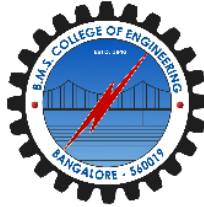


B.M.S. COLLEGE OF ENGINEERING
Basavanagudi, Bengaluru- 560019
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



LAB REPORT

On

Object Oriented Java Programming
(23CS3PCOOJ)

Submitted By:

ABHISHEK YADAV
1BM22CS009

In partial fulfilment of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
2023-24

Faculty-In-Charge

Swathi Sridharan

Assistant Professor
Department of Computer Science and Engineering

SL. NO	TOPIC	Date	PageNo
1	Quadratic Equation	22-12-2023	1
2	SGPA Calculator	29-12-2023	3
3	Book Problem	12-01-2024	7
4	Shapes	12-01-2024	9
5	Bank Problem	19-01-2024	13
6	Student External and Internal Marks	02-02-2024	17
7	Exception Handling	16-02-2024	20
8	Threads	16-02-2024	22
9	AWT	23-02-2024	24

LAB-1:QUADRATIC EQUATION

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

```
QuadEqn.java X
QuadEqn.java
1  import java.util.Scanner;
2  import java.lang.Math;
3  public class QuadEqn{
4      public static void main(String[] args){
5          float a, b, c, dsc, r1, r2;
6          Scanner reader = new Scanner(System.in);
7          System.out.println("Enter the co-efficient of x^2: ");
8          a = reader.nextFloat();
9          if (a==0){
10             System.out.println("Invalid Input");
11         }
12         else{
13             System.out.println("Enter the co-efficient of x: ");
14             b = reader.nextFloat();
15             System.out.print("Enter the value of the constant: ");
16             c = reader.nextFloat();
17             dsc = (float)Math.pow(b,2) - 4*a*c;
18             if(dsc > 0){
19                 r1 = (float)(-b + Math.sqrt(dsc))/(2*a);
20                 r2 = (float)(-b - Math.sqrt(dsc))/(2*a);
21                 System.out.println("The roots are:" + r1+ " and " + r2);
22             }
23             else if(dsc==0){
24                 r1 = (float)-b/(2*a);
25                 System.out.println("The roots are real and equal is: "+ r1);
26             }
27             else{
28                 System.out.println("No real roots exist for this equation");
29             }
30             System.out.println("Abhishek Yadav 1BM22CS009");
31         }
32     }
33 }
34
```

OUTPUT :

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java QuadEqn
Enter the co-efficient of x^2:
2
Enter the co-efficient of x:
5
Enter the value of the constant: 3
The roots are:-1.0 and -1.5
Abhishek Yadav 1BM22CS009
```

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java QuadEqn
Enter the co-efficient of x^2:
20
Enter the co-efficient of x:
1
Enter the value of the constant: 1
No real roots exist for this equation
Abhishek Yadav 1BM22CS009
```

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java QuadEqn
Enter the co-efficient of x^2:
1
Enter the co-efficient of x:
4
Enter the value of the constant: 4
The roots are real and equal is: -2.0
Abhishek Yadav 1BM22CS009
```

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java QuadEqn
Enter the co-efficient of x^2:
0
Invalid Input
```

LAB-2: STUDENT SGPA CALCULATION

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.

```
sgpa.java X
sgpa.java
1  import java.util.Scanner;
2
3  class Student {
4      String usn;
5      String name;
6      int[] credits = new int[8];
7      int[] marks = new int[8];
8      // Method to accept student details
9      public void acceptDetails() {
10         Scanner scanner = new Scanner(System.in);
11
12         System.out.print("Enter USN: ");
13         usn = scanner.nextLine();
14
15         System.out.print("Enter Name: ");
16         name = scanner.nextLine();
17
18         System.out.println("Enter details for each subject : \n");
19
20         for (int i = 0; i < credits.length; i++) {
21             System.out.print("\nEnter credits for Subject " + (i + 1) + ": ");
22             credits[i] = scanner.nextInt();
23
24             System.out.print("\nEnter marks for Subject " + (i + 1) + ": ");
25             marks[i] = scanner.nextInt();
26         }
27         scanner.close();
28     }
```

```

30      // Method to calculate SGPA
31      public double calculateSGPA() {
32          int totalCredits = 0;
33          int weightedSum = 0;
34          double ans;
35
36          for (int i = 0; i < credits.length; i++) {
37              totalCredits += credits[i];
38              int gradePoints;
39
40              gradePoints = (marks[i]/10)+1;
41
42              if(gradePoints == 11){
43                  gradePoints=10;
44              }
45
46              else if(gradePoints<=4){
47                  gradePoints=0;
48              }
49
50              weightedSum += gradePoints * credits[i];
51          }
52
53          ans = (double) weightedSum / (double) totalCredits;
54          return ans;
55      }
56  }
57

