

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT
on**

**Object Oriented Java Programming
(23CS3PCOOJ)**

Submitted by

Hitesh Sharma(**1BM23CS114**)

in partial fulfilment 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 **HITESH SHARMA (1BM23CS114)**, who is Bonafide student of **B.M.S. College of Engineering**. It is in partial fulfilment 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.

Lab faculty in charge Name 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	1-10-2024	Quadratic Equation	4-6
2	8-10-2024	Calculating SGPA	6-10
3	15-10-24	Book details using to String ()	11-15
4	22-10-24	Shapes and area using Abstract class	16-18
5	29-10-24	Bank class using inheritance	19-23
6	12-11-24	Marks card using packages	24-30
7	19-11-24	Area and perimeter of shapes using interfaces	31-34
8	26-11-24	Father and son's age using exception handling	35-37
9	3-12-24	Display college name S department using threads	38-41
10	3-12-24	Dividing two integers, using exception handling	42-46



GitHub Link :

https://github.com/Hitesh6747/OOJ_114

Program 1

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

Algorithm:

```
import java.util.Scanner;
public class Quadratic {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the value of a");
        double a = sc.nextDouble();
        System.out.print("Enter the value of b");
        double b = sc.nextDouble();
        System.out.print("Enter the value of c");
        double c = sc.nextDouble();
        double d = (b*b) - (4*a*c);
        if (d > 0) {
            double root1 = (-b + Math.sqrt(d)) / (2*a);
            double root2 = (-b - Math.sqrt(d)) / (2*a);
            System.out.println("root1 = " + root1 + " root2 = " + root2);
        } else if (d == 0) {
            double root1 = -b / (2*a);
            double root2 = -b / (2*a);
            System.out.println("Root1 and Root2 = " + root1);
        } else {
            System.out.println("No real roots");
        }
    }
}
```

prameethabai.cs@bmsce.ac.in

Code:

```
import java.util.Scanner;
public class quad {
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the coefficients:");
        int a=s.nextInt();
        int b=s.nextInt();
        int c=s.nextInt();
        int d=b*b-4*a*c;
        if(d==0){
            System.out.println("Roots are equal");
            System.out.println("Roots are:");
            System.out.println(-b/2*a);
        }
        else if(d>0){
            System.out.println("Roots are unique");
            System.out.println((-b+Math.sqrt(d))/(2*a));
            System.out.println((-b-Math.sqrt(d))/(2*a));
        }
        else{
            System.out.println("No real roots");
        }
        System.out.println("Hitesh Sharma");
        System.out.println("1BM23CS114");
    }
}
```

OUTPUT

```
Enter the coefficients:  
1  
2  
1  
Roots are equal  
Roots are:  
-1  
HITESH SHARMA  
1BM23CS114  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

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:

Q4 Programm do calculate SGPA OF A STUDENT; } 1. ans

```

import java.util.*;
class Student {
    int grade;
    int sum = 0;
    int csum = 0;
    int credits[] = new int[8];
    int marks[] = new int[8];
    String name, USN;
    float sgpa;
}

void read () {
    Scanner sc = new Scanner (System.in);
    System.out.print ("Enter the name of the student : ");
    name = sc.nextLine();
    System.out.print ("Enter the USN of the student : ");
    USN = sc.nextLine();
    System.out.print ("Enter the marks of the subjects : ");
    for (int i=0; i<8; i++) {
        marks[i] = sc.nextInt();
    }
    System.out.print ("Enter the credits of the subjects : ");
    for (int i=0; i<8; i++) {
        credits[i] = sc.nextInt();
    }
}

void calculate () {
    for (int i=0; i<8; i++) {
        if (marks[i] > 100) {
            System.out.println ("Invalid Marks");
            break;
        }
    }
}

```

```

    else if (marks[i] >= 70)
        grade = 10;
    else if (marks[i] >= 60 && marks[i] < 70)
        grade = 9;
    else if (marks[i] >= 50 && marks[i] < 60)
        grade = 8;
    else if (marks[i] >= 40 && marks[i] < 50)
        grade = 7;
    else if (marks[i] >= 30 && marks[i] < 40)
        grade = 6;
    else if (marks[i] >= 20 && marks[i] < 30)
        grade = 5;
    else if (marks[i] >= 10 && marks[i] < 20)
        grade = 4;
    else if (marks[i] < 10)
        grade = 3;

    if (grade < 4)
        System.out.println("Your are failed!");
        break;
    else
        sum += grade * credits[i];
        csum += credits[i];
}

sgpa = (float) sum / csum;

```

```

void display()
{
    System.out.println("Name of the Student:" + name);
    System.out.println("USN of the Student:" + USN);
    System.out.println("Marks of the Subjects:");
    for (int i = 0; i < 8; i++)
        System.out.print(marks[i] + " ");
    System.out.println("Credits of the Subjects:");
    for (int i = 0; i < 8; i++)
        System.out.print(credits[i] + " ");
    System.out.println("In The SGPA of the student:" + sgpa);
}

```

