VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

V. KENNY PHILIP (1BM21CS232)

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
Oct 2022-Feb 2023

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**" carried out by **V. KENNY PHILIP (1BM21CS232),** 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 during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (22CS3PCOOJ)** work prescribed for the said degree.

Dr. NANDHINI VINEETH

Assistant Professor
Department of CSE
BMSCE, Bengaluru

DR. JYOTHI S NAYAK

Professor and Head
Department of CSE
BMSCE, Bengaluru

Index Sheet

	d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.	
6	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.	23
7	Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.	27
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.	30
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.	32
10	Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals have 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.	36
11	Demonstrate Inter process Communication and deadlock	40

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.

```
import java.util.Scanner;
import java.lang.Math;

class Quadratic {
public static void main (String args[]){
Scanner s =new Scanner (System.in);

System.out.println("Enter the values of a, b and c");
double a= s.nextInt();
double b= s.nextInt();
double c= s.nextInt();

//Discriminant is D
double D=b*b-4*a*c;
double root1, root2;
if(D>0){
System.out.println("Roots are real and Unique");
root1= -b+Math.sqrt(D)/(2*a);
root2= -b-Math.sqrt(D)/(2*a);
System.out.println("Root1= "+root1+" AND "+"Root2= "+root2);
}
else if(D==0) {
System.out.println("Roots are real and equal");
root1=root2=-b/(2*a);
System.out.println("Root1=Root2= "+root1);
}
```

```
else {
System.out.println("There are no real solutions");
double realpart=-b/(2*a);
double imagpart=Math.sqrt(-D)/(2*a);
System.out.println("Root1= "+realpart+" + "+imagpart+"i"+" AND "+"Root2= "+realpart+" - "+imagpart+"i");
}
}
}
```

OUTPUT:

```
C:\Users\BMSCECSE\Desktop>java Quadratic
Enter the values of a, b and c
2
-11
14
Roots are real and Unique
Root1= 11.75 AND Root2= 10.25

C:\Users\BMSCECSE\Desktop>java Quadratic
Enter the values of a, b and c
1
7
3
Roots are real and Unique
Root1= -3.9586187348508903 AND Root2= -10.04138126514911

C:\Users\BMSCECSE\Desktop>java Quadratic
Enter the values of a, b and c
1
3
7
There are no real solutions
Root1= -1.5 + 2.1794494717703337i AND Root2= -1.5 - 2.1794494717703337i

C:\Users\BMSCECSE\Desktop>java Quadratic
Enter the values of a, b and c
1
-10
25
Roots are real and equal
```

Program 2:

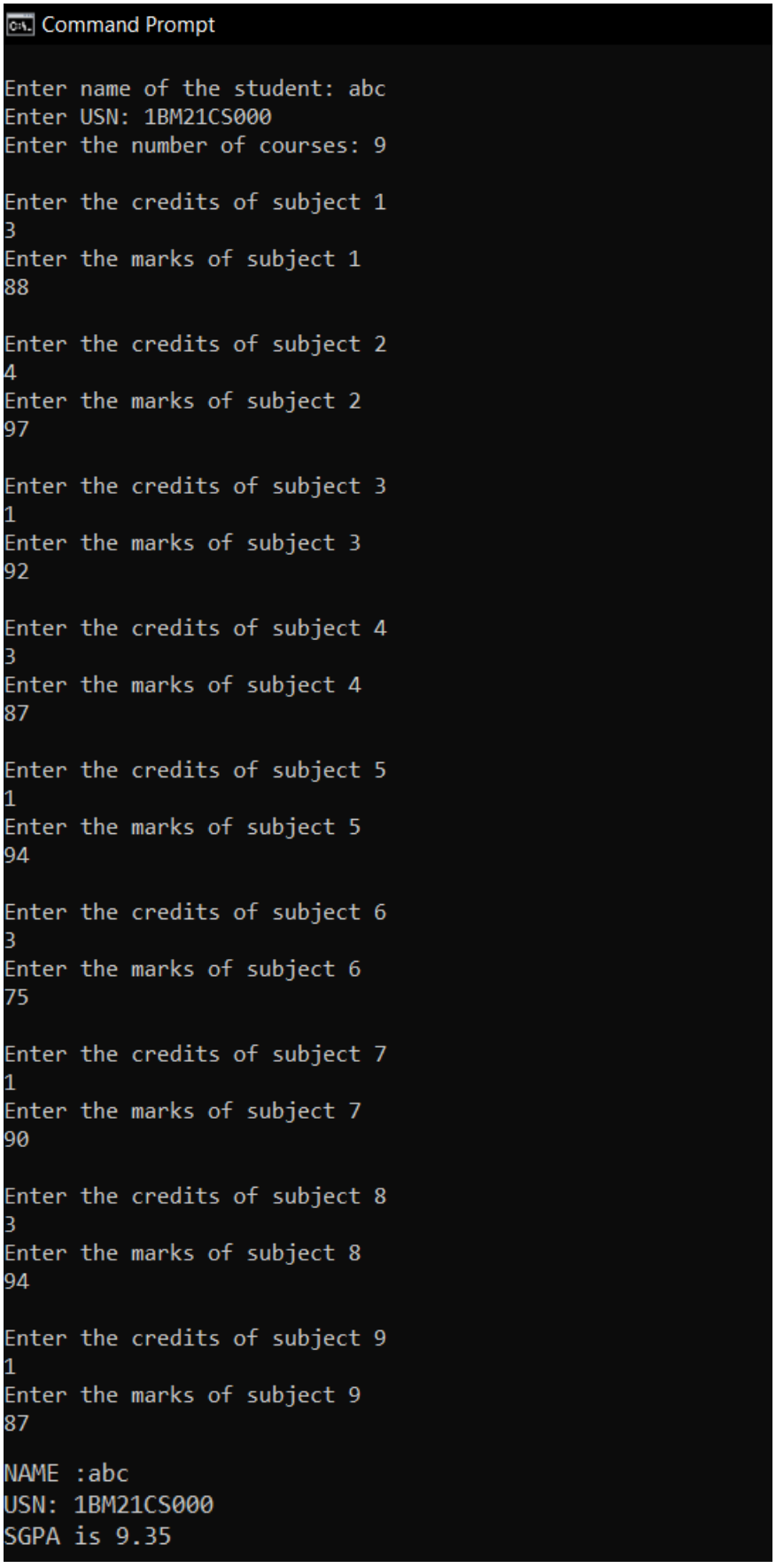
Develop a Java program to create a class Student with members usn, name, an array credits and an array mark. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;
class Student {
	String name,USN;
	int marks[] = new int[10];
	int credits[] = new int[10];
	double SGPA=0;
	int n;
	Scanner s=new Scanner(System.in);

	void Input(){
		System.out.println();
		System.out.print("Enter name of the student: ");
		name=s.nextLine();
		System.out.print("Enter USN: ");
		USN=s.nextLine();
		System.out.print("Enter the number of courses: ");
		n=s.nextInt();

		for(int i=0;i<n;i++){
			System.out.println();
			System.out.println("Enter the credits of subject "+(i+1));
			credits[i]=s.nextInt();
			System.out.println("Enter the marks of subject "+(i+1));
			marks[i]=s.nextInt();
		}
	}
```

```
void Display(){
        System.out.println();
        System.out.println("NAME :"+name);
        System.out.println("USN: "+USN);
        System.out.println("SGPA is "+SGPA);
}

void CalcSGPA(){
        double grade=0, totalcred=0;
        int i;

for(i=0;i<n;i++){
        totalcred=totalcred+credits[i];
        }

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

        if(marks[i]>=90)
        {
                grade+=credits[i]*10;
        }
        else if(marks[i]>=80)
        {
                grade+=credits[i]*9;
        }
        else if(marks[i]>=70)
        {
                grade+=credits[i]*8;
        }
        else if(marks[i]>=60)
```

```
                {
                        grade+=credits[i]*7;
                }
                else if(marks[i]>=50)
                {
                        grade+=credits[i]*6;
                }
                else if(marks[i]>=40)
                {
                        grade+=credits[i]*5;
                }
                else{
                        grade=0;
                }
        }
        SGPA=grade/totalcred;
        }
}

class SGPA{
        public static void main(String args[]){

                Student st= new Student();