```

```
58 public class sgpa {
59     public static void main(String[] args) {
60         Scanner scanner = new Scanner(System.in);
61
62         // Create a Student object
63         Student student = new Student();
64
65         // Accept and display details
66         student.acceptDetails();
67
68         System.out.println("\nStudent Details :");
69
70         System.out.println("USN : " + student.usn);
71         System.out.println("Name : " + student.name);
72         // Calculate and display SGPA
73         double sgpa = student.calculateSGPA();
74         System.out.println("\nSGPA: " + sgpa);
75         scanner.close();
76     }
77 }
```

Output :

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java sgpa
Enter USN: 1BM22CS009
Enter Name: Abhishek Yadav
Enter details for each subject :

Enter credits for Subject 1: 4
Enter marks for Subject 1: 85
Enter credits for Subject 2: 4
Enter marks for Subject 2: 95
Enter credits for Subject 3: 3
Enter marks for Subject 3: 80
Enter credits for Subject 4: 4
Enter marks for Subject 4: 90
Enter credits for Subject 5: 3
Enter marks for Subject 5: 90
Enter credits for Subject 6: 2
Enter marks for Subject 6: 90
Enter credits for Subject 7: 2
Enter marks for Subject 7: 85
Enter credits for Subject 8: 2
Enter marks for Subject 8: 79

Student Details :
USN : 1BM22CS009
Name : Abhishek Yadav

SGPA: 9.458333333333334
```


LAB-3: BOOK DETAILS

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.

```
BookProg.java X
BookProg.java
1  import java.util.Scanner;
2
3  class Book {
4      Scanner reader = new Scanner(System.in);
5      String name, author;
6      int price, num_pages;
7
8      Book() {
9          setDetails();
10     }
11
12     void setDetails() {
13         System.out.println("\n Enter the name, author, price, number of pages in order: ");
14         name = reader.next();
15         author = reader.next();
16         price = reader.nextInt();
17         num_pages = reader.nextInt();
18     }
19
20     public String toString() {
21         return "\n Details of book: \nName: " + name + "\nAuthor: " + author + "\nPrice: " + price + "\nNum pages: "
22             + num_pages;
23     }
24
25     void getDetails() {
26         System.out.println(toString());
27     }
28 }
```

```
29
30 public class BookProg {
31     public static void main(String[] args) {
32         int i, n;
33         System.out.println("Enter the value of n: ");
34         Scanner mains = new Scanner(System.in);
35         n = mains.nextInt();
36         Book[] books = new Book[n];
37         for (i = 0; i < n; i++) {
38             Book draft = new Book();
39             books[i] = draft;
40         }
41         for (i = 0; i < n; i++) {
42             books[i].getDetails();
43         }
44         System.out.println("\nBM22CS009 Abhishek Yadav");
45     }
46 }
47
```

OUTPUT :

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java BookProg
Enter the value of n:
2

Enter the name, author, price, number of pages in order:
Nuclear
Abhishek
299
121

Enter the name, author, price, number of pages in order:
Reaction
Yadav
399
151

Details of book:
Name: Nuclear
Author: Abhishek
Price: 299
Num pages: 121

Details of book:
Name: Reaction
Author: Yadav
Price: 399
Num pages: 151

1BM22CS009 Abhishek Yadav
```

