter withdrawal amount: INR ");
        double withdrawAmount = scanner.nextDouble();
        for (Account acc : accounts) {
            if (acc.accountNumber == accNum) {
                acc.withdraw(withdrawAmount);
            }
        }
        break;
    case 3:
```

```

        System.out.print("Enter account number: ");
        accNum = scanner.nextInt();
        for (Account acc : accounts) {
            if (acc.accountNumber == accNum) {
                acc.displayBalance();
            }
        }
        break;
case 4:
    System.out.print("Enter account number (for Savings account): ");
    accNum = scanner.nextInt();
    for (Account acc : accounts) {
        if (acc.accountNumber == accNum && acc instanceof SavAcct) {
            ((SavAcct) acc).computeInterest();
        }
    }
    break;
case 5:
    exit = true;
    break;
default:
    System.out.println("Invalid choice. Please enter a valid option.");
}
}

```

### Output:

```
ARUGUNTA HAMSIKA
1BM22CS054
Enter the number of users: 1

User 1
Enter customer name: athmica
Enter account number: 123
Enter initial deposit amount: INR 5000
Enter account type (Savings/Current): savings

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 1
Enter account number: 123
Enter deposit amount: INR 200
Deposit of INR 200.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 2
Enter account number: 123
Enter withdrawal amount: INR 100
Withdrawal of INR 100.0 successful

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
```

```
Enter your choice: 3
Enter account number: 123
Account Number: 123
Customer Name: athmica
Account Type: savings
Balance: INR 5100.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 4
Enter account number (for Savings account): 123
Interest of INR 255.0 added to the account

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings only)
5. Exit
Enter your choice: 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.

### Code:

```
//CIE PACKAGE
```

```
package CIE;
```

```
public class student {
```

```
    public String usn;
```

```
    public String name;
```

```
    public int sem;
```

```
    public student(String usn, String name, int sem) {
```

```
        this.usn = usn;
```

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

```
        this.sem = sem;
```

```
    }
```

```
}
```

```
//internals.java
```

```
package CIE;
```

```
public class internals extends student {
```

```
    public int[] internalMarks;
```

```
public internals(String usn, String name, int sem, int[] internalMarks) {  
    super(usn, name, sem);  
    this.internalMarks = internalMarks;  
}  
}
```

//SEE Package

```
package SEE;
```

```
import CIE.student;
```

```
public class externals extends student {  
    public int[] seeMarks;  
  
    public externals(String usn, String name, int sem, int[] seeMarks) {  
        super(usn, name, sem);  
        this.seeMarks = seeMarks;  
    }  
}
```

//main1.java

```
import CIE.internals;  
import SEE.externals;  
import java.util.Scanner;  
  
public class main1 {  
    public static void main(String[] args) {
```

```

System.out.println("ARUGUNTA HAMSIKA");
System.out.println("1BM22CS054");
Scanner scanner = new Scanner(System.in);
System.out.print("Enter the number of students: ");
int n = scanner.nextInt();
internals[] cieStudents = new internals[n];
externals[] seeStudents = new externals[n];

// Input CIE marks
for (int i = 0; i < n; i++) {
    System.out.println("Enter details for CIE of student " + (i + 1));
    System.out.print("USN: ");
    String usn = scanner.next();
    System.out.print("Name: ");
    String name = scanner.next();
    System.out.print("Semester: ");
    int sem = scanner.nextInt();
    int[] cieMarks = new int[5];
    System.out.print("Enter CIE marks for 5 courses: ");
    for (int j = 0; j < 5; j++) {
        cieMarks[j] = scanner.nextInt();
    }
    cieStudents[i] = new internals(usn, name, sem, cieMarks);
}

// Input SEE marks
for (int i = 0; i < n; i++) {

```

```

        System.out.println("Enter details for SEE of student " + (i + 1));

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

        String usn = scanner.next();

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

        String name = scanner.next();

        System.out.print("Semester: ");

        int sem = scanner.nextInt();

        int[] seeMarks = new int[5];

        System.out.print("Enter SEE marks for 5 courses: ");

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

            seeMarks[j] = scanner.nextInt();

        }

        seeStudents[i] = new externals(usn, name, sem, seeMarks);

    }

    // Displaying final marks

    System.out.println("\nFinal Marks of Students:");

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

        System.out.println("\nDetails of Student " + (i + 1));

        System.out.println("USN: " + cieStudents[i].usn);

        System.out.println("Name: " + cieStudents[i].name);

        System.out.println("Semester: " + cieStudents[i].sem);

        System.out.println("CIE Marks: ");

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

            System.out.print(cieStudents[i].internalMarks[j] + " ");

        }
    }

```

```
System.out.println("\nSEE Marks: ");  
for (int j = 0; j < 5; j++) {  
    System.out.print(seeStudents[i].seeMarks[j] + " ");  
}  
}  
}  
}
```