                st.Input();
                st.CalcSGPA();
                st.Display();
        }
}
```

OUTPUT:


Command Prompt
Enter name of the student: abc
Enter USN: 1BM21CS000
Enter the number of courses: 9
Enter the credits of subject 1
3
Enter the marks of subject 1
88
Enter the credits of subject 2
4
Enter the marks of subject 2
97
Enter the credits of subject 3
1
Enter the marks of subject 3
92
Enter the credits of subject 4
3
Enter the marks of subject 4
87
Enter the credits of subject 5
1
Enter the marks of subject 5
94
Enter the credits of subject 6
3
Enter the marks of subject 6
75
Enter the credits of subject 7
1
Enter the marks of subject 7
90
Enter the credits of subject 8
3
Enter the marks of subject 8
94
Enter the credits of subject 9
1
Enter the marks of subject 9
87
NAME :abc
USN: 1BM21CS000
SGPA is 9.35

Program 3:

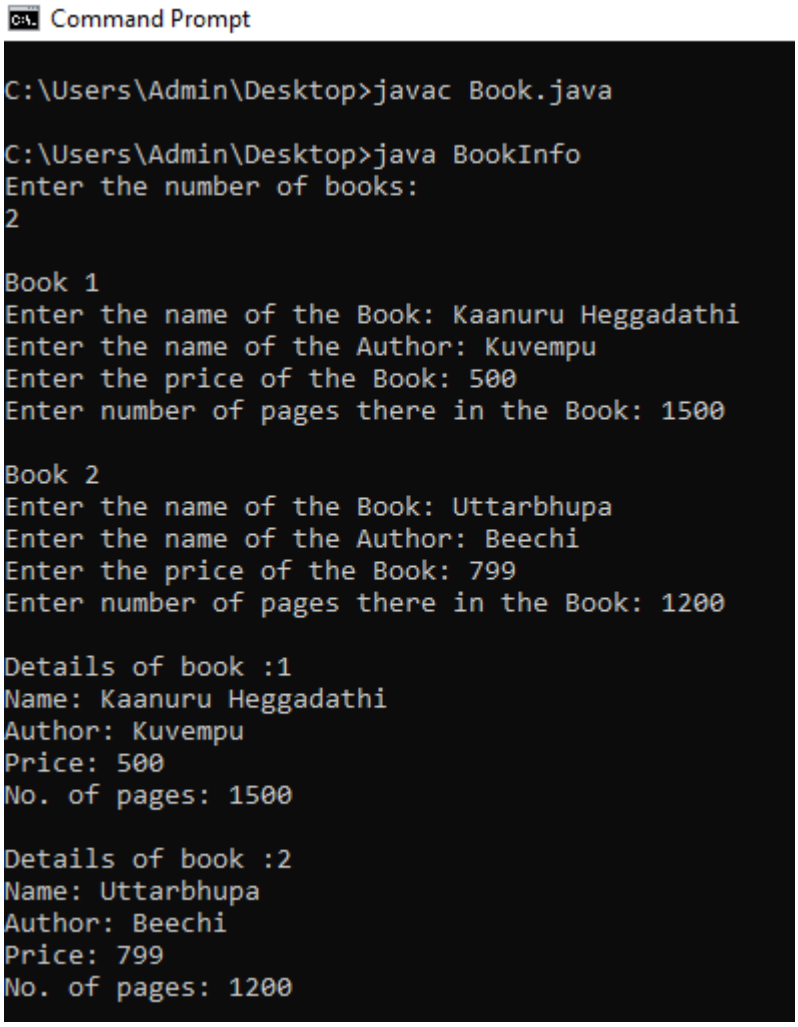
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.*;
class Book{
	String name,author;
	int price,num_pages;

	Book(){
		name="";
		author="";
		price=0;
		num_pages=0;
	}

	void setnget(String n, String a, int p, int nop){
		name=n;
		author=a;
		price=p;
		num_pages=nop;
	}

	public String toString(){
		String s="";
		s+="Name: "+name+"\nAuthor: "+author+"\nPrice: "+price+"\nNo. of pages: "+num_pages;
		return s;
	}
}
```

```
class BookInfo{
	public static void main(String args[]){
		Scanner sc=new Scanner(System.in);

		System.out.println("Enter the number of books: ");
		int num=sc.nextInt();

		Book b[]=new Book[num];

		for(int i=0;i<num;i++){
	b[i]=new Book();
	}

		for(int i=0;i<num;i++)
	{
			System.out.println();
	System.out.println("Book "+(i+1));
	sc.nextLine();
	System.out.print("Enter the name of the Book: ");
	String n=sc.nextLine();
	System.out.print("Enter the name of the Author: ");
	String a=sc.nextLine();
	System.out.print("Enter the price of the Book: ");
	int p=sc.nextInt();
	System.out.print("Enter number of pages there in the Book: ");
	int nop=sc.nextInt();

	b[i].setnget(n,a,p,nop);
	}
```

```
    for(int i=0;i<num;i++)
    {
                        System.out.println();
       System.out.println("Details of book :"+(i+1));
                System.out.println(b[i]);
    }
      }
}
```