LAB-4: AREA CALCULATION

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.

```
areas.java X
areas.java
1  import java.util.Scanner;
2
3  abstract class Shape {
4      double dimension1;
5      double dimension2;
6
7      // Abstract method to print the area
8      public abstract void printArea();
9  }
10
11 class Rectangle extends Shape {
12     // Constructor to initialize dimensions
13     public Rectangle(double length, double width) {
14         this.dimension1 = length;
15         this.dimension2 = width;
16     }
17
18     // Implementation of abstract method to print area
19     @Override
20     public void printArea() {
21         double area = dimension1 * dimension2;
22         System.out.println("Area of Rectangle: " + area);
23     }
24 }
25
26 class Triangle extends Shape {
27     // Constructor to initialize dimensions
28     public Triangle(double base, double height) {
29         this.dimension1 = base;
30         this.dimension2 = height;
31     }
```

```

32
33     // Implementation of abstract method to print area
34     @Override
35     public void printArea() {
36         double area = 0.5 * dimension1 * dimension2;
37         System.out.println("Area of Triangle: " + area);
38     }
39 }
40
41 class Circle extends Shape {
42     // Constructor to initialize dimension
43     public Circle(double radius) {
44         this.dimension1 = radius;
45     }
46
47     // Implementation of abstract method to print area
48     @Override
49     public void printArea() {
50         double area = Math.PI * dimension1 * dimension1;
51         System.out.println("Area of Circle: " + area);
52     }
53 }
54
55 public class areas {
56     public static void main(String[] args) {
57         System.out.println("Abhishek Yadav");
58         System.out.println("1BM22CS009\n");
59         Scanner scanner = new Scanner(System.in);
60
61         System.out.println("Choose a shape To Calculate Area :");
62         System.out.println("1. Rectangle");
63         System.out.println("2. Triangle");
64         System.out.println("3. Circle");
65         System.out.println("4. ..Exit..\n ");

```

```

67     int run = 1;
68     while(run==1){
69         System.out.print("Enter Choice : ");
70         int choice = scanner.nextInt();
71         switch (choice) {
72             case 1:
73                 System.out.print("Enter length of Rectangle: ");
74                 double length = scanner.nextDouble();
75                 System.out.print("Enter width of Rectangle: ");
76                 double width = scanner.nextDouble();
77                 Rectangle rectangle = new Rectangle(length, width);
78                 rectangle.printArea();
79                 break;
80             case 2:
81                 System.out.print("Enter base of Triangle: ");
82                 double base = scanner.nextDouble();
83                 System.out.print("Enter height of Triangle: ");
84                 double height = scanner.nextDouble();
85                 Triangle triangle = new Triangle(base, height);
86                 triangle.printArea();
87                 break;
88             case 3:
89                 System.out.print("Enter radius of Circle: ");
90                 double radius = scanner.nextDouble();
91                 Circle circle = new Circle(radius);
92                 circle.printArea();
93                 break;
94             case 4:
95                 run=0;
96             default:
97                 System.out.println("Invalid choice. Please choose a valid choice....");
98         }
99         System.out.println("");
100     }
101     scanner.close();
102 }

```

OUTPUT :

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java areas
Abhishek Yadav
1BM22CS009

Choose a shape To Calculate Area :
1. Rectangle
2. Triangle
3. Circle
4. ..Exit..

Enter Choice : 1
Enter length of Rectangle: 2
Enter width of Rectangle: 2
Area of Rectangle: 4.0

Enter Choice : 2
Enter base of Triangle: 2
Enter height of Triangle: 2
Area of Triangle: 2.0

Enter Choice : 3
Enter radius of Circle: 2
Area of Circle: 12.566370614359172

Enter Choice : 4
Invalid choice. Please choose a valid choice....
```

LAB-5: BANK ACCOUNT DETAILS

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.

```
Bank.java X
Bank.java
1  import java.util.Scanner;
2
3  class Account {
4      String customerName;
5      int accountNumber;
6      String accountType;
7      double balance;
8
9      public Account(String customerName, int accountNumber, String accountType) {
10         this.customerName = customerName;
11         this.accountNumber = accountNumber;
12         this.accountType = accountType;
13         this.balance = 0.0;
14     }
15
16     public void deposit(double amount) {
17         balance += amount;
18         System.out.println("Deposit successful. Updated balance: " + balance);
19         checkMinimumBalance();
20     }
21
22     public void displayBalance() {
23         System.out.println("Account Balance: " + balance);
24     }
25 }
```