```

public static void main(String args[])
{
    Student obj = new Student();
    obj.read();
    obj.calculate();
    obj.display();
}

```

Late Submission

CODE

```

import java.util.*;
class Student
{
    int grade,sum=0,csum=0;
    int credits[]={}; //new int[8];
    int marks[]={}; //new int[8];
    String name,USN;
    float sgpa;

    void read()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the name of the Student : ");
        name=sc.nextLine();
        System.out.print("\nEnter the USN of the Student : ");
        USN=sc.next();
        System.out.print("\nEnter the marks of the subjects : ");
        for(int i=0;i<8;i++)

```

```

        marks[i]=sc.nextInt();
        System.out.print("\nEnter the credits of the subjects : ");
        for(int i=0;i<8;i++)
            credits[i]=sc.nextInt();
    }

void display()
{
    System.out.println("\nName of the Student : "+name);
    System.out.println("\nUSN of the Student : "+USN);
    System.out.println("\nMarks of the subjects : ");
    for(int i=0;i<8;i++)
        System.out.print(marks[i]+" ");
    System.out.println("\nCredits of the subjects : ");
    for(int i=0;i<8;i++)
        System.out.print(credits[i]+" ");
    System.out.println("\nThe SGPA of the student : "+sgpa);
}

void calculate()
{
    for(int i=0;i<8;i++)
    {
        if(marks[i]>100)
        {
            System.out.println("Invalid Marks");
            break;
        }
        else if(marks[i]>=90)
            grade=10;
        else if(marks[i]>=80 && marks[i]<90)
            grade=9;
        else if(marks[i]>=70 && marks[i]<80)
            grade=8;
        else if(marks[i]>=60 && marks[i]<70)
            grade=7;
        else if(marks[i]>=50 && marks[i]<60)
            grade=6;
        else if(marks[i]>=40 && marks[i]<50)
            grade=5;
        else
        {
            System.out.println("You are failed !!");
            break;
        }
        sum+=grade*credits[i];
        csum+=credits[i];
    }
    sgpa=(float)sum/csum;
}

public static void main(String args[])
{

```

```

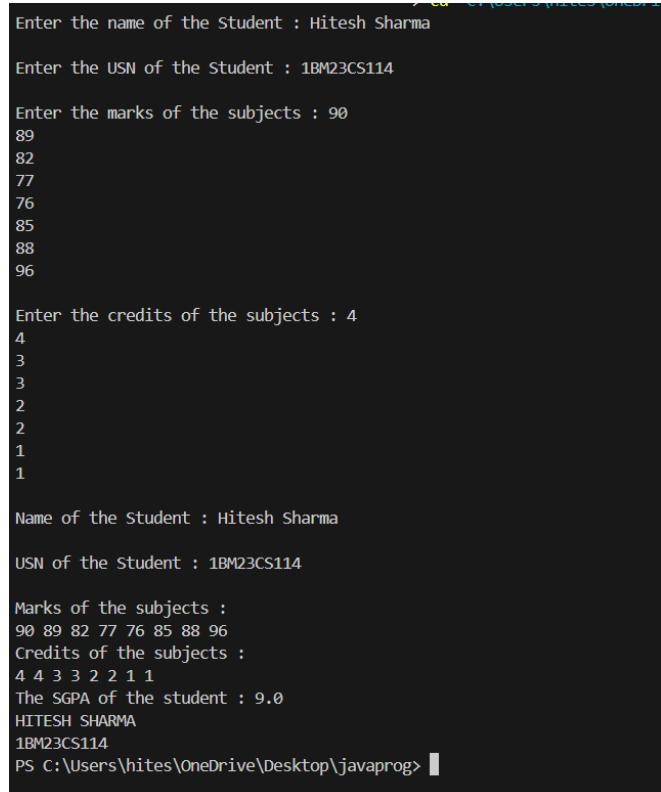
Student obj=new Student();
obj.read();
obj.calculate();
obj.display();

System.out.println("HITESH SHARMA");
System.out.println("1BM23CS114");
}

}

```

OUTPUT:



The screenshot shows a terminal window with the following text output:

```

Enter the name of the Student : Hitesh Sharma
Enter the USN of the Student : 1BM23CS114
Enter the marks of the subjects : 90
89
82
77
76
85
88
96
Enter the credits of the subjects : 4
4
3
3
2
2
1
1
Name of the Student : Hitesh Sharma
USN of the Student : 1BM23CS114
Marks of the subjects :
90 89 82 77 76 85 88 96
Credits of the subjects :
4 4 3 3 2 2 1 1
The SGPA of the student : 9.0
HITESH SHARMA
1BM23CS114
PS C:\Users\hites\OneDrive\Desktop\javaproj>

```

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:

```
import java.util.Scanner;
public class Book
{
    private String name; //author's name
    private String author; //author's name
    private double price;
    private int numPages; //number of pages

    public Book(String name, String author, double price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String getName()
    {
        return name;
    }

    public String getAuthor()
    {
        return author;
    }

    public double getPrice()
    {
        return price;
    }

    public int getNumPages()
    {
        return numPages;
    }
}
```

```

        {
            this.name = name
        }

    public void setAuthor (String author)
    {
        this.author = author
    }

    public void setPrice (double price)
    {
        this.price = price
    }

    public String toString ()
    {
        return "Book [Name = "+name+", Author = "+author+", Price = "+price+", NumPages = "+numPages+"]";
    }

    import java.util.*;
    public class BookStore
    {
        public static void main (String args[])
        {
            Scanner sc = new Scanner (System.in);
            System.out.println ("Enter the no. of books : ");
            int n = sc.nextInt ();
            Book [] books = new Book[n];
        }
    }

```

```

for (int i=0; i<n; i++)
{
    System.out.println ("Enter the Details of Book");
    System.out.println ("Enter the name of book : ");
    String name = sc.nextLine();
    System.out.println ("Enter the name of author : ");
    String author = sc.nextLine();
    System.out.println ("Enter the price : ");
    double price = sc.nextDouble();
    System.out.println ("Enter the number of pages : ");
    int numPages = sc.nextInt();

    books[i] = new Book (name, author, price, numPages);
}

```

Review

```

        System.out.println ("Book details are ");
        for (int i=0; i<n; i++)
        {
            System.out.println (books[i].toString ());
        }
    }

```

CODE:

```
import java.util.Scanner;
class Book {
    private String name;
    private String author;
    private double price;
    private int numpages;
    public Book(String name, String author, double price, int numpages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numpages = numpages;
    }
    public String getName() {
        return name;
    }
    public String getAuthor() {
        return author;
    }
    public double getPrice() {
        return price;
    }
    public int getNumpages() {
        return numpages;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public void setNumpages(int numpages) {
        this.numpages = numpages;
    }
    @Override
    public String toString() {
        return "Book Details" + "\n" +
            "Name : " + name + "\n" +
            "Author : " + author + "\n" +
            "Price : " + price + "\n" +
            "Numpages : " + numpages + "\n";
    }
}
public class Bookstore {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the no. of books :");
        int n = sc.nextInt();
```

```

sc.nextLine();
Book[] books = new Book[n];
for (int i = 0; i < n; i++) {
    System.out.println("Enter the details of book:" + (i + 1));
    System.out.println("Enter the name of the book :");
    String name = sc.nextLine();
    System.out.println("Enter the name of the author:");
    String author = sc.nextLine();
    System.out.println("Enter the price of the book: ");
    double price = sc.nextDouble();
    System.out.println("Enter the no. of pages ");
    int numPages = sc.nextInt();
    sc.nextLine();
    books[i] = new Book(name, author, price, numPages);
}
System.out.println("Books details are::");
for (int i = 0; i < n; i++) {
    System.out.println(books[i].toString());
}
System.err.println("HITESH SHARMA");
System.out.println("1BM23CS114");
}
}

```

OUTPUT:

```
Enter the no. of books :  
2  
Enter the details of book:1  
Enter the name of the book :  
Java Basics  
Enter the name of the author:  
Danish K  
Enter the price of the book:  
499  
Enter the no. of pages  
350  
Enter the details of book:2  
Enter the name of the book :  
Python  
Enter the name of the author:  
599  
Enter the price of the book:  
250  
Enter the no. of pages  
200  
Books details are::  
Book Details  
Name : Java Basics  
Author : Danish K  
Price : 499.0  
NumPages : 350  
  
Book Details  
Name : Python  
Author : 599  
Price : 250.0  
NumPages : 200  
  
HITESH SHARMA  
1BM23CS114  
PS C:\Users\hites\OneDrive\Desktop\javaproj> █
```

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:

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. Each one of the classes contain only the method printArea() that prints the area of the given shape!

CODE :-

```
abstract class Shape {  
    int d1, d2;  
  
    Shape (int d1, int d2) {  
        this.d1 = d1;  
        this.d2 = d2;  
    }  
  
    abstract void printArea();  
}  
  
class Rectangle extends Shape {  
    Rectangle (int length, int breadth) {  
        super (length, breadth);  
    }  
  
    @Override  
    void printArea () {  
        int area = d1 * d2;  
        System.out.println ("Area of Rectangle : " + area);  
    }  
}  
  
class Triangle extends Shape {  
    Triangle (int base, int height) {  
        super (base, height);  
    }  
  
    @Override  
    void printArea () {  
        double area = 0.5 * d1 * d2;  
        System.out.println ("Area of Triangle : " + area);  
    }  
}  
  
class Circle extends Shape {  
    Circle (int radius) {  
        super (radius, 0);  
    }  
  