OUTPUT:



Program 4:

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

```
import java.util.*;
abstract class Shape{
	int a,b;
	Shape(int x, int y){
		a=x;
		b=y;
	}
	abstract double printArea();
}
class Rectangle extends Shape{
	Rectangle(int length,int breadth){
		super(length,breadth);
	}
	double printArea(){
		return a*b;
	}
}
class Triangle extends Shape{
	Triangle(int length, int height){
		super(length,height);
	}
	double printArea(){
		return 0.5*a*b;
	}
}
```

```
class Circle extends Shape{
        Circle(int r){
                super(r,r);
        }
        double printArea(){
                return Math.PI*a*b;
        }
}

class AREA{
        public static void main(String args[]){

                Rectangle R=new Rectangle(10,20);
                Triangle T=new Triangle(15,30);
                Circle C=new Circle(5);

                Shape s;
                s=R;
                System.out.println("Area of the Rectangle : " +s.printArea());
                s=T;
                System.out.println("Area of the Triangle : " +s.printArea());

                s=C;
                System.out.println("Area of the Circle : " +s.printArea());
        }
}
```

OUTPUT:

```
C:\Users\Admin>set path="C:\Program Files\Java\jdk-19\bin"

C:\Users\Admin>cd "Desktop"

C:\Users\Admin\Desktop>javac abst.java

C:\Users\Admin\Desktop>java abst
area of rectangle:200.0
area of triangle:10.0
area of circle:28.259999999999998

C:\Users\Admin\Desktop>
```

Program 5:

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

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

```
import java.util.Scanner;


class Account {
    String customerName;
    int accountNumber;
    String typeOfAccount;
    double balance;

    Account(String customerName, int accountNumber, String typeOfAccount) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.typeOfAccount = typeOfAccount;
    }

    void details() {
        System.out.println();
        System.out.println("Name: " + customerName);
```

```
        System.out.println("Acc no.: " + accountNumber);
        System.out.println("Type: " + typeOfAccount);
        System.out.println("Balance: " + balance);
    }
}

class SavAcc extends Account {
    Scanner sc = new Scanner(System.in);

    SavAcc(String customerName, int accountNumber, String typeOfAccount) {
        super(customerName, accountNumber, typeOfAccount);
    }

    void details() {
        super.details();
        System.out.println("Minimum Balance: No Minimum Balance for Savings Account");
    }

    void acceptDeposit() {
        System.out.println();
        System.out.println("Enter amount to be deposited");
        double deposit = sc.nextDouble();
        balance += deposit;
        System.out.println("\nTransaction Successfull!!\n");
        System.out.println("Updated Balance: " + balance);
    }

    void permitWithdrawal() {
        System.out.println();
        System.out.println("Withdrawal amount");
```

```
        double withdraw = sc.nextDouble();
        if (balance == 0) {
            System.out.println("\nTransaction Failed");
            System.out.println("Zero Balance");
            return;
        }
        balance -= withdraw;
        System.out.println("\nTransaction Successfull!!\n");
        System.out.println("Updated Balance: " + balance);
    }

    void interest() {
        System.out.println("Months");
        double month = sc.nextInt();
        month /= 3;
        balance = balance + (balance * 0.10 * (month / 4));
        System.out.println("Updated Balance after depositing interest: " + balance);
    }
}

class CurAcc extends Account {
    double minBalance = 2000;
    Scanner sc = new Scanner(System.in);
    CurAcc(String customerName, int accountNumber, String typeOfAccount) {
        super(customerName, accountNumber, typeOfAccount);
    }

    void details() {
        super.details();
        System.out.println("Minimum Balance: " + minBalance);
```

```
}

void acceptDeposit() {
    System.out.println();
    System.out.println("Enter amount to be deposited");
    double deposit = sc.nextDouble();
    balance += deposit;
    System.out.println("\nTransaction Successfull!!\n");
    System.out.println("Updated Balance: " + balance);
}

void permitWithdrawal() {
    System.out.println();
    System.out.println("Withdrawal amount");
    double withdraw = sc.nextDouble();
    if (balance == 0) {
        System.out.println("\nTransaction Failed");
        System.out.println("Zero Balance");
        return;
    }
    balance -= withdraw;
    if (balance < minBalance) {
        System.out.println();
        System.out.println("oppps!!  balance is less than minimum balance");
        System.out.println("You have to pay penalty of Rs " + 1000);
        balance -= 1000;
        System.out.println("Updated Balance after deducting penalty: " + balance);
        return;
    }
    System.out.println("\nTransaction Successfull!!\n");
```