Output

ARUGUNTA HAMSIKA

1BM22CS054

Enter the number of students: 1

Enter details for CIE of student 1

USN: 1BM22CS080

Name: Akanya

Semester: 1

Enter CIE marks for 5 courses: 90 85 92 88 95

Enter details for SEE of student 1

USN: 1BM22CS080

Name: Akanya

Semester: 1

Enter SEE marks for 5 courses: 90 85 92 88 95

Final Marks of Students:

Details of Student 1

USN: 1BM22CS080

Name: Akanya

Semester: 1

CIE Marks:

90 85 92 88 95

SEE Marks:

90 85 92 88 95

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 takes 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 message) {  
        super(message);  
    }  
}  
  
class Father {  
    int fatherAge;  
  
    public Father(int fatherAge) throws WrongAge {  
        if (fatherAge < 0) {  
            throw new WrongAge("Age cannot be negative");  
        }  
        this.fatherAge = fatherAge;  
    }  
}  
  
class Son {  
    int sonAge;
```

```

public Son(int fatherAge, int sonAge) throws WrongAge {
    if (sonAge >= fatherAge) {
        throw new WrongAge("Son's age must be less than Father's age");
    }
    this.sonAge = sonAge;
}
}

```

```

public class fatherson {
    public static void main(String[] args) {
        System.out.println("ARUGUNTA HAMSIKA");
        System.out.println("1BM22CS054");
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter father's age and son's age: ");
        int fa = sc.nextInt();
        int sa = sc.nextInt();
        try {
            Father father = new Father(fa);
            Son son = new Son(fa, sa);
            System.out.println("Father's age: " + father.fatherAge);
            System.out.println("Son's age: " + son.sonAge);
        } catch (WrongAge e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```



}

Output:

```
ARUGUNTA HAMSIKA
1BM22CS054
Enter father's age and son's age:
18
50
Error: Son's age must be less than Father's age
```

```
ARUGUNTA HAMSIKA
1BM22CS054
Enter father's age and son's age:
50
18
Father's age: 50
Son's age: 18
```

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 DisplayThread extends Thread {
    private String message;
    private int intervalMillis;
    public DisplayThread(String message, int intervalMillis) {
        this.message = message;
        this.intervalMillis = intervalMillis;
    }
    public void run() {
        while (true) {
            try {
                System.out.println(message);
                Thread.sleep(intervalMillis);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class DisplayProgram {
    public static void main(String[] args) {
        System.out.println("ARUGUNTA HAMSIKA");
        System.out.println("1BM22CS054");
        DisplayThread thread1 = new DisplayThread("BMS College of Engineering", 10000); // 10
seconds
        DisplayThread thread2 = new DisplayThread("CSE", 2000); // 2 seconds
        thread1.start();
        thread2.start();
    }
}
```

Output:

```
ARUGUNTA HAMSIKA
1BM22CS054
BMS Ccollege of Engineering
CSE
CSE
CSE
CSE
CSE
BMS Ccollege of Engineering
BMS Ccollege of Engineering
BMS Ccollege of Engineering
BMS Ccollege of Engineering
```

Synchronized code:

```
import java.util.*;
class NewThread implements Runnable {
    String name;
    Thread t;
    NewThread(String name) {
        this.name = name;
        t = new Thread(this, name);
        t.start();
    }
    public synchronized void run() {
        for (int i = 0; i < 5; i++) {
            if (name.equals("one")) {
                System.out.println("BMS College of Engineering");
                try {
                    Thread.sleep(10000);
                } catch (InterruptedException e) {
                    System.out.println("Interrupted");
                }
            } else if (name.equals("two")) {
                System.out.println("CSE");
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    System.out.println("Interrupted");
                }
            }
        }
    }
}
```

```

    }
    }
    }
}
class Demo {
    public static void main(String args[]) {
        NewThread obj1 = new NewThread("one");
        NewThread obj2 = new NewThread("two");
        try {
            obj1.t.join();
            obj2.t.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted");
        }
    }
}

```

## 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 app");
        jfrm.setSize(265,150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JLabel jlab=new JLabel("Enter the divider and dividend: ");
        JTextField ajtf=new JTextField(8);
        JTextField bjtf=new JTextField(8);

        JButton button = new JButton("Calculate");

        JLabel err=new JLabel();
        JLabel alab=new JLabel();
        JLabel blab= new JLabel();
        JLabel anslab=new JLabel();

        //add in order
        jfrm.add(err);//to display error
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
    }
}
```

```

jfrm.add(anslab);

ActionListener I = new ActionListener()
{
    public void actionPerformed(ActionEvent evt)
    {
        System.out.println("Action event from a text field");
    }
};
ajtf.addActionListener(I);
bjtf.addActionListener(I);

button.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent evt)
    {
        if (err.getText() != null)
            err.setText("");
        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 + " Arugunta Hamsika [1BM22CS054]");
        }
        catch (NumberFormatException e)
        {
            alab.setText("");
            blab.setText("");
            ansLab.setText("Arugunta Hamsika [1BM22CS054]");

            err.setText("Enter only Integers!");
        }
        catch (ArithmeticException e)
        {
            alab.setText("");
            blab.setText("");
        }
    }
});