    @Override  
    void printArea () {  
        double area = Math.PI * d1 * d2;  
        System.out.println ("Area of Circle : " + area);  
    }  
}  
  
public class Shape_abstract {  
    public static void main (String [] args) {  
        Rectangle rectangle = new Rectangle (12, 5);  
        Triangle triangle = new Triangle (12, 5);  
        Circle circle = new Circle (5);  
    }  
}
```

CODE:

```
abstract class Shape {  
    int dim1, dim2;  
  
    Shape(int dim1, int dim2) {  
        this.dim1 = dim1;  
        this.dim2 = dim2;  
    }  
  
    abstract void printArea();  
}  
class Rectangle extends Shape {  
  
    Rectangle(int length, int breadth) {  
        super(length, breadth);  
    }  
  
    @Override  
    void printArea() {  
        int area = dim1 * dim2;  
        System.out.println("Area of Rectangle: " + area);  
    }  
}  
class Triangle extends Shape {  
    // Constructor for Triangle  
    Triangle(int base, int height) {  
        super(base, height);  
    }  
  
    @Override  
    void printArea() {  
        double area = 0.5 * dim1 * dim2;  
        System.out.println("Area of Triangle: " + area);  
    }  
}  
class Circle extends Shape {  
  
    Circle(int radius) {  
        super(radius, 0);  
    }  
  
    @Override  
    void printArea() {  
        double area = Math.PI * dim1 * dim1;  
        System.out.println("Area of Circle: " + area);  
    }  
}  
public class Shapes1 {  
    public static void main(String[] args) {  
  
        Shape rectangle = new Rectangle(10, 5);  
        Shape triangle = new Triangle(10, 5);
```

```
Shape circle = new Circle(7);

rectangle.printArea();
triangle.printArea();
circle.printArea();
}
}
```

OUTPUT:>

```
cd "c:\Users\hites\OneDrive\Desktop\javaprog" ; if ($?) { javac Shapes1.java } ; if (?) { java Shapes1 }

Area of Rectangle: 50
Area of Triangle: 25.0
Area of Circle: 153.93804002589985
PS C:\Users\hites\OneDrive\Desktop\javaprog>
```

PROGRAM 5:

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

ALGORITHM:

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called current account provider but no saving account provides compound interest and withdrawal facilities but no cheque book facility. The 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 class Cur-Acc and Sav-Acc.

(a) Accept deposit from customer and update balance
(b) Display the balance
(c) Compute and deposit interest
(d) Permit withdrawal and update the balance

```
import java.util.*;  
class Account {  
    String Name; int Acc-no; double balance; String Acc-type;  
    public Account (String Name, int Acc-no, double balance, String Acc-type) {  
        this.Name = Name; this.Acc-no = Acc-no;  
        this.balance = balance;  
        this.Acc-type = Acc-type;  
    }  
    public void deposit (double amount) {  
        if (amount > 0) {  
            balance += amount;  
            System.out.println ("The balance of account is :" + balance);  
        } else {  
            System.out.println ("Invalid amount entered");  
        }  
    }  
}
```

```

class Account {
    protected String customerName;
    protected String accountNumber;
    protected double balance;
    public Account(String customerName, String accountNumber, double initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful. Updated balance: " + balance);
        } else {
            System.out.println("Deposit unsuccessful. Amount must be positive.");
        }
    }
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Amount withdrawn: " + amount);
            System.out.println("Current balance: " + balance);
        } else {
            System.out.print("Insufficient balance");
        }
    }
}

class Current extends Account {
    public Current (String Name, int Acc.no, double balance,
                  String Acctype) {
        Super (Name, Acc.no, balance, "Current");
    }
}

int minimum_balance = 700;
int service_charge = 50;

public void withdraw (double amount) {
    if (amount >= minimum_balance) {
        balance = balance - amount;
        balance = balance - service_charge;
        System.out.println("Service charge applied");
    } else {
        balance = balance - amount;
        System.out.println("Withdraw successful of amount");
    }
}

class Bank {
    public static void main (String [] args) {
        Scanner sc = new Scanner();
        Sav-Acc Sav-Acc = new Account ("Hitesh", "45068666
5000, "Savings");

        Cur-Acc Cur-Acc = new Account ("Hemant", "450687780
7000, "Current");

        Sav-Acc deposit (1000);
        Sav-Acc com_interest (100);
        Sav-Acc . withdraw (200);
        Cur-Acc . deposit (2000);
        Cur-Acc . withdraw (1000);
    }
}

```

CODE:>

```

class Account {
    protected String customerName;
    protected String accountNumber;
    protected double balance;
    public Account(String customerName, String accountNumber, double
initialBalance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful. Updated balance: " + balance);
        } else {
    }
}

```