```

26     public void withdraw(double amount) {
27         if (amount <= balance) {
28             balance -= amount;
29             System.out.println("Withdrawal successful. Updated balance: " + balance);
30             checkMinimumBalance();
31         } else {
32             System.out.println("Insufficient funds for withdrawal.");
33         }
34     }
35
36     public void checkMinimumBalance() {
37         // Default implementation does nothing; override in specific account types if needed
38     }
39
40     public void computeInterest() {
41         // Default implementation does nothing; override in specific account types if needed
42     }
43 }
44
45 class CurAcct extends Account {
46     double minimumBalance;
47     double serviceCharge;
48
49     public CurAcct(String customerName, int accountNumber) {
50         super(customerName, accountNumber, "Current");
51         this.minimumBalance = 1000.0; // Example minimum balance requirement
52         this.serviceCharge = 50.0;    // Example service charge amount
53     }
54
55     @Override
56     public void checkMinimumBalance() {
57         if (balance < minimumBalance) {
58             balance -= serviceCharge;
59             System.out.println("Service charge imposed. Updated balance: " + balance);
60         }
61     }
62 }

```

```

64 class SavAcct extends Account {
65     double interestRate;
66
67     public SavAcct(String customerName, int accountNumber) {
68         super(customerName, accountNumber, "Savings");
69         this.interestRate = 0.05; // Example interest rate (5%)
70     }
71
72     @Override
73     public void computeInterest() {
74         double interest = balance * interestRate;
75         balance += interest;
76         System.out.println("Interest computed and deposited. Updated balance: " + balance);
77     }
78 }
79

```



```

80 public class Bank {
81     public static void main(String[] args) {
82         Scanner scanner = new Scanner(System.in);
83
84         System.out.print("Enter customer name: ");
85         String customerName = scanner.nextLine();
86
87         System.out.print("Enter account number: ");
88         int accountNumber = scanner.nextInt();
89
90         System.out.print("Enter account type (Current/Savings): ");
91         String accountType = scanner.next();
92
93         Account bankAccount;
94         if (accountType.equalsIgnoreCase("Current")) {
95             bankAccount = new CurAcct(customerName, accountNumber);
96         } else if (accountType.equalsIgnoreCase("Savings")) {
97             bankAccount = new SavAcct(customerName, accountNumber);
98         } else {
99             System.out.println("Invalid account type. Please enter either 'Current' or 'Savings'.");
100             scanner.close();
101             return;
102         }
103
104         int choice;
105         do {
106             System.out.println("Choose an option:");
107             System.out.println("1. Deposit");
108             System.out.println("2. Withdraw");
109             System.out.println("3. Display Balance");
110             System.out.println("4. Compute Interest (Savings)");
111             System.out.println("0. Exit");
112             System.out.print("Enter your choice: ");
113             choice = scanner.nextInt();
114

```

```

115             switch (choice) {
116                 case 1:
117                     System.out.print("Enter deposit amount: ");
118                     double depositAmount = scanner.nextDouble();
119                     bankAccount.deposit(depositAmount);
120                     break;
121                 case 2:
122                     System.out.print("Enter withdrawal amount: ");
123                     double withdrawalAmount = scanner.nextDouble();
124                     bankAccount.withdraw(withdrawalAmount);
125                     break;
126                 case 3:
127                     bankAccount.displayBalance();
128                     break;
129                 case 4:
130                     if (bankAccount instanceof SavAcct) {
131                         ((SavAcct) bankAccount).computeInterest();
132                     } else {
133                         System.out.println("Interest computation is only applicable to Savings accounts.");
134                     }
135                     break;
136                 case 0:
137                     System.out.println("Exiting...");
138                     break;
139                 default:
140                     System.out.println("Invalid choice. Please enter a valid option.");
141             }
142             System.out.println("");
143         } while (choice != 0);
144
145         scanner.close();
146     }
147 }

```

OUTPUT :

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java Bank
Enter customer name: Abhishek Yadav
Enter account number: 12345
Enter account type (Current/Savings): Savings
Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings)
0. Exit
Enter your choice: 1
Enter deposit amount: 1000
Deposit successful. Updated balance: 1000.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings)
0. Exit
Enter your choice: 2
Enter withdrawal amount: 300
Withdrawal successful. Updated balance: 700.0

Choose an option:
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings)
0. Exit
Enter your choice: 0
Exiting...
```