```

```

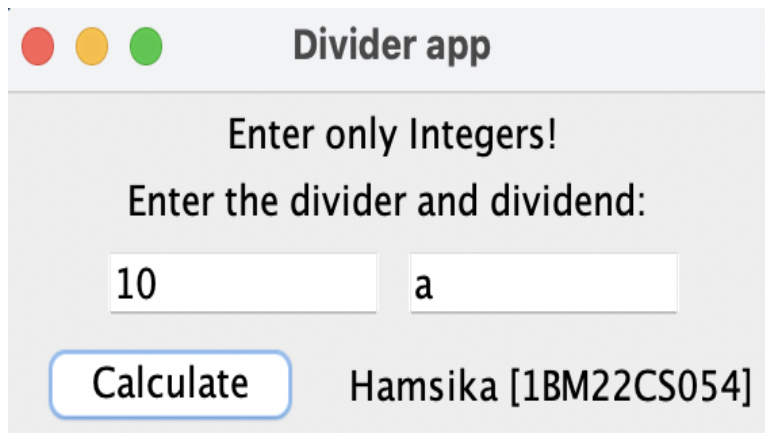
        anslab.setText("Arugunta Hamsika [1BM22CS054]");
err.setText("B should be NON zero!");
    }
}
});

//display frame
jfrm.setVisible(true);
}

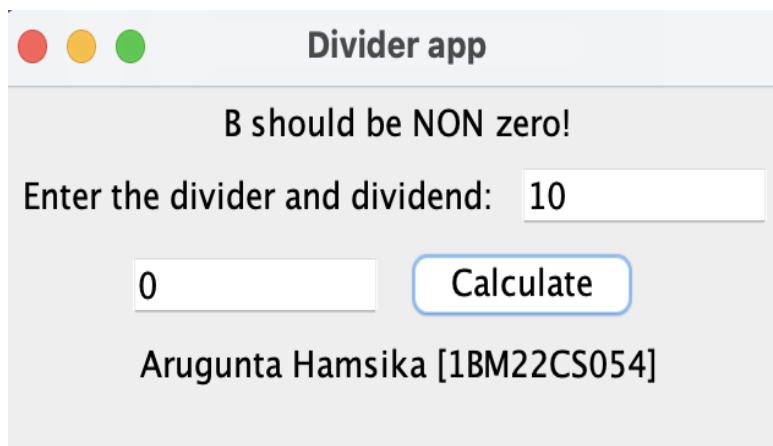
public static void main(String args[])
{
    //create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable()
    {
        public void run()
        {
            new SwingDemo();
        }
    });
}
}

```

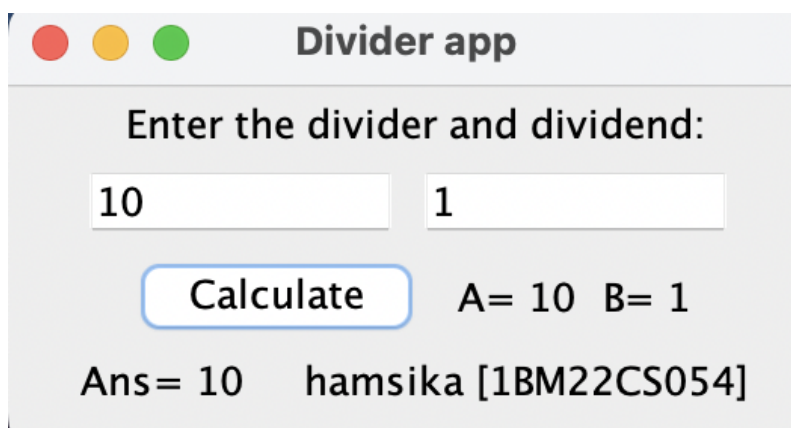
Output:



A screenshot of a window titled "Divider app". The window has a light gray background and a title bar with three colored buttons (red, yellow, green). The text inside the window reads: "Enter only Integers!" followed by "Enter the divider and dividend:". Below this, there are two input fields. The first field contains the number "10" and the second field contains the letter "a". Below the input fields, there is a blue button labeled "Calculate". To the right of the button, the text "Hamsika [1BM22CS054]" is displayed.



A screenshot of a window titled "Divider app". The window has a light gray background and a title bar with three colored buttons (red, yellow, green). The text inside the window reads: "B should be NON zero!" followed by "Enter the divider and dividend:". Below this, there are two input fields. The first field contains the number "0" and the second field contains the number "10". Below the input fields, there is a blue button labeled "Calculate". To the right of the button, the text "Arugunta Hamsika [1BM22CS054]" is displayed.



A screenshot of a window titled "Divider app". The window has a light gray background and a title bar with three colored buttons (red, yellow, green). The text inside the window reads: "Enter the divider and dividend:". Below this, there are two input fields. The first field contains the number "10" and the second field contains the number "1". Below the input fields, there is a blue button labeled "Calculate". To the right of the button, the text "A= 10 B= 1" is displayed. Below the button, the text "Ans= 10 hamsika [1BM22CS054]" is displayed.