```

        System.out.println("Invalid deposit amount.");
    }
}
public void displayBalance() {
    System.out.println("Current balance: " + balance);
}
}
class SavingsAccount extends Account {
    private final double interestRate;
    public SavingsAccount(String customerName, String accountNumber, double initialBalance, double interestRate) {
        super(customerName, accountNumber, initialBalance);
        this.interestRate = interestRate;
    }
    public void computeAndDepositInterest(int years) {
        double interest = balance * Math.pow(1 + interestRate / 100, years) - balance;
        balance += interest;
        System.out.println("Interest deposited for " + years + " years. Updated balance: "
" + balance);
    }
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient balance.");
        }
    }
}
class CurrentAccount extends Account {
    private final double minimumBalance;
    private final double serviceCharge;
    public CurrentAccount(String customerName, String accountNumber, double initialBalance, double minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, initialBalance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            if (balance < minimumBalance) {

```

```

        balance -= serviceCharge;
        System.out.println("Minimum balance not maintained. Service charge
imposed.");
    }
    System.out.println("Withdrawal successful. Updated balance: " + balance);
} else {
    System.out.println("Invalid withdrawal amount or insufficient balance.");
}
}
}

public class Bank1 {
    public static void main(String[] args) {
        SavingsAccount savingsAccount = new SavingsAccount("Alice", "SA12345",
1000, 5);
        CurrentAccount currentAccount = new CurrentAccount("Bob", "CA12345",
2000, 1000, 50);
        System.out.println("Savings Account:");
        savingsAccount.deposit(500);
        savingsAccount.displayBalance();
        savingsAccount.computeAndDepositInterest(2);
        savingsAccount.withdraw(300);
        savingsAccount.displayBalance();
        System.out.println("\nCurrent Account:");
        currentAccount.deposit(1000);
        currentAccount.displayBalance();
        currentAccount.withdraw(1500);
        currentAccount.displayBalance();
        currentAccount.withdraw(1000);
        currentAccount.displayBalance();
    }
}

```

OUTPUT:

```
> cd "c:\U
Savings Account:
Deposit successful. Updated balance: 1500.0
Current balance: 1500.0
Interest deposited for 2 years. Updated balance: 1653
Withdrawal successful. Updated balance: 1353.75
Current balance: 1353.75

Current Account:
Deposit successful. Updated balance: 3000.0
Current balance: 3000.0
Withdrawal successful. Updated balance: 1500.0
Current balance: 1500.0
Minimum balance not maintained. Service charge imposed
Withdrawal successful. Updated balance: 450.0
Current balance: 450.0
PS C:\Users\hites\OneDrive\Desktop\javaproj> █
```

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>

Student :

```
package CIE;
public class Student {
    public string USN;
    public string name;
    public int sem;

    public Student (string USN, string name, int sem) {
        this.USN = USN;
        this.name = name;
        this.sem = sem;
    }

    public void displayStudentInfo () {
        System.out.println ("USN: " + USN);
        System.out.println ("Name: " + name);
        System.out.println ("Semester: " + sem);
    }
}
```

Bafna Gold
Date: Page 27

```

Internals
package CIE;
import java.util.*;
public class Internals extends Student {
    public int[] marks = new int[5];
    public Internals (String USN, String name, int sem) {
        super (USN, name, sem);
    }
    public void getintmarks () {
        Scanner s = new Scanner (System.in);
        for (int i = 0; i < n; i++) {
            System.out.println ("Enter internal marks");
            System.out.print ("Course " + (i+1) + ":" );
            marks [i] = s.nextInt ();
            s.nextLine ();
        }
    }
    public void displayintmarks () {
        System.out.println ("Internal Marks:");
        for (int i = 0; i < n; i++) {
            System.out.print ("Course " + (i+1) + ":" + marks [i]);
        }
    }
    public void getextmarks () {
        Scanner s = new Scanner (System.in);
        for (int i = 0; i < 5; i++) {
            System.out.println ("Enter external marks for course " + (i+1) + ":" );
            extmarks [i] = s.nextInt ();
            s.nextLine ();
        }
    }
    public void displayextmarks () {
        System.out.println ("External Marks:");
        for (int i = 0; i < 5; i++) {
            System.out.println ("Course " + (i+1) + " ext marks: " + extmarks [i]);
        }
    }
    public void finalrank () {
        System.out.println ("Final Marks (Internals + Externals):");
        for (int i = 0; i < 5; i++) {
            int finalMark = mark [i] + extmarks [i];
            System.out.println ("Course " + (i+1) + " final rank: " + finalMark);
        }
    }
}

```

Final Marks:

```
import CTE;  
import SFE;  
import java.util.*;
```

```
public class Main_Marks {  
    public static void main (String [] args) {  
        Scanner S = new Scanner (System.in);  
        System.out.println ("Enter number of students:");  
        int n = S.nextInt();
```