LAB-6: CALCULATION OF MARKS

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.

```
Internals.java X
Internals.java
1  package CIE;
2
3  public class Internals
4  {
5      public int im[]=new int[5];
6  }
```

```
External.java X
External.java
1  package SEE;
2  import CIE.Student;
3  public class External extends Student
4  {
5      // instance variables - replace the example below with your own
6      public int sm[]=new int[5];
7  }
8
```

```
Student.java X
Student.java
1  package CIE;
2  import java.util.*;
3  public class Student
4  {
5      // instance variables - replace the example below with your own
6      public int sem;
7      public String usn;
8      public String name;
9
10     public void accept()
11     {
12         Scanner scan = new Scanner(System.in);
13         System.out.println("Enter U, N, S:\n");
14         usn=scan.nextLine();
15         name=scan.nextLine();
16         sem=scan.nextInt();
17     }
18 }
```

FinalMarks.java X

FinalMarks.java

```
1
2  import java.util.*;
3  import SEE.*;
4  import CIE.*;
5  public class FinalMarks
6  {
7      public static void main(String args[])
8      {
9          int fm[]=new int[5];
10         Scanner sc= new Scanner(System.in);
11         System.out.println("Enter n: ");
12         int n=sc.nextInt();
13         SEE.External st[]=new SEE.External[n];
14         CIE.Internals s[]=new CIE.Internals[n];
15         for(int i=0; i<n; i++)
16         {
17             st[i]=new SEE.External();
18             s[i]=new CIE.Internals();
19             System.out.println("Enter details "+(i+1));
20             st[i].accept();
21             for(int j=0; j<5; j++)
22             {
23                 System.out.println("Enter im and sm of sub "+(j+1));
24                 s[i].im[j]=sc.nextInt();
25                 st[i].sm[j]=sc.nextInt();
26                 fm[j]=s[i].im[j]+st[i].sm[j];
27             }
28             System.out.println("Final marks of "+st[i].name);
29             for(int k=0; k<5; k++)
30             {
31                 System.out.println("Course "+(k+1)+" = "+fm[k]);
32             }
33         }
34     }
35 }
```

OUTPUT :

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java FinalMarks
Enter n:
1
Enter details 1
Enter U, N, S:

123
Abhishek
1
Enter im and sm of sub 1
40
50
Enter im and sm of sub 2
45
45
Enter im and sm of sub 3
46
48
Enter im and sm of sub 4
29
39
Enter im and sm of sub 5
50
50
Final marks of Abhishek
Course 1 = 90
Course 2 = 90
Course 3 = 94
Course 4 = 68
Course 5 = 100
```

LAB-7: EXCEPTION HANDLING

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.

```
ExceptionInheritanceDemo.java X
ExceptionInheritanceDemo.java
1  class WrongAge extends Exception {
2      public WrongAge(String message) {
3          super(message);
4      }
5  }
6
7  class Father {
8      private int age;
9
10     public Father(int age) throws WrongAge {
11         if (age < 0) {
12             throw new WrongAge("Age cannot be negative");
13         }
14         this.age = age;
15     }
16
17     public int getAge() {
18         return age;
19     }
20 }
21
22 class Son extends Father {
23     private int sonAge;
24
25     public Son(int fatherAge, int sonAge) throws WrongAge {
26         super(fatherAge);
27         if (sonAge >= fatherAge) {
28             throw new WrongAge("Son's age cannot be greater than or equal to father's age");
29         }
30         this.sonAge = sonAge;
31     }
32
33     public int getSonAge() {
34         return sonAge;
35     }
36 }
37
```

```

38 public class ExceptionInheritanceDemo {
39     public static void main(String[] args) {
40         System.out.println("Abhishek Yadav");
41         System.out.println("1BM22CS009");
42         try {
43             Father father = new Father(45);
44             Son son = new Son(45, 40);
45
46             System.out.println("Father's age: " + father.getAge());
47             System.out.println("Son's age: " + son.getSonAge());
48         } catch (WrongAge e) {
49             System.out.println("Exception caught: " + e.getMessage());
50         }
51     }
52 }
53 }

```

OUTPUT :

```

C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java ExceptionInheritanceDemo
Abhishek Yadav
1BM22CS009
Father's age: 45
Son's age: 40

```

LAB-8: MULTITHREADING

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.