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

class Program5 {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int type;

        System.out.print("Enter account holder name: ");
        String name = sc.nextLine();

        System.out.print("Enter SavingsAccount number: ");
        int AccNo1 = sc.nextInt();
        SavAcc sav = new SavAcc(name, AccNo1, "SavingsAccount");// Creating
SavingsAccount object

        System.out.print("Enter CurrentAccount number: ");
        int AccNo2 = sc.nextInt();
        CurAcc curr = new CurAcc(name, AccNo2, "CurrentAccount");// Creating
CuurentAccount object

        // Menu
        while (true) {
          System.out.println("\n1.AccountDetails\n2.Deposit\n3.WithDraw\n4.Interest\n5.Exit");
            System.out.print("Enter your Choice: ");
            int ch = sc.nextInt();
            switch (ch) {
                case 1:
                    System.out.println("\nAccount Type");
                    System.out.println("1.Savings Acc \n2.Current Acc");
```

```
        type = sc.nextInt();
        if (type == 1) {
            sav.details();
        } else if (type == 2) {
            curr.details();
        }
        break;
    case 2:
        System.out.println("\nAccount Type");
        System.out.println("1.Savings Acc \n2.Current Acc");
        type = sc.nextInt();
        if (type == 1) {
            sav.acceptDeposit();
        } else if (type == 2) {
            curr.acceptDeposit();
        }
        break;
    case 3:
        System.out.println("\nAccount Type");
        System.out.println("1.Savings Acc \n2.Current Acc");
        type = sc.nextInt();
        if (type == 1) {
            sav.permitWithdrawal();
        } else if (type == 2) {
            curr.permitWithdrawal();
        }
        break;
    case 4:
        System.out.println("\nAccount Type");
        System.out.println("1.Savings Acc \n2.Current Acc");
```

```
                type = sc.nextInt();
                if (type == 1) {
                    sav.interest();
                } else if (type == 2) {
                    System.out.println("\nSorry CurrentAccount don't have interst facility");
                }
                break;
            case 5:
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice");
        }
    }
  }
}
```

OUTPUT:

```
Enter account holder name: abc
Enter SavingsAccount number: 123
Enter CurrentAccount number: 456

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 2

Account Type
1.Savings Acc
2.Current Acc
1

Enter amount to be deposited
50000

Transaction Successfull!!

Updated Balance: 50000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 2

Account Type
1.Savings Acc
2.Current Acc
2

Enter amount to be deposited
100000

Transaction Successfull!!

Updated Balance: 100000.0
```

```
1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 3

Account Type
1.Savings Acc
2.Current Acc
1

Withdrawal amount
25000

Transaction Successfull!!

Updated Balance: 25000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 3

Account Type
1.Savings Acc
2.Current Acc
2

Withdrawal amount
100000

oppps!!  balance is less than minimum balance
You have to pay penalty of Rs 1000
Updated Balance after deducting penalty: -1000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 3
```

```
Account Type
1.Savings Acc
2.Current Acc
2

Withdrawal amount
25000

oppps!!  balance is less than minimum balance
You have to pay penalty of Rs 1000
Updated Balance after deducting penalty: -27000.0

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 1

Account Type
1.Savings Acc
2.Current Acc
1

Name: abc
Acc no.: 123
Type: SavingsAccount
Balance: 25000.0
Minimum Balance: No Minimum Balance for Savings Account