```
        External student = new External (sum, name, sem);
```

```
        student.displayStudentInfo ();
```

```
        System.out.println ("Enter Internal Marks:");
```

```
        student.getIntMarks ();
```

```
        student.displayIntMarks ();
```

```
        System.out.println ("Enter External Marks:");
```

```
        student.getExtMarks ();
```

```
        student.displayExtMarks ();
```

```
        student.finalMarks ();
```

3. close

3

3

CODE:>

```
package CIE;
public class Personal {
    String usn;
    String name;
    int sem;
    public Personal(String usn, String name, int sem){
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
    public void displaydetails(){
        System.out.println("Name:" + name);
        System.out.println("USN:" + usn);
        System.out.println("Sem:" + sem);
    }
}

package CIE;
import java.util.Scanner;
public class Internals extends Personal {
    public Internals(String usn, String name, int sem){
        super(usn, name, sem);
    }
    public int[] intmarks = new int[5];
    public void getintmarks(){
        Scanner s = new Scanner(System.in);
        for(int i=0; i<5; i++) {
            System.out.println("enter internal marks:");
            intmarks[i] = s.nextInt();
            s.nextLine();
        }
    }
    public void dispintmarks(){
        for(int i=0; i<5; i++){
            System.out.println("internal marks:" + intmarks[i]);
        }
    }
}

package SEE;
import CIE.*;
import java.util.Scanner;
public class Externals extends Internals{
    public Externals(String usn, String name, int sem){
        super(usn, name, sem);
    }
    int[] extmarks = new int[5];
    public void getextmarks(){
        Scanner s = new Scanner(System.in);
        for(int i=0; i<5; i++){
            System.out.println("enter external marks:");
            extmarks[i] = s.nextInt();
            s.nextLine();
        }
    }
    public void displayextmarks(){
```

```

        for(int i=0;i<5;i++) {
            System.out.println("external marks:" + extmarks[i]);
        }
    }
    public void totalmarks(){
        for(int i=0;i<5;i++){
            int total = intmarks[i]+extmarks[i];
            System.out.println("total marks:"+total);
        }
    }
}

import SEE.*;
import java.util.Scanner;
public class Mainmarks {
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("enter no of students:");
        int n = s.nextInt();
        s.nextLine();
        for(int i=0;i<n;i++){
            System.out.println("name:");
            String name = s.nextLine();
            System.out.println("usn:");
            String usn = s.nextLine();
            System.out.println("Sem:");
            int sem = s.nextInt();
            s.nextLine();
            Externals e = new Externals(usn,name,sem);
            e.displaydetails();
            e.getintmarks();
            e.dispmarks();
            e.getextmarks();
            e.displayextmarks();
            e.totalmarks();
            System.out.println("HITESH SHARMA");
            System.out.println("1BM23CS114");
        }
    }
}

```

OUTPUT:

```
enter no of students:
```

```
1
```

```
name:
```

```
hitesh sharma
```

```
usn:
```

```
1BM23CS114
```

```
Sem:
```

```
3
```

```
Name:hitesh sharma
```

```
USN:1BM23CS114
```

```
Sem:3
```

```
enter internal marks:
```

```
43
```

```
enter internal marks:
```

```
44
```

```
enter internal marks:
```

```
45
```

```
enter internal marks:
```

```
4
```

```
enter internal marks:
```

```
45
```

```
internal marks:43
```

```
internal marks:44
```

```
internal marks:45
```

```
internal marks:4
```

```
internal marks:45
```

```
enter external marks:  
45  
enter external marks:  
46  
enter external marks:  
47  
enter external marks:  
48  
enter external marks:  
49  
external marks:45  
external marks:46  
external marks:47  
external marks:48  
external marks:49  
total marks:88  
total marks:90  
total marks:92  
total marks:52  
total marks:94  
HITESH SHARMA  
1BM23CS114  
PS C:\Users\hites\OneDrive\Desktop\javaproj> █
```

Program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called

“Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age<0. In Son class,

implement a constructor that cases both father and son’s age and throws an exception if son’s age is \geq father’s age.

ALGORITHM:

Bafna Gold
Date: Page 38

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived 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 \geq father’s age.

```
import java.util.*;  
  
class WrongAge extends Exception {  
    public WrongAge (String message) {  
        super(message);  
    }  
}  
  
class Father {  
    int age;  
    public Father (int age) throws WrongAge {  
        if (age < 0)  
            throw new WrongAge ("Age cannot be negative");  
        this.age = age;  
        System.out.println ("Father's age is :" + this.age);  
    }  
}
```

```

class Son extends Father {
    int sonAge; // it's incorporated in child
    public Son (int fatherAge, int sonAge) {
        throws WrongAge { // if age is less than
            super (fatherAge); // it's greater than
        }
        if (SonAge >= FatherAge) {
            throws new WrongAge ("Son's age cannot be
            greater than or equal to Father's age : ");
        }
        this.SonAge = SonAge;
        System.out.println ("Son Age is : " + SonAge);
    }
}

public class ExceptionMain {
    public static void main (String [] args) {
        try {
            Son son1 = new Son (50, 40);
        } catch (WrongAge e) {
            System.out.println ("Error : " + e.getMessage());
        }
    }
}

```

proceed

CODE->

