VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

M S Shailesh (1BM23CS172)

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
Sep-2024 to Jan-2025

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 Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **M S Shailesh(1BM23CS172),** who is bonafide student 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 Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Srushti C S Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Sl. No.	Date	Experiment Title	Page No.
1	30/9/24	Quadratic Equation	4-6
2	7/10/24	SGPA calculation	7-11
3	14/10/24	Book Details	12-14
4	21/10/24	Area of shapes using abstract methods	15-18
5	28/10/24	Bank class using Inheritance	19-27
6	12/11/24	Packages	28-32
7	28/11/24	User defined Exceptions	33-35
8	28/11/24	Multithreading	36-37
9	28/11/24	User Interface for division program	38-41
10	28/11/24	Demonstration of IPC and Deadlock	42-48

Github Link:
https://github.com/msshailesh45/javaLab/tree/main

Program 1

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

Algorithm:

⑧ 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 -ve, display message that there are no solutions. [lab program 1]

```
import java.util.*;
class quadratic
{ int a,b,c;
  double r1,r2,d;
  void getd()
  {
    System.out.println("Enter the value of a,b,c");
    Scanner s=new Scanner(System.in);
    a=s.nextInt();
    if (a==0)
    {  System.out.println("not a quadratic equation, enter
       a=s.nextInt();                                  other value");
    }
    b=s.nextInt();
    c=s.nextInt();
  }
  void compute()
  {
    d=b*b-4*a*c;
    if (d==0)
    {
      r1=(-b)/(2*a);
      r2=r1;
      System.out.println("Roots of the equation is\n"+r1+"\n"+r2);
    }
    else if (d>0)
    {  r1=((-b)+(Math.sqrt(d)))/(double)(2*a);
       r2=((-b)+(Math.sqrt(d)))/(double)(2*a);
    }  System.out.println("roots of the equation"+r1+r2);
```

```
    else
    System.out.println("equation has imaginary roots");
  }
}
class QuadraticEquation
{
  public static void main(String arg[])
  {
    quadratic q=new quadratic();
    q.getd();
    q.compute();
    System.out.println("Name: MS Shailesh");
    System.out.println("USN: 1BM23CS172");
  }
}
```

Output

```
Enter the value a,b,c
1
-4
4
roots of equation is
2
2
=
Enter the value a,b,c
1
-7
-3
roots of the equation is
7.405
-0.405

Enter the value a,b,c
2
1
1
quadratic equation has imaginary roots
```

Code:

```
import java.util.*;
class quadratic
{
   int a,b,c;
   double r1,r2,d;
   void getd()
   {
      System.out.println("enter the value of a,b,c");
      Scanner s=new Scanner(System.in);
      a=s.nextInt();
      if(a==0)
      {
        System.out.println("not a quadratic equation, enter the value of a other then 0");
        a=s.nextInt();
      }
      b=s.nextInt();
      c=s.nextInt();
   }

   void compute()
   {
      d=b*b-4*a*c;
      if(d==0)
      {
         r1=(-b)/(2*a);
         r2=r1;
         System.out.println("Roots of the equation is\n"+r1+"\n"+r2);
      }else if(d>0)
      {
         r1=((-b)+(Math.sqrt(d)))/(double)(2*a);
         r2=((-b)-(Math.sqrt(d)))/(double)(2*a);
             System.out.println("Roots of the equation is\n"+r1+"\n"+r2);
      }
      else
      {
         System.out.println("quadratic equation has imaginary roots");
      }
   }
}
class QuadraticEquation
{
   public static void main(String[] args)
   {
      quadratic q=new quadratic();
      q.getd();
      q.compute();
      System.out.println("Name: M S Shailesh");
      System.out.println("USN: 1BM23CS172");
   }
}
```

Output;

```
D:\1BM23CS172(java)>java QuadraticEquation
enter the value of a,b,c
1
-4
4
Roots of the equation is
2.0
2.0
Name: M S Shailesh
USN: 1BM23CS172

D:\1BM23CS172(java)>java QuadraticEquation
enter the value of a,b,c
1
-7
-3
Roots of the equation is
7.405124837953327
-0.405124837953327
Name: M S Shailesh
USN: 1BM23CS172

D:\1BM23CS172(java)>java QuadraticEquation
enter the value of a,b,c
2
1
1
quadratic equation has imaginary roots
Name: M S Shailesh
USN: 1BM23CS172

D:\1BM23CS172(java)>java QuadraticEquation
enter the value of a,b,c
0
not a quadratic equation, enter the value of a other then 0
1
-7
-3
Roots of the equation is
7.405124837953327
-0.405124837953327
Name: M S Shailesh
USN: 1BM23CS172
```

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

Algorithm:

ⓐ 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 students [lab program 2]

Ans:- ① Array of objects
② this keyword.

```
import java.util.*;
class subject
{ int subjectMarks, credits, grade; }
class student
{
    String name, usn;
    double SGPA, totalCredits = 0, totalNum=0;
    Scanner s=new Scanner(System.in);
    subject[] subjects;
    student()
    {
        subjects =new subject[8];
        for (int i=0; i<8 ; i++)
            subjects[i] =new subject();
    }
    void getstudentDetails()
    {
        System.out.println("Enter the name");
        name=s.nextLine();
        System.out.println("Enter the usn");
        USN = s.nextline();
    }
    void getMarks()
    {
        for (int i=0; i<8; i++)
        {
            s.system.out.println("Enter the marks"+(i+1)+"subject");
            subjects[i]. subjectMarks = s.nextInt();
            System.out.println("Enter the credits of "+(i+1)+"sub");
            subjects[i]. credits = s.nextInt();
```

```
            subjects[i]. grade = ((subjects[i]. subject marks)/10)+1;
            if(subjects[i].grade==11)
                subjects[i]. grade =10;
            else if (subjects[i]. grade ==4)
                subjects[i]. grade =0;
        }
    }
    void computeSGPA()
    {
        for (int i=0; i<8; i++)
        {
            totalCredits = totalcredits + subjects[i]. credits;
            totalNum = totalNum +(subjects[i]. grade * subjects[i]
                                credits);
        }
        SGPA = (totalNum/totalcredits);
    }
    Void display()
    {
        System.out.println("Name:"+name);
        System.out.println("USN:"+USN);
        System.out.println("SGPA:"+SGPA);
    }
    Public static void main (String args[])
    {
        Scanner Sc=new Scanner(system.in)
        System.out.println("Enter the number of students");
        int n;
        n= Sc.nextInt();
        student[] s1 =new student[n];
        for (int i=0; i<n; i++)
        {
            s1[i] = new student();
        }
        for(int i=0; i<2; i++)
        {
            s1[i]. getstudentDetails();
            s1[i]. getMarks();
            s1[i]. computeSGPA();
            s1[i]. display();
        }
    }
}
```