1.AccountDetails
2.Deposit
3.WithDraw
4.Interest
5.Exit
Enter your Choice: 1

Account Type
1.Savings Acc
2.Current Acc
2

Name: abc
Acc no.: 456
Type: CurrentAccount
Balance: -27000.0
Minimum Balance: 2000.0
```

Program 6:

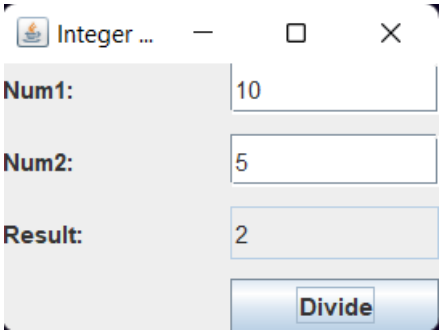
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

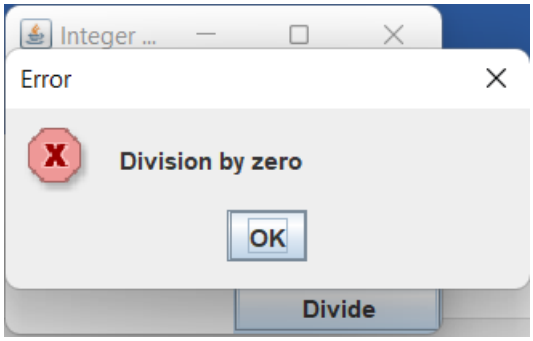
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

 class IntegerDivisionUI {
    private JFrame frame;
    private JTextField num1Field;
    private JTextField num2Field;
    private JTextField resultField;
    private JButton divideButton;

    public IntegerDivisionUI() {
        initUI();
    }

    private void initUI() {
        frame = new JFrame("Integer Division");
        frame.setLayout(new GridLayout(4, 2, 10, 10));
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel num1Label = new JLabel("Num1:");
        JLabel num2Label = new JLabel("Num2:");
        JLabel resultLabel = new JLabel("Result:");
```

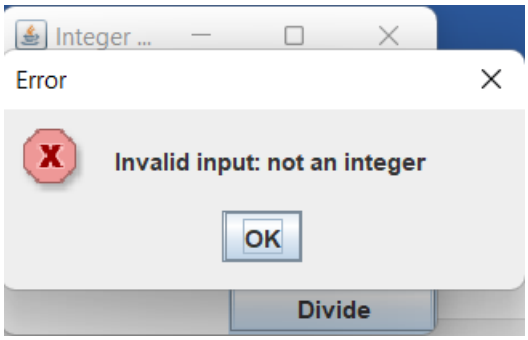
```
        num1Field = new JTextField(10);
        num2Field = new JTextField(10);
        resultField = new JTextField(10);
        resultField.setEditable(false);

        divideButton = new JButton("Divide");
        divideButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());

                    if (num2 == 0) {
                        throw new ArithmeticException("Division by zero");
                    }

                    int result = num1 / num2;
                    resultField.setText(String.valueOf(result));
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(frame, "Invalid input: not an integer", "Error",
JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(frame, ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
                }
            }
        });

        frame.add(num1Label);
```

```
        frame.add(num1Field);
        frame.add(num2Label);
        frame.add(num2Field);
        frame.add(resultLabel);
        frame.add(resultField);
        frame.add(new JLabel());
        frame.add(divideButton);

        frame.pack();
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new IntegerDivisionUI();
            }
        });
    }
}
```

OUTPUT:







Experiment 7:

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is >=father's age.

```
import java.util.*;
```

```
class Wrongage extends Exception
{    int detail;
  Wrongage(int d)
  { detail=d;
}       public String toString()
   {  return "Entered Wrong age is ["+detail+"]";
    }
 }
class Father
 {     int f;
   Scanner in=new Scanner(System.in);
   Father()
   {
     System.out.println("Enter father age ");       f=in.nextInt();
    }
    void checkage() throws Wrongage
   {
if(f<0)
      {
         throw new Wrongage(f);
       }
       System.out.println("Father age positive");
   }
}  class Son extends Father
{     int
s;
  Scanner in=new Scanner(System.in);
   Son()
   {        super();
      System.out.println("Enter son age "); s=in.nextInt();
    }
```

```
   void checkages() throws Wrongage
  {
super.checkage();        if(s<0)
     {
        throw new Wrongage(f);
      }
      System.out.println("Son age positive");
   }
   void checkage() throws Wrongage
   {         if(s>=f)
      {
          throw new Wrongage(s);
      }
      System.out.println("Father-Son age correct");
    }
  }
class Newdemo
{     public static void main(String args[])
  {      int f,s;
     Father fath=new Father();       Father r;
r=fath;       try
     {
        r.checkage();        }
       catch(Wrongage e)
     {
        System.out.println("Father age wrong"+e);
     }
     Son sn=new Son();
      r=sn;        try
      {
```

```
sn.checkages();          r.checkage();
}         catch(Wrongage e)
    {
        System.out.println("Son age wrong"+e);
     }
  }
}
```