```
class WrongAgeException extends Exception {  
    public WrongAgeException(String message) {  
        super(message);  
    }  
}  
  
class Father {  
    protected int age;  
  
    public Father(int age) throws WrongAgeException {  
        if (age < 0) {  
            throw new WrongAgeException("Father's age cannot be negative.");  
        }  
        this.age = age;  
    }  
}  
  
class Son extends Father {  
    public int sonAge;  
  
    public Son(int fatherAge, int sonAge) throws WrongAgeException {  
        super(fatherAge);  
        if (sonAge < 0) {  
            throw new WrongAgeException("Son's age cannot be negative.");  
        }  
        if (sonAge >= fatherAge) {  
            throw new WrongAgeException("Son's age cannot be greater than or equal  
to Father's age.");  
        }  
        this.sonAge = sonAge;  
    }  
}  
  
public class Familytree {  
    public static void main(String[] args) {  
        try {  
            int fatherAge1 = 40;  
            int sonAge1 = 15;  
            Son son1 = new Son(fatherAge1, sonAge1);  
        }  
    }  
}
```

```
System.out.println("Father's age: " + son1.age);
System.out.println("Son's age: " + son1.sonAge);

int fatherAge2 = 50;
int sonAge2 = 60;
Son son2 = new Son(fatherAge2, sonAge2);
System.out.println("Father's age: " + son2.age);
System.out.println("Son's age: " + son2.sonAge);

} catch (WrongAgeException e) {
    System.out.println("Error: " + e.getMessage());
}
System.out.println("HITESH SHARMA");
System.out.println("1BM23CS114");
}
```

OUTPUT->

```
PS C:\Users\hites\OneDrive\Desktop\javaprog> cd "c:\Users\hites\OneDrive\Desktop\javaprog"
Father's age: 40
Son's age: 15
Error: Son's age cannot be greater than or equal to Father's age.
HITESH SHARMA
1BM23CS114
PS C:\Users\hites\OneDrive\Desktop\javaprog> 
```

PROGRAMM 8:

We have created an interface named `Polygon`. It includes a default method `getPerimeter()` and an abstract method `getArea()`.

We can calculate the perimeter of all polygons in the same manner so we implemented the body of `getPerimeter()` in `Polygon`.

Now, all polygons that implement `Polygon` can use `getPerimeter()` to calculate perimeter.

However, the rule for calculating the area is different for different polygons.
Hence, `getArea()` is included without implementation.

Any class that implements `Polygon` must provide an implementation of `getArea()`.

ALGORITHM:

```
class Rectangle implements Polygon {  
    private double width;  
    private double length;  
    public Rectangle (double length, double width) {  
        this.length = length;  
        this.width = width;  
    }  
    public double getArea () {  
        return length * width;  
    }  
}  
  
public class RectangleMain {  
    public static void main (String [] args) {  
        Polygon rect = new Rectangle (2.0, 9);  
        System.out.println ("Area of Rectangle = " + rect.getArea());  
        System.out.println ("Perimeter of Rectangle = " + rect.getPerimeter());  
    }  
}
```

Output:
Area of Rectangle = 18
Perimeter of Rectangle = 22

Aug 19/11

CODE->

```
interface Polygon {  
    default double getPerimeter(){  
        return 0;  
    }  
    double getArea();  
}  
class Circless implements Polygon {  
    double radius;  
    Circless(double radius) {  
        this.radius = radius;  
    }  
    @Override  
    public double getArea() {  
        return Math.PI * radius * radius;  
    }  
    @Override  
    public double getPerimeter(){  
        return 2*Math.PI*radius;  
    }  
}  
class Square implements Polygon {  
    double side;  
    Square(double side) {  
        this.side = side;  
    }  
    @Override  
    public double getArea() {  
        return side*side;  
    }  
    @Override  
    public double getPerimeter(){  
        return 4*side;  
    }  
}  
public class Shapess{  
    public static void main(String[] args){  
        Circless c1 = new Circless(10);  
        double carea=c1.getArea();  
        double cperimeter=c1.getPerimeter();  
        Square s1 = new Square(10);  
    }  
}
```

```
        double sarea=s1.getArea();
        double sperimeter=s1.getPerimeter();
        System.out.println("Area of Circle:"+carea);
        System.out.println("Perimeter of Circle:"+cperimeter);
        System.out.println("Area of Sqaure:"+sarea);
        System.out.println("Perimeter of Square:"+sperimeter);
        System.out.println("HITESH SHARMA");
        System.out.println("1BM23CS114");
    }
}
```

OUTPUT->