Output

Enter the number of students
2
Enter the name:
Karthik
Enter the USN
1BM23CS172
Enter the marks 1 subject
99
Enter the credits 1 subject
4
Enter the marks 2 subject
90
Enter the credits 2 subject
4
Enter the marks 3 subject
87
Enter the credits 3 subject
3
Enter the marks 4 subject
88
Enter the credits 4 subject
3
Enter the marks 5 subject
78
Enter the credits 5 subject
3
Enter the marks 6 subject
100
Enter the credits 6 subject
1
Enter the marks 7 subject
98
Enter the credits 7 subject
1
Enter the marks 8 subject
90
Enter the credits 8 subject
1
name: Karthik
USN: 1BM23CS172
SGPA: 9.4

Enter the name:
LikhithD
Enter the USN:
1BM23CS170
Enter the marks 1 subject
98
Enter the credits 1 subject
4
Enter the marks 2 subject
87
Enter the credits 2 subject
4
Enter the marks 3 subject
87
Enter the credits 3 subject
3
Enter the marks 4 subject
88
Enter the credits 4 subject
3
Enter the marks 5 subject
97
Enter the credits 5 subject
3
Enter the marks 6 subject
99
Enter the credits 6 subject
1
Enter the marks 7 subject
100
Enter the credits 7 subject
1
Enter the marks 8 subject
100
Enter the credits 8 subject
1
name: LikhithD
USN: 1BM23CS170
SGPA: 9.2
Name: M.S. Shailesh
USN: 1BM23CS172

8/10/24

10/10

code:

```
import java.util.*;
class subject
{
	int subjectMarks,credits,grade;
}
class student
{
	String name,usn;
	double SGPA,totalCredits=0,totalNum=0;
	Scanner s=new Scanner(System.in);
	subject[] subjects;
	student()
	{
		subjects=new subject[8];
		for(int i=0;i<8;i++)
			subjects[i]=new subject();
	}
	void getStudentDetails()
	{
		System.out.println("enter the name");
		name=s.nextLine();
		System.out.println("enter the usn");
		usn=s.nextLine();
	}
	void getMarks()
	{
		for(int i=0;i<8;i++)
		{
			System.out.println("enter the marks of "+(i+1)+" subject");
			subjects[i].subjectMarks=s.nextInt();
			System.out.println("enter the credits of "+(i+1)+" subject");
			subjects[i].credits=s.nextInt();
			subjects[i].grade=((subjects[i].subjectMarks)/10)+1;
			if(subjects[i].grade==11)
				subjects[i].grade=10;
			else if(subjects[i].grade==4)
				subjects[i].grade=0;
		}
	}
	void computeSGPA()
	{
		for(int i=0;i<8;i++)
		{
			totalCredits=totalCredits+subjects[i].credits;
			totalNum=totalNum+(subjects[i].grade*subjects[i].credits);
		}
		SGPA=(totalNum/totalCredits);
	}
```

```
        void display()
        {
                System.out.println("Name:"+name);
                System.out.println("USN:"+usn);
                System.out.println("SGPA:"+SGPA);
        }
        public static void main(String args[])
        {
                int n;
                System.out.println("Enter the number of students");
                Scanner Sc=new Scanner(System.in);
                n=Sc.nextInt();
                student[] s1=new student[n];
                for(int i=0;i<n;i++)
                {
                        s1[i]=new student();
                }

                for(int i=0;i<n;i++)
                {
                        s1[i].getStudentDetails();
                        s1[i].getMarks();
                        s1[i].computeSGPA();
                        s1[i].display();
                }
                System.out.println("Name: M S Shailesh");
                System.out.println("USN: 1BM23CS172");
        }
}
```

Output:

```
enter the name
Likhith D
enter the usn
1BM23CS170
enter the marks of 1 subject
99
enter the credits of 1 subject
4
enter the marks of 2 subject
90
enter the credits of 2 subject
4
enter the marks of 3 subject
87
enter the credits of 3 subject
3
enter the marks of 4 subject
88
enter the credits of 4 subject
3
enter the marks of 5 subject
78
enter the credits of 5 subject
3
enter the marks of 6 subject
100
enter the credits of 6 subject
1
enter the marks of 7 subject
90
enter the credits of 7 subject
1
enter the marks of 8 subject
98
enter the credits of 8 subject
1
Name:Likhith D
USN:1BM23CS170
SGPA:9.4
Name: M S Shailesh
USN: 1BM23CS172
```

```
Enter the number of students
2
enter the name
Karthik
enter the usn
1BM23CS172
enter the marks of 1 subject
98
enter the credits of 1 subject
4
enter the marks of 2 subject
87
enter the credits of 2 subject
4
enter the marks of 3 subject
67
enter the credits of 3 subject
3
enter the marks of 4 subject
88
enter the credits of 4 subject
3
enter the marks of 5 subject
97
enter the credits of 5 subject
3
enter the marks of 6 subject
99
enter the credits of 6 subject
1
enter the marks of 7 subject
100
enter the credits of 7 subject
1
enter the marks of 8 subject
100
enter the credits of 8 subject
1
Name:Karthik
USN:1BM23CS172
SGPA:9.2
```

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.

Algorithm:

14/10/24

10.) Create class Book which contains four members:
→name → Price
→author → 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.