```
threadDemo.java X
threadDemo.java
1  class DisplayMessage extends Thread {
2      String message;
3      int interval;
4
5      public DisplayMessage(String message, int interval) {
6          this.message = message;
7          this.interval = interval;
8      }
9
10     public void run() {
11         while (true) {
12             try {
13                 System.out.println(message);
14                 Thread.sleep(interval * 1000); // Convert seconds to milliseconds
15             } catch (InterruptedException e) {
16                 e.printStackTrace();
17             }
18         }
19     }
20 }
21
22 public class threadDemo {
23     public static void main(String[] args) {
24         System.out.println("Abhishek Yadav");
25         System.out.println("1BM22CS009");
26         DisplayMessage thread1 = new DisplayMessage("BMS College of Engineering", 10);
27         DisplayMessage thread2 = new DisplayMessage("CSE", 2);
28
29         thread1.start();
30         thread2.start();
31     }
32 }
```


OUTPUT :

```
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>java threadDemo
Abhishek Yadav
1BM22CS009
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
^C
C:\Users\Abhishek Yadav\OneDrive\Desktop\Java>
```

LAB-9: AWT

09) 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 `ArithmeticException`. Display the exception in a message dialog box.

```
SwingDemo.java X
SwingDemo.java
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.*;
4
5  class SwingDemo{
6      SwingDemo(){
7          // create jframe container
8          JFrame jfrm = new JFrame("Divider App");
9          jfrm.setSize(275, 150);
10         jfrm.setLayout(new FlowLayout());
11         // to terminate on close
12         jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13
14         // text label
15         JLabel jlab = new JLabel("Enter the dividend and divider:");
16
17         // add text field for both numbers
18         JTextField ajtf = new JTextField(8);
19         JTextField bjtf = new JTextField(8);
20
21         // calc button
22         JButton button = new JButton("Calculate");
23
24         // labels
25         JLabel err = new JLabel();
26         JLabel alab = new JLabel();
27         JLabel blab = new JLabel();
28         JLabel anslab = new JLabel();
29     }
```

```

30         // add in order :)
31         jfrm.add(err); // to display error boi
32         jfrm.add(jlab);
33         jfrm.add(ajtf);
34         jfrm.add(bjtf);
35         jfrm.add(button);
36         jfrm.add(alab);
37         jfrm.add(blab);
38         jfrm.add(anslab);
39
40         ActionListener l = new ActionListener() {
41             public void actionPerformed(ActionEvent evt) {
42                 System.out.println("Action event from a text field");
43             }
44         };
45         ajtf.addActionListener(l);
46         bjtf.addActionListener(l);
47
48         button.addActionListener(new ActionListener() {
49             public void actionPerformed(ActionEvent evt) {
50                 try{
51                     int a = Integer.parseInt(ajtf.getText());
52                     int b = Integer.parseInt(bjtf.getText());
53                     int ans = a/b;
54
55                     alab.setText("\nA = " + a);
56                     blab.setText("\nB = " + b);
57                     anslab.setText("\nAns = "+ ans);
58                 }

```

```

59                 catch(NumberFormatException e){
60                     alab.setText("");
61                     blab.setText("");
62                     anslab.setText("");
63                     err.setText("Enter Only Integers!");
64                 }
65                 catch(ArithmeticException e){
66                     alab.setText("");
67                     blab.setText("");
68                     anslab.setText("");
69                     err.setText("B should be NON zero!");
70                 }
71             }
72         });
73
74         // display frame
75         jfrm.setVisible(true);
76     }
77
78     public static void main(String args[]){
79         // create frame on event dispatching thread
80         System.out.println("Abhishek Yadav");
81         System.out.println("1BM22CS009\n");
82         SwingUtilities.invokeLater(new Runnable(){
83             public void run(){
84                 new SwingDemo();
85             }
86         });
87     }
88 }

```

Output :