OUTPUT:

```
Enter father age
-20
Father age wrongEntered Wrong age is [-20]
Enter father age
25
Enter son age
30
Father age positive
Son age positive
Son age wrongEntered Wrong age is [30]
```

```
Enter father age
40
Father age positive
Enter father age
45
Enter son age
30
Father age positive
Son age positive
Father-Son age correct
```

```
Enter father age
12
Father age positive
Enter father age
12
Enter son age
12
Father age positive
Son age positive
Son age wrongEntered Wrong age is [12]
```

```
Enter father age
-12
Father age wrongEntered Wrong age is [-12]
Enter father age
-89
Enter son age
-56
Son age wrongEntered Wrong age is [-89]
```

Program 8:

Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
class BMSCE extends Thread {
   public void run() {
      try {
         for (int i = 5; i > 0; i--) {
            System.out.println("BMS College Of Engineering");
            Thread.sleep(10000);
         }
      } catch (InterruptedException e) {
         System.out.println("Interrupted");
      }
   }
}

class CSE extends Thread {
   public void run() {
      try {
         for (int i = 5; i > 0; i--) {
            System.out.println("CSE");
            Thread.sleep(2000);
         }
      } catch (InterruptedException e) {
         System.out.println("Interrupted");
      }
   }
}
```

```
class ThreadDemo {
    public static void main(String args[]) {
        new BMSCE().start();
        new CSE().start();
    }
}
```

OUTPUT:

```
BMS College Of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College Of Engineering
BMS College Of Engineering
BMS College Of Engineering
BMS College Of Engineering
```

Program 9:

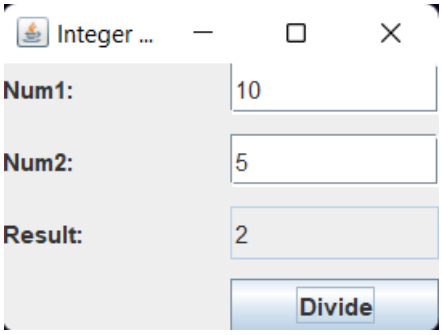
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

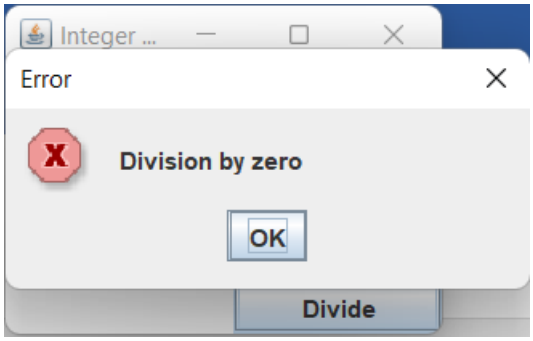
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

 class IntegerDivisionUI {
    private JFrame frame;
    private JTextField num1Field;
    private JTextField num2Field;
    private JTextField resultField;
    private JButton divideButton;

    public IntegerDivisionUI() {
        initUI();
    }

    private void initUI() {
        frame = new JFrame("Integer Division");
        frame.setLayout(new GridLayout(4, 2, 10, 10));
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel num1Label = new JLabel("Num1:");
        JLabel num2Label = new JLabel("Num2:");
        JLabel resultLabel = new JLabel("Result:");
```

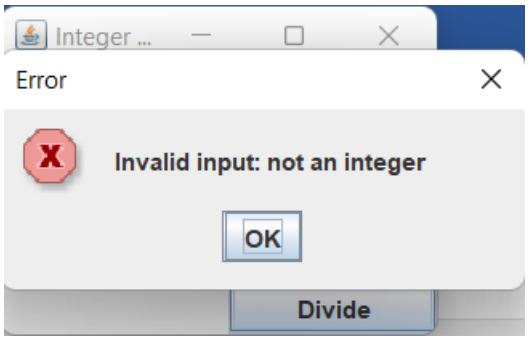
```
        num1Field = new JTextField(10);
        num2Field = new JTextField(10);
        resultField = new JTextField(10);
        resultField.setEditable(false);

        divideButton = new JButton("Divide");
        divideButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());

                    if (num2 == 0) {
                        throw new ArithmeticException("Division by zero");
                    }

                    int result = num1 / num2;
                    resultField.setText(String.valueOf(result));
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(frame, "Invalid input: not an integer", "Error",
JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(frame, ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
                }
            }
        });

        frame.add(num1Label);
```

```
        frame.add(num1Field);
        frame.add(num2Label);
        frame.add(num2Field);
        frame.add(resultLabel);
        frame.add(resultField);
        frame.add(new JLabel());
        frame.add(divideButton);

        frame.pack();
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                new IntegerDivisionUI();
            }
        });
    }
}
```