[lab program 3]

```
import java.util.*;
Class Books
{
    String name;
    String author;
    int price;
    int numPages;
    Books(String name, String author, int price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numpages = numpages;
    }
    Public String toString()
    {
        String name, author, price, numpages;
        name = "Book name:" + this.name + "\n";
        author = "Author name:" + this.author + "\n";
        price = "Price :" + this.price + "\n";
        numpages = "Number of pages:" + this.numpages + "\n";
    }
}
Class bookDetails
{ Public static void main(String args[])
  { Scanner s=new Scanner(System.in);
    int n;
    String name, author;
    int price, numpages;
    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("Enter the name:");
      name = s.next();
      system.out.println("Enter the author");
      author = s.next();
      system.out.println("Enter the price:");
      price = s.nextInt();
      System.ou.println("Enter the number of pages");
      numPages = s.nextInt();
      b[i] =new Books(name, author, prince, numpages);
    }
    for (int i=0; i<n; i++)
    {
      system.out.println(b[i]);
    }
  }
}
```

Output

```
Enter the number of books
2
Enter the name:
Math Textbook
Enter the author:
Shailesh
Enter the price:
300
Enter the numpages:
500
Enter the name:
Science Textbook
Enter the author:
likithP
Enter the price:
450
Enter the numpages:
600
Book name: Math textbook
Author name: shailesh
Price: 300
num of pages:500
Book name : Science Textbook
Author name: Likith P
Price : 450
Number of pages : 600
Name: Nas.shailesh
USN: 1BM23CS172
```

14/10/24

code:

```
import java.util.*;
class Books
{
	String name;
	String author;
	int price;
	int numPages;
	Books(String name, String author, int price, int numPages)
	{
		this.name = name;
		this.author = author;
		this.price = price;
		this.numPages = numPages;
	}
	public String toString()
	{
		String name, author, price, numPages;
		name = "Book name: " + this.name + "\n";
		author = "Author name: " + this.author + "\n";
		price = "Price: " + this.price + "\n";
		numPages = "Number of pages: " + this.numPages + "\n";
		return name + author + price + numPages;
	}
}

class bookDetails
{
	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("Enter the number of books");
		n = s.nextInt();
		Books[] b = new Books[n];
		for(int i=0;i<n;i++)
		{
			System.out.println("Enter the Name:");
			name=s.next();
			System.out.println("Enter the Author:");
			author=s.next();
			System.out.println("Enter the Price:");
			price=s.nextInt();
			System.out.println("Enter the Number of pages:");
			numPages=s.nextInt();
			b[i]=new Books(name,author,price,numPages);
```

```
                }


                for(int i=0;i<n;i++)
                {
                        System.out.println(b[i]);
                }
                System.out.println("Name:M S Shailesh");
                System.out.println("USN: 1BM23CS172");
        }
}
```

Output:

```
D:\1BM23CS172(java)>javac bookDetails.java

D:\1BM23CS172(java)>javac bookDetails.java

D:\1BM23CS172(java)>java bookDetails
Enter the number of books
2
Enter the Name:
MathTextBook
Enter the Author:
Shailesh
Enter the Price:
300
Enter the Number of pages:
500
Enter the Name:
ScienceTextBook
Enter the Author:
LikhithD
Enter the Price:
450
Enter the Number of pages:
600
Book name: MathTextBook
Author name: Shailesh
Price: 300
Number of pages: 500

Book name: ScienceTextBook
Author name: LikhithD
Price: 450
Number of pages: 600

Name:M S Shailesh
USN: 1BM23CS172
```

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.

Algorithm:

21/10/24

11) 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.

[lab program 4]

```
Abstract class Shape
{
  Abstract printArea();
}

class Rectangle extends Shape
{
   double length, breath;
   Scanner s=new Scanner(System.in);
   Rectangle()
   {
      System.out.println("Enter the dimension of rectangle (length, breath)");
      length = s.nextDouble();
      breath = s.nextDouble();
   }
   void printArea()
   {
      System.out.println("Area of Rectangle :"+(length*breath
   }
}
```

```
class Triangle extends Shape
{ Double base, height;
  Scanner s=new Scanner(System.in);
  Triangle()
  { System.out.println("Enter the dimension of Triangle (base, height)");
    base = s.nextDouble();
    height = s.nextDouble();
  }
  void printArea()
  { System.out.println("Area of the triangle :"+(0.5*base
                                                 *height));
  }
}

class circle extends Area
{ Double radius
  Scanner s=new Scanner(System.in);
  Circle()
  { System.out.Println("Enter the dimension of circle (radius)");
    radius =s.nextDouble();
  }
  void PrintArea()
  { System.out.println("Area of the circle :"+(3.14*radius*
                                                 radius));
  }
}
Class Area
{
  Public static void main(String args[])
  {
    Rectangle r=new Rectangle();
    Triangle t=new Triangle();
    Circle c=new circle();
    r.printArea();
    t.printArea();
    c.printArea();
  }
}
```

Output

Enter the dimension of Rectangle (length, breath)
45
60
Enter the dimension of Triangle (base, height).
4
3
Enter the dimension of circle (radius)
7
Area of the Rectangle : 2700.0
Area of the Triangle : 6.0
Area of the circle : 153.86

21/10/24

code:

```
import java.util.*;
abstract class shape
{
	abstract void printArea();
}

class Rectangle extends shape
{
	Double length,breath;
	Scanner s=new Scanner(System.in);
	Rectangle()
	{
		System.out.println("Enter the dimension of rectangle(length,breath)");
		length=s.nextDouble();
		breath=s.nextDouble();
	}
	void printArea()
	{
		System.out.println("Area of the Rectangle : "+(length*breath));
	}
}

class Triangle extends shape
{
	Double base,height;
	Scanner s=new Scanner(System.in);
	Triangle()
	{
		System.out.println("Enter the dimension of Triangle(base,height)");
		base=s.nextDouble();
		height=s.nextDouble();
	}
	void printArea()
	{
		System.out.println("Area of the Triangle : "+(0.5*base*height));
	}
}

class Circle extends shape
{
	Double radius;
	Scanner s=new Scanner(System.in);
	Circle()
	{
		System.out.println("Enter the dimension of Circle(radius)");
		radius=s.nextDouble();
	}
	void printArea()
	{
		System.out.println("Area of the Circle : "+(3.14*radius*radius));
	}
}
```

```
class Area
{
        public static void main(String args[])
        {
                Rectangle r=new Rectangle();
                Triangle t=new Triangle();
                Circle c=new Circle();
                r.printArea();
                t.printArea();
                c.printArea();
                System.out.println("Name: M S Shailesh");
                System.out.println("USN: 1BM23CS172");
        }
}
```

Output:

```
D:\1BM23CS172(java)>javac Area.java

D:\1BM23CS172(java)>java Area
Enter the dimension of rectangle(length,breath)
45
60
Enter the dimension of Triangle(base,height)
4
3
Enter the dimension of Circle(radius)
7
Area of the Rectangle : 2700.0
Area of the Triangle : 6.0
Area of the Circle : 153.86
Name: M S Shailesh
USN: 1BM23CS172

D:\1BM23CS172(java)>java Area
Enter the dimension of rectangle(length,breath)
43
56
Enter the dimension of Triangle(base,height)
4
6
Enter the dimension of Circle(radius)
8
Area of the Rectangle : 2408.0
Area of the Triangle : 12.0
Area of the Circle : 200.96
Name: M S Shailesh
USN: 1BM23CS172
```

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.