```
PS C:\Users\hites\OneDrive\Desktop\javaproj> cd "c:\"
Area of Circle:314.1592653589793
Perimeter of Circle:62.83185307179586
Area of Sqaure:100.0
Perimeter of Square:40.0
HITESH SHARMA
1BM23CS114
PS C:\Users\hites\OneDrive\Desktop\javaproj> []
```

PROGRAM 9:

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->

Bafna Gold
Date: Page: 57

1 Write programm that creates 2 threads one shows "CSE" every 2 sec & one shows "BMSCE" every 10 sec.

Class TwoThread Together {

```
static class T1 extends Thread {
    @override
    public void run() {
        try {
            while (true) {
                System.out.println("BMSCE");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread 1 Interrupted");
        }
    }
}
```

static class T2 extends Thread {
 @override
 public void run() {
 try {
 while (true) {
 System.out.println("CSE");
 Thread.sleep(2000);
 }
 } catch (InterruptedException e) {
 System.out.println("Thread 2 interrupted");
 }
 }
}

```
public static void main (String [ ] args) {
```

```
T1 a = new T1 ();
```

```
T2 b = new T2 ();
```

```
a.start ();
```

```
b.start ();
```

o/p:

```
CSE
```

```
BMSCE
```

```
CSE
```

```
BMSCE
```

Output → 2:

Current Thread: Thread [#1, main, 5, main]

Name is: main

Output → 3:

Thread: main, State: New

CODE:-

```
class BMSCollegeThread extends Thread {  
    public void run() {  
        try {  
            for(int i=0;i<5;i++) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("BMSCollegeThread interrupted.");  
        }  
    }  
}  
  
class CSEThread extends Thread {  
    public void run() {  
        try {  
            for(int i=0;i<5;i++) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("CSEThread interrupted.");  
        }  
    }  
}  
  
public class ThreadDisplay {  
    public static void main(String[] args) {  
        BMSCollegeThread bmsThread = new BMSCollegeThread();  
        CSEThread cseThread = new CSEThread();  
  
        bmsThread.start();  
        cseThread.start();  
        System.out.println("HITESH SHARMA");  
        System.out.println("1BM23CS114");  
    }  
}
```

OUTPUT:-

```
> cd C:\Users\Hitesh Sharma  
HITESH SHARMA  
1BM23CS114  
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  
PS C:\Users\hites\OneDrive\Desktop\javaproj\CIE>
```

PROGRAM 10:

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

ALGORITHM: -

Write a program that creates user interface to perform integer division. User enters 2 numbers in text fields. Division of both numbers is displayed when divide is clicked. If either of them isn't

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

public class DivCalc {
    public static void main (String [] args) {
        JFrame frame = new JFrame ("Integer Calculator");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize (400, 250);
```

```
JPanel panel = new JPanel();
panel.setLayout (new GridLayout (4, 2));
JLabel labelNum1 = new JLabel ("Enter Num1:");
JTextField textField1 = new JTextField ();
JLabel labelNum2 = new JLabel ("Enter Num2:");
JTextField textField2 = new JTextField ();
```

```
JButton button = new JButton ("Divide");
button.addActionListener (new ActionListener () {
    public void actionPerformed (ActionEvent e) {
```

Bafna Gold
Date: Page: 43

```
try {
    int num1 = Integer.parseInt (textField1.getText ());
    int num2 = Integer.parseInt (textField2.getText ());
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog (frame, "Please Enter valid integers", "Input Error", JOptionPane.ERROR_MESSAGE);
} catch (ArithmeticException e) {
    JOptionPane.showMessageDialog (frame, e.getMessage (), "Math Error", JOptionPane.ERROR_MESSAGE);
}
panel.add (labelNum1);
panel.add (labelNum2);
panel.add (textField1);
panel.add (textField2);
panel.add (button Divide);
frame.add (panel);
frame.setVisible (true);
```

CODE:

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

public class DivCalc {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Integer Division Calculator");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 250);

        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 2));

        JLabel labelNum1 = new JLabel("Enter Num1: ");
        JTextField textNum1 = new JTextField();
        JLabel labelNum2 = new JLabel("Enter Num2: ");
        JTextField textNum2 = new JTextField();

        JButton buttonDivide = new JButton("Divide");

        buttonDivide.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(textNum1.getText());
                    int num2 = Integer.parseInt(textNum2.getText());

                    if (num2 == 0) {
                        throw new ArithmeticException("Cannot divide by zero");
                    }

                    int result = num1 / num2;
                    JOptionPane.showMessageDialog(frame, "Result: " + result + "\nName: Daivya Priyankumar Shah\nUSN: 1BM23CS084", "Division Result", JOptionPane.INFORMATION_MESSAGE);
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(frame, "Please enter valid integers.", "Input Error", JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(frame, ex.getMessage(), "Math
```

```
Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});

panel.add(labelNum1);
panel.add(textNum1);
panel.add(labelNum2);
panel.add(textNum2);
panel.add(buttonDivide);

frame.add(panel);
frame.setVisible(true);
}
}
```

OUTPUT->