OUTPUT:







Program 10:

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current

semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
package cie;
public class Internals {
	public int internal[]=new int[5];
}
package cie;
public class Student {
	public String name;
	public int usn;
	public int sem;
}
package see;
import cie.Internals;
public class External extends Internals {
	public int external[]=new int[5];
}
import java.util.Scanner;
import cie.Student;
import see.External;
public class Marks {
	public static void main(String[] args) {
		int n;
		Scanner sc=new Scanner(System.in);
		System.out.println("enter number of students");
		n=sc.nextInt();
		External student[]=new External[n];
		Student details[]=new Student[n];
		int final_marks[][]=new int[n][5];
		for(int i=0;i<n;i++)
```

```
{
        student[i]=new External();
        details[i]=new Student();
        System.out.println("Enter Student  usn and sem respectively");
        details[i].usn=sc.nextInt();
        details[i].sem=sc.nextInt();
        System.out.println("Enter Internal marks of 5 subject in respective
order");
        for(int j=0;j<5;j++)
        {

                student[i].internal[j]=sc.nextInt();
        }
        System.out.println("Enter external marks of 5 subject in respective
order");
        for(int k=0;k<5;k++)
        {

                student[i].external[k]=sc.nextInt();
        }
}
for(int i=0;i<n;i++)
{
        for(int j=0;j<5;j++)
        final_marks[i][j]=student[i].internal[j]+(int)(student[i].external[j]/2);
}
for(int i=0;i<n;i++)
{
        //System.out.println("Name: "+details[i].name);
        System.out.println("USN: "+details[i].usn);
        System.out.println("Sem: "+details[i].sem);
```

```
                    System.out.println("Marks of the student is");
                    for(int j=0;j<5;j++)
                    {
                        System.out.println(final_marks[i][j]);
                    }
                }
        }
}
```

OUTPUT:

```
enter number of students
2
Enter Student usn and sem respectively
220 3
Enter Internal marks of 5 subject in respective order
34
33
32
28
40
Enter external marks of 5 subject in respective order
45
67
87
98
78
Enter Student usn and sem respectively
221
3
```

```
Enter external marks of 5 subject in respective order
45
67
87
98
78
Enter Student usn and sem respectively
221
3
Enter Internal marks of 5 subject in respective order
30
28
34
40
43
Enter external marks of 5 subject in respective order
89
87
76
65
54
USN: 220
Sem: 3
Marks of the student is
56
66
75
77
79
USN: 221
Sem: 3
Marks of the student is
74
71
72
72
70
```

Program 11:

Demonstrate Inter process Communication and deadlock.

```
class printer{
	String str;
	printer()
	{
		str="";
	}
	synchronized void print(String str)
	{
		System.out.print("["+str);
		try {
			Thread.sleep(1000);
		}catch(InterruptedException e)
		{
			System.out.println("Error occured");
		}
		try {
			System.out.println("]");
			Thread.sleep(1000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}

	}
}
class SampleThread implements Runnable
{
```

```
	String msg;
	printer pt;
	Thread t;
	public SampleThread(printer pr,String message)
	{
		pt=pr;
		msg=message;
		t=new Thread(this);
		t.start();
	}

	@Override
	public void run() {
		// TODO Auto-generated method stub
		pt.print(msg);
	}

}
public class InterThread {
	public static void main(String[] args) {
		printer pt=new printer();
		SampleThread s1=new SampleThread(pt,"HELLO");
		SampleThread s2=new SampleThread(pt,"CSE");
		SampleThread s3=new SampleThread(pt,"WORLD");
		SampleThread s4=new SampleThread(pt,"BMS");

		try {
			s1.t.join();
			s2.t.join();
			s3.t.join();
```

```
                s4.t.join();
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
}
```

OUTPUT:

```
[HELLO]
[BMS]
[WORLD]
[CSE]
```