Algorithm:

12) Develop a java program to create a class Bank that maintains two kinds of account for it's customers 1.saving account and 2.current account. The saving account provides compound interest and withdrawal facilities. The current account provides cheque book facility but no interest. current account holders should maintain a min balance and if balance falls below service charge is imposed.

create a class Account that stores name, Acc Num, type of account. from this derive sav-acc and cur-acc and show these properties

→ Deposit → withdraw → interest calc → Account Details

→ service charge imposition. [lab program 5]

```
import java.util.*;
Class Account
{   string customerName;
    int accountNum;
    double balance;
    Account (string customerName, int accountNum, double
                                        balance)
    {   this.customerName = customerName;
        this.accountNum = accountNum;
        this.balance = balance;
    }
    void deposit (double amount)
    {   balance = balance + amount;
        System.out.println (" Balance after Deposit ="+balance)
    }
    void withdraw (double amount)
    {   if (amount > balance)
                System.out.println ("Insufficient balance
                                        in account");
        else
        {   balance = balance - amount;
            System.out.println ("Balance after withdraw"+
                                        balance);
        }
    }
    void get balance()
    {   system.out.println ("Balance =" + balance);
    }
```

```
Class SavingAccount extends Account
{   double InterestRate;
    SavingAccount (String customerName, int accountNumber,
                double balance, double InterestRate)
    {   super (customerName, accountNumber, balance);
        this.InterestRate = InterestRate;
    }
    Void calcInterest()
    {   double Interest = (super.balance * InterestRate)/100;
        balance = balance + Interest;
        System.out.println(" Interest = "+interest + "\n"+"balance
                                    = "+ balance);
    }
}
Class CurrentAccount extends Account
{
    double MIN_BALANCE = 500.0;
    double SERVICE_CHARGE = 50;
    CurrentAccount (String customerName, int accountNumber,
                double balance)
    {   super (customerName, accountNum, balance);
    }
    void withdraw (double amount)
    {   if (amount > balance)
            System.out.println ("Insufficient balance");
        else if (balance - amount < MIN_BALANCE)
        {   System.out.println ("SERVICE charge imposed");
            balance = balance - amount;
            System.out.println ("balance = "+balance);
        }
        else
        {   balance = balance - amount;
            System.out.println ("balance = "+balance);
        }
    }
```

```
class Bank
{ public static void main(String args[])
  {
    Scanner s =new Scanner(System.in);
    System.out.println("Enter the Name, Account number
                        and balance");
    String customerName = s.nextLine();
    int accountNum =s.nextInt();
    double balance= s.nextDouble();
    double interestRate = 5;
    SavingAccount sa=new SavingAccount(customerName, accountNum, balance,
                                       interestRate);
    CurrentAccount ca=new CurrentAccount(customerName, accountNum,
                                         balance);

    System.out.println("Enter the type of account
                        1. SavingAccount , 2. CurrentAccount");

    int type = s.nextInt();
    while (true)
    { System.out.println("------Enter your choice -----");
      System.out.println("1.Deposit \n 2. Withdraw \n 3. Interest
                          \n 4. Account Details \n 5. Exit");

      int count=0
      switch (choice)
      {
        case 1: { System.out.println("Enter the amount");
                  double amount =s.nextDouble();
                  if (type ==1)
                      sa.deposit(amount);
                  else
                      ca.deposit(amount);
                  break;
                }
        case 2: { System.out.println("Enter the amount");
                  double amount = s.nextDouble();
                  if (type ==1)
                      sa.withdraw(amount);
                  else
                      ca.withdraw(amount);
                } break;
```



```
        case 3: { if (type ==1)
                      sa.calcInterest();
                  else
                      System.out.println("not possible for current
                                          account");
                  break;
                }
        case 4: { System.out.println("Customer Name"+customerName);
                  System.out.println("Account number" + accountNum);
                  System.out.println("Account type" + type);
                  if (type ==1)
                      sa.getBalance();
                  else
                      ca.getBalance();
                  break;
                }
        default: System.out.println("Invalid choice");
                 count=1;
                 break;
      }
      if(count ==1)
          break;
    }
  }
}
```

PTO

Output

Enter the name and Account Number and balance

Shailesh

21

1000

Which type of account do you have?

1. saving account
2. current Account

1.

------ Enter your choice ------

1. Deposit
2. withdraw
3. Interest calculation
4. Account Details
5. Exit

1

Enter the amount:

500

Balance after Deposit = 1500.0

------ Enter your choice ------

1. Deposit
2. withdraw
3. Interest calculation
4. Account Details
5. Exit

2

Enter the amount: 300

Balance after withdraw = 1200.0

------ Enter your choice ------

1. Deposit
2. withdraw
3. Interest calculation
4. Account Details
5. Exit

3

interest Rs 60.0

Your new Balance Rs 1260.0

------ Enter your choice ------

1. Deposit
2. withdraw
3. Interest calculate
4. Account Details
5. Exit

Customer Name shailesh

Account number 21

Account type 1

Balance = 1260.0

------ Enter your choice ------

1. Deposit
2. withdraw
3. interest calculate
4. Account Details
5. Exit.

6

Invalid choice.

Output 2

Enter the Name, Account number and Balance

Shailesh

22

700

Which type of account do you have?

1. saving Account
2. current account

2

------ Enter your choice ------

1. Deposit
2. withdraw
3. Interest calculation
4. Account Details
5. Exit

1.

Enter the amount: 200

Balance after Deposit = 900.0

```
-------- Enter your choice -----
1. Deposit
2. withdraw
3. Interest calculation
4. Account Details
5. Exit
2
Enter the amount: 500
service charge will imposed
Balance after service charge imposed. = 350.0
-------- Enter your choice -----
1. Deposit
2. withdraw
3. Interest calculation
4. Account Details
5. Exit
3
not possible for current, account
------- Enter your choice -----
1. Deposit
2. withdraw
3. Interest calculation
4. Account Details
5. Exit
4.
customer name : Shailesh
Account Number : 22
Account type 2
Balance = 350.0
------- Enter your choice -----
1. Deposit
2. withdraw
3. Interest calculation
4. Account Details
5. Exit
5
invalid choice.
```

10/10

30/10/24

code:

```
import java.util.*;
class Account
{
	String customerName;
	int accountNum;
	double balance;

	Account(String customerName,int accountNum,double balance)
	{
		this.customerName=customerName;
		this.accountNum=accountNum;
		this.balance=balance;
	}

	void deposit(double amount)
	{
		balance=balance+amount;
		System.out.println("Balance after Deposit = "+balance);
	}

	void withdraw(double amount)
	{
		if(amount > balance)
			System.out.println("Insufficient Balance in Account");
		else
		{
			balance=balance-amount;
			System.out.println("Balance after Withdraw = "+balance);
		}
	}

	void getBalance()
	{
		System.out.println("Balance = "+balance);
	}
}

class savingAccount extends Account
{
	double interestRate;

	savingAccount(String customerName, int accountNumber, double balance, double interestRate)
	{
	super(customerName, accountNumber, balance);
	this.interestRate = interestRate;
	}

	void calcInterest()
	{
		double interest = (super.balance*interestRate)/100;
		System.out.println("Interest is "+interest);
	}
```

```
}

class currentAccount extends Account
{
	double MIN_BALANCE = 500.0;
	double SERVICE_CHARGE = 50.0;

	currentAccount(String customerName, int accountNumber, double balance)
	{
	super(customerName, accountNumber, balance);
	}

	void withdraw(double amount)
	{
		if(amount > balance)
			System.out.println("Insufficient Balance in Account");
		else if(balance-amount<MIN_BALANCE)
		{
			System.out.println("Service charge will be imposed");
			balance=balance-SERVICE_CHARGE;
			System.out.println("Balance after Service charge imposed = "+balance);
		}
		else
		{
			balance=balance-amount;
			System.out.println("Balance after Withdraw = "+balance);
		}
	}
}

class Bank
{
	public static void main(String args[])
	{
		Scanner s=new Scanner(System.in);
		System.out.println("Enter the Name and Account Number And Balance, Interest Rate");
		String customerName=s.nextLine();
		int accountNum=s.nextInt();
		double balance=s.nextDouble();
		double interestRate=5;
		savingAccount sa=new savingAccount(customerName,accountNum,balance,interestRate);
		currentAccount ca=new currentAccount(customerName,accountNum,balance);
		System.out.println("Which type of account do you have?\n 1.Saving Account\n 2.Current
Account");
		int type=s.nextInt();
		while(true)
		{
			System.out.println("-----------Enter your
choice----------\n1.Deposit\n2.withdraw\n3.Interest Calculation\n4.Account Details\n5.Exit");
			int choice=s.nextInt();
			int count=0;
			switch(choice)
			{
```

```
                    case 1:
                        {
                            System.out.println("Enter the amount : ");
                            double amount=s.nextDouble();
                            if(type==1)
                                sa.deposit(amount);
                            else
                                ca.deposit(amount);
                            break;
                        }
                    case 2:
                        {
                            System.out.println("Enter the amount : ");
                            double amount=s.nextDouble();
                            if(type==1)
                                sa.withdraw(amount);
                            else
                                ca.withdraw(amount);
                            break;
                        }
                    case 3:
                        {
                            if(type==1)
                                sa.calcInterest();
                            else
                                System.out.println("not possible for current
account");
                            break;
                        }
                    case 4:
                        {
                            System.out.println("Customer Name "+customerName);
                            System.out.println("Account Number "+accountNum);
                            System.out.println("Account Type "+type);
                            if(type==1)
                                sa.getBalance();
                            else
                                ca.getBalance();
                            break;
                        }
                    default:System.out.println("Invalid choice");
                        count=1;
                        break;

                }
                if(count==1)
                    System.out.println("Name: M S Shailesh");
                    System.out.println("USN: 1BM23CS172");
                    break;
            }
        }
}
```

Output:

```
D:\1BM23CS172(java)>java Bank
Enter the Name and Account Number And Balance, Interest Rate
shailesh
21
1000
Which type of account do you have?
 1.Saving Account
 2.Current Account
1
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
1
Enter the amount :
500
Balance after Deposit = 1500.0
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit

2
Enter the amount :
300
Balance after Withdraw = 1200.0
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
3
Interest is 60.0
```

```
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
4
Customer Name shailesh
Account Number 21
Account Type 1
Balance = 1200.0
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
5
Invalid choice
Name: M S Shailesh
USN: 1BM23CS172
```

```
D:\1BM23CS172(java)>java Bank
Enter the Name and Account Number And Balance, Interest Rate
Shailesh
22
700
Which type of account do you have?
 1.Saving Account
 2.Current Account
2
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
1
Enter the amount :
200
Balance after Deposit = 900.0
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
2
Enter the amount :
500
Service charge will be imposed
Balance after Service charge imposed = 850.0
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
3
not possible for current account
```

```
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
4
Customer Name Shailesh
Account Number 22
Account Type 2
Balance = 850.0
-----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
5
Invalid choice
Name: M S Shailesh
USN: 1BM23CS172
```

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.

Algorithm:

13) Create a package CIE which has two classes Student and Internals. The class Student has members like USN, name, sem. The class Internal derived from student has an that stores the Internal marks scored in five courses of the current sem of student. create another package SEE which has the class External which is derived class of Student. This class has an array that stores the see marks scored in five courses of the current sem of the student. Import the two packages in a file that declares the final marks of n students in all five courses. (lab program 6)

= CIE folder

```
Pakage CIE;
Import java.util.*;
class student
{
    Public String USN;
    Public String name;
    Public Int Sem;
    Public void InputStudentDetails()
    {
        Scanner s=new Scanner (System.in);
        USN=S.nextLine();
        name=S.nextLine();
        Sem=S.nextInt();
    }
    Public void displayStudentDetails()
    {
        System.out.println("name:"+name+"\n usn"+usn+"\n sem"+sem);
    }
}
public class Internals extends student
{
    Public Int marks[] = new Int [5];
    Public void InputCIEMarks()
    {
        Scanner S=new Scanner (System.in);
        for (int i=0; i<5; i++)
            marks [i]=S.next Int();
    }
}
```

```
package SEE
import CIE.*;
import java.util.*;
public class Externals extends Internals
{
    public Int Seemarks[]=new Int[5];
    public Int finalMarks[]=new Int[5];
    Scanner S=new Scanner (System.in);
    Public void InputSEEmarks()
    {
        for (int i=0; i<5; i++)
            Seemarks[i]=S.nextInt();
    }
    Public void calculateFinalMarks()
    {
        for (int i=0; i<5; i++)
            finalMarks[i]=(mark[i])+(Seemarks[i]/2);
    }
    Public void displayFinalMarks()
    {
        for(int i=0; i<5; i++)
            System.out.println(finalMarks[i]);
    }
}
=
import SEE.*;
import CIE.*;
import java.util.*;
class Main program
{
    Public static void main (String arg[])
    {
        Int n;
        Scanner S=new Scanner (System.in);
        System.out.println("Enter the number of student");
        n=S.nextInt();
        Externals e[]=new Externals [n];
        for(int i=0; i<n; i++)
        {
            e[i]=new externals()
            e[i].InputStudentDetails();
            e[i]. InputCIE Marks();
            e[i]. InputSEE marks ();
            e[i]. calculate Final marks();
            e[i]. display student Details();
            e[i]. display Final marks();
        }
```

Output

```
> Javac CIE/Internals.Java
> Javac SEE/Externals.Java
> Javac Mainprogram.Java
> Java Main program
Enter no of students
2
Enter Name, USN, sem
Shailesh
1BM123CS172
3
Enter the internals marks
44
50
45
42
43
Enter the see marks
88
89
99
97
99
Student details, final marks
Shailesh
1BM123CS172
3
88
94
94
90
92
Enter the name, USN, sem
likith P
1BM23CS170
3
Enter the internals marks
45
50
45
44
43
Enter see marks
88
89
99
90
98
~~Hem~~likith D
1BM23CS170
3
89
94
94
```

Snk
12/11/24

code:

```
package CIE;
import java.util.*;
class Student
{
	public String usn;
	public String name;
	public int sem;

	public void inputStudentDetails()
	{
		Scanner s=new Scanner(System.in);
		usn=s.nextLine();
		name=s.nextLine();
		sem=s.nextInt();
	}
	public void displayStudentDetails()
	{
		System.out.println("Name:"+name);
		System.out.println("USN:"+usn);
		System.out.println("Semester:"+sem);
	}
}

public class Internals extends Student
{
	public int marks[] = new int[5];
	public void inputCIEmarks()
	{
		Scanner s=new Scanner(System.in);
		for(int i=0;i<5;i++)
		{
			marks[i]=s.nextInt();
		}
	}

}
```

```
package SEE;
import CIE.*;
import java.util.*;
public class Externals extends Internals
{
	public int seemarks[]=new int[5];
	public int finalMarks[]=new int[5];
	Scanner s=new Scanner(System.in);
	public void inputSEEmarks()
	{
		for(int i=0;i<5;i++)
			seemarks[i]=s.nextInt();
	}
	public void calculateFinalMarks()
	{
		for(int i=0;i<5;i++)
			finalMarks[i]=(marks[i])+(seemarks[i]/2);
	}
	public void displayFinalMarks()
	{
		for(int i=0;i<5;i++)
			System.out.println(finalMarks[i]);
	}
}
import SEE.*;
import CIE.*;
import java.util.*;
class MainProgram
{
	public static void main(String args[])
	{
		int n;
		Scanner s=new Scanner(System.in);
		System.out.println("Enter the Number of students");
		n=s.nextInt();
		Externals e[]=new Externals[n];
		for(int i=0;i<n;i++)
		{
			e[i]=new Externals();
			System.out.println("Enter Name,USN,Semester");
			e[i].inputStudentDetails();
			System.out.println("Enter the Internals marks of five Courses");
			e[i].inputCIEmarks();
			System.out.println("Enter the SEE marks of five Courses");
			e[i].inputSEEmarks();
			e[i].calculateFinalMarks();
			System.out.println("Student Details and Final Marks");
			e[i].displayStudentDetails();
			e[i].displayFinalMarks();
		}
	}
}
```

Output:

```
D:\1BM23CS172(java)>javac MainProgram.java

D:\1BM23CS172(java)>java MainProgram
Enter the Number of students
2
Enter Name,USN,Semester
shailesh
1BM23CS172
3
Enter the Internals marks of five Courses
44
50
45
42
43
Enter the SEE marks of five Courses
88
89
99
97
99
Student Details and Final Marks
Name:1BM23CS172
USN:shailesh
Semester:3
88
94
94
90
92
Enter Name,USN,Semester
Likith
1BM23CS170
3
Enter the Internals marks of five Courses
45
50
45
44
43
Enter the SEE marks of five Courses
88
89
99
90
98
Student Details and Final Marks
Name:1BM23CS170
USN:Likith
Semester:3
89
94
94
89
92
```

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

Algorithm:

```
(iii) write a program that demostrates 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 uses both father and son's age and throws an exception if son's age >= father's age. [lab program 7]

import java.util.*;
class wrongAage extends Exception
{
    Public WrongAge (String str)
    { super(str); }
}
class father
{
    Scanner s=new Scanner (System.in);
    int fatherAge;
    father()
    { try
      { system.out.println ("Enter the father's Age");
        fatherAge = s.nextInt();
        if(fatherAge <0) throw new
                WrongAge("Age cannot be negative");
      }
    catch(WrongAge e){ System.out. println(e); }
    }
    void display()
    {
      system. out. println ("father's age is"+ fatherAge);
    }
}
```

```
class son extends father
{
    Scanner s=new Scanner (system. in);
    int sonAge;
    son()
    { super();
      try
      { system.out.println("enter the son's age");
        sonAge = s.nextInt();
        if (sonAge<0)throw new
                WrongAge("Age cannot be negative");
        if(fatherAge <sonAge) throw new
                WrongAge("father's age cannot be
                        less than son's age");
      }
      catch(WrongAge e){system.out.println(e);}
    }
    void display()
    {
      system.out.println("son's age is"+sonAge);
    }
}
class program
{
    public static void main (String args[])
    {
      son. so=new son();
    }
}
```

Output

```
Enter the father's age
40
Enter the son's age
50
WrongAge: father's age cannot
be less than son's age
```

```
Enter the father's age
40
Enter the son's age
-12
Age cannot be negative
```

10/10

code:

```
import java.util.*;
class wrongAge extends Exception
{
	public wrongAge(String str){
super(str);
}
}

class father
{
	Scanner s=new Scanner(System.in);
	int fatherAge;
	father()
	{
		try
		{
			System.out.println("enter the father's age");
			fatherAge=s.nextInt();
			if(fatherAge<0) throw new wrongAge("Age cannot be negative");
		}

	catch(wrongAge e){System.out.println(e);}
	}
	void display()
	{
		System.out.println("father's age is "+fatherAge);
	}
}

class son extends father
{
	Scanner s=new Scanner(System.in);
	int sonAge;
	son()
	{
		super();
		try
		{
			System.out.println("enter the son's age");
			sonAge=s.nextInt();
			if(sonAge<0) throw new wrongAge("Age cannot be negative");
			if(fatherAge<sonAge)
			throw new wrongAge("father's age cannot be less than son's age");
		}
		catch(wrongAge e){System.out.println(e);}
	}
	void display()
	{
		System.out.println("son's age is "+sonAge);
	}
}
```

```
class Program
{
        public static void main(String args[])
        {
                son so=new son();
                System.out.println("Name: M S Shailesh");
                System.out.println("USN: 1BM23CS172");
        }
}
```

Output:

```
C:\Users\shail\OneDrive\Desktop>javac Program.java

C:\Users\shail\OneDrive\Desktop>java Program
enter the father's age
40
enter the son's age
50
wrongAge: father's age cannot be less than son's age
Name: M S Shailesh
USN: 1BM23CS172

C:\Users\shail\OneDrive\Desktop>java Program
enter the father's age
-12
wrongAge: Age cannot be negative
enter the son's age
-10
wrongAge: Age cannot be negative
Name: M S Shailesh
USN: 1BM23CS172
```

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.

Algorithm:

29

8) write a program which creates two threads, on thread displaysing "BMSCE" once every ten seconds and another displaying "CSE" once two seconds.

[Lab program 8]

```
Class BMSE extends Thread
{
    Public void run()
    {
        for (int i=0; i<5; i++)
        {
            system.out.println ("BMSCE");
            try{ Thread.steep(10000);}
            Catch (InterruptedException e) {}
        }
    }
}
Class CSE extends Thread.
{
    Public void run()
    {
        for (int i=0; i<5; i++)
        {
            System.out.println("CSE");
            try { Thread.steep(2000);}
            Catch (InterruptedException e) {}
        }
    }
}
Class Main program
{
    Public static void main (String args[])
    {
        BMSCE b=new BMSCE();
        CSE c=new CSE();
        b.start();
        c.start();
    }
}
```

//output
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
BMSCE
BMSCE
BMSCE
BMSCE

2/12/24

10/10

code:

```
class BMSCE extends Thread
{
	public void run()
	{
		for(int i=0;i<5;i++)
		{
			System.out.println("BMSCE");
			try{Thread.sleep(10000);}
			catch(InterruptedException e){}
		}
	}
}
class CSE extends Thread
{
	public void run()
	{
		for(int i=0;i<5;i++)
		{
			System.out.println("CSE");
			try{Thread.sleep(2000);}
			catch(InterruptedException e){}
		}
	}
}
class mainProgram
{
	public static void main(String args[])
	{
		BMSCE b=new BMSCE();
		CSE c=new CSE();
		b.start();
		c.start();
		try{Thread.sleep(50000);}
			catch(InterruptedException e){}
		System.out.println("Name: M S Shailesh");
		System.out.println("USN: 1BM23CS172");
	}
}
```

Output:

```
C:\Users\shail\OneDrive\Desktop>javac mainProgram.java

C:\Users\shail\OneDrive\Desktop>java mainProgram
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
BMSCE
BMSCE
BMSCE
Name: M S Shailesh
USN: 1BM23CS172
```

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.

Algorithm:

16) 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 Number Format Exception. If Num2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

[Lab program 9]

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(275,150);
    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();
    jfrm.add(err);
    jfrm.add(ajtf);
    jfrm.add(bjtf);
    jfrm.add(button);
    jfrm.add(alab);
    jfrm.add(blab);
    jfrm.add(anslab);
```

21

```
ActionListener l = new ActionListener()
{ public void actionPerformed(ActionEvent evt)
  { systemout.println("Action Event from a textfield");
  };
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[])
{ SwingUtilities.invokeLater(new Runnable()
  public void run(){ new SwingDemo(); }});
}
}
```

User Interface

Output 1:

Enter the divider and divident:

[56] [7]

[Calculate] A = 56 B = 7 Ans = 8

Output 2:

B should be NON zero

Enter the divider and divident:

[50] [0]

[Calculate]

Output 3:

Enter only Integers

Enter the divider and divident:

[the] [89]

[Calculate]

10/10 2/12/24

code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
SwingDemo(){
// create jframe container
	JFrame jfrm = new JFrame("Divider App");
	jfrm.setSize(275, 150);
	jfrm.setLayout(new FlowLayout());
// to terminate on close
	jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
	JLabel jlab = new JLabel("Enter the divider and divident:");
// add text field for both numbers
	JTextField ajtf = new JTextField(8);
	JTextField bjtf = new JTextField(8);
// calc button
	JButton button = new JButton("Calculate");
// labels
	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 bois
	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 text field"); }
};
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!"); }
}
});
// 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:

Program 10

Demonstrate Inter process Communication and deadlock

Algorithm:

17) Demostrate Inter process communication and deadlock. [Lab program 10]

```
class Q
{ int n
  boolean valueset = false;
  synchronized int get()
  {  while (!valueset)
     try { system.out.println ("consumer waiting");
           wait();
         }
     catch (InterruptedException e){ }
     system.out.println ("Got: "+n);
     valueset = false
     system.out.println ("Intimate producer\n");
     notify();
     return n;
  }
  synchronized void put (int n)
  { while (valueset)
    try {
      system.out.println ("Producer waiting\n");
      wait();
    }
    catch (Exception e) { }
    this.n=n;
    value set = true;
    System.out.println ("Put :"+n);
    System.out.println ("Intimate Consumer\n");
    notify();
  }
}
```

33

```
class producer implements Runnable
{ Q q;
  Producer (Q q)
  {this.q = q;
   new thread (This, "producer").start();
  }
  public void run()
  { int p=0;
    while (p<r)
    { q.put(p++);}
  }
}
class consumer implements Runnable
{ Q q;
  consumer (Q q)
  {this.q = q;
   new Thread (this, "consumer").start();
  }
  public void run()
  { int p=0;
    while (p<r)
    { int r= q.get();
      system.out.println ("consumed : "+r);
      p++;
    }
  }
}
class PCFixed
{ public static void main (String args[])
  { Q q=new Q();
    new producer (q);
    new Consumer(q);
    System.out.println ("Press Control + C to stop");
  }
}
```

IPC output:

Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate producer
Put: 1
Intimate consumer
consumed: 0
producer waiting
Got 1
Intimate producer
consumed: 1
Put: 2
Intimate Consumer
producer waiting
Got: 2
Intimate producer
consumed: 2
Put: 3
Intimate Consumer
Producer waiting
Got: 3
Intimate producer
Consumed: 3
Put: 4
Intimate consumer
Got: 4
Intimate Producer
consumed: 4

10/10

2/12/24

code:

```
class Q {
	int n;
	boolean valueSet = false;
	synchronized int get() {
		while(!valueSet)
		try {
			System.out.println("Consumer waiting\n");
			wait();
		} catch(InterruptedException e) {
	System.out.println("InterruptedExceptioncaught");
		}
	System.out.println("Got: " + n);
	valueSet = false;
	System.out.println("Intimate Producer\n");
	notify();
	return n;
}
synchronized void put(int n) {
while(valueSet)
try {
System.out.println("Producer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("Intimate Consumer\n");
notify();
}}
class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<5) {
q.put(i++);
}}}
class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
}
```

```
public void run() {
int i=0;
while(i<5) {
int r=q.get();
System.out.println("consumed:"+r);
i++;
}}}
class PCFixed {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}}
```

Output:

```
C:\Users\shail\OneDrive\Desktop>javac PCFixed.java

C:\Users\shail\OneDrive\Desktop>java PCFixed
Press Control-C to stop.
Name: M S Shailesh
USN: 1BM23CS172
Put: 0
Intimate Consumer

Producer waiting

Got: 0
Intimate Producer

Put: 1
Intimate Consumer

Producer waiting

consumed:0
Got: 1
Intimate Producer

consumed:1
Put: 2
Intimate Consumer

Producer waiting

Got: 2
```

```
Intimate Producer

consumed:2
Put: 3
Intimate Consumer

Producer waiting

Got: 3
Intimate Producer

consumed:3
Put: 4
Intimate Consumer

Got: 4
Intimate Producer

consumed:4
```

Algorithm:

```
class A
{
  synchronized void foo (B b)
  {  String name = Thread.currentThread.getName();
     System.out.println(name + " entered A.foo");
     try { Thread.sleep(1000); }
     catch(Exception e) { S.o.p("A Interrupted"); }
     System.out.println(name + " trying to call B.last());
     b.last();
  }
  void last()
  { System.out.println("Inside A.last");
  }
}
class B
{ synchronized void bar (A a)
  { String name = Thread.currentThread().getName();
    System.out.println(name + "entered B.bar");
    try { Thread.sleep(1000); }
    catch (Exception e) { S.o.p("B interrupted"); }
    System.out.println(name + " trying to call A.last
    a.last();
  }
  void last(){
  System.out.println("Inside A.last");
  }
}
```

```
class Deadlock implements Runnable
{ A a=new A();
  b b=new B();
  Deadlock()
  { Thread.currendThread().setName("MainThread");
    Thread t=new Thread(this, "Racing Thread");
    t.start();
    a.foo(b);
    System.out.println("Back in main thread");
  }
  public void run()
  { b.bar(a);
    System.out.println("Back in other thread");
  }
  public static void main(String args[])
  { new Deadlock();
  }
}
```

Output

```
MainThread entered A.foo
Racing Thread entered B.bar
MainThread trying to call B.last
Inside A.last
Back in main Thread
Racing Thread trying to call A.last()
Inside A.last
Back in other thread
```

10/10

code:

```
class A {
synchronized void foo(B b) {
String name =Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("A Interrupted");
}
System.out.println(name + " trying to call B.last()");
b.last();
}
void last() {
System.out.println("Inside A.last");
}
}

class B {
synchronized void bar(A a) {
String name =Thread.currentThread().getName();
System.out.println(name + " entered B.bar");
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("B Interrupted");
}
System.out.println(name + " trying to call A.last()");
a.last();
}
void last() {
System.out.println("Inside A.last");
}
}

class Deadlock implements Runnable
{
A a = new A();
B b = new B();
Deadlock() {
Thread.currentThread().setName("MainThread");
Thread t = new Thread(this,"RacingThread");
t.start();
a.foo(b); // get lock on a in this thread.
System.out.println("Back in main thread");
}
public void run() {
b.bar(a); // get lock on b in other thread.
System.out.println("Back in other thread");
}
public static void main(String args[]) {
new Deadlock();
}}
```

Output:

```
C:\Users\shail\OneDrive\Desktop>javac Deadlock.java

C:\Users\shail\OneDrive\Desktop>java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
RacingThread trying to call A.last()
Inside A.last
Back in other thread
Inside A.last
Back in main thread
Name:M S Shailesh
USN: 1BM23CS172
```