

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**
“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT
on**
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

Gopika Pushparajan (**1BM23CS101**)

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 Gopika Pushparajan (**1BM23CS101**), 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.

Surabhi S Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
--	--

Index

Sl. No.	Date	Experiment Title	Page No.
1	1/10/24	Quadratic Equation Implementation	4
2	8/10/24	Calculate SGPA of student	6
3	15/10/24	Class Book, <code>toString()</code> , array of objects	10
4	22/10/24	Abstract class implementation, finding area	14
5	29/10/24	Class Bank, Savings account and current account	18
6	12/11/24	Packages, CIE, SEE, calculating total marks	25
7	19/11/24	Exception handling	30
8	26/11/24	Interface	33
9	3/12/24	Threads	36
10	3/12/24	User interface	38

Github Link: <https://github.com/gopikapushparajan/java>

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:

3. Program to perform quadratic equation

```
import java.util.Scanner;
class Main {
    public static void main(String [] args) {
        int a,b,c,d,root1,root2;
        Scanner input = new Scanner(System.in);
        System.out.println("Enter value of a");
        a = input.nextInt();
        System.out.println("Enter value of b");
        b = input.nextInt();
        System.out.println("Enter value of c");
        c = input.nextInt();
        d = b*b - 4*a*c;
        System.out.println("Discriminant is :" + d);
        if (d < 0)
            System.out.println("No real solution");
        else if (d == 0)
            roots = -b/(2*a);
            System.out.println("Roots are equal, root1 = root2 = " + roots);
        else
            root1 = (-b + Math.sqrt(d))/((2*a));
            root2 = (-b - Math.sqrt(d))/((2*a));
            System.out.println("Root 1 is :" + root1);
            System.out.println("Root 2 is :" + root2);
    }
}
```

→ Output
3. Enter value of a
1
Enter value of b
5
Enter value of c
6
Discriminant is: 1.0
Root 1 is 1.0
Root 2 is -3.0

Enter value of a
43
Enter value of b
23
Enter value of c
54
Discriminant is: -8759.0
No real solution

Enter value of a
1
Enter value of b
2
Enter value of c
1
Discriminant is: 0.0
Roots are equal, root 1 = root 2 = -1.0

Code:

```
import java.util.*  
class Main {  
    public static void main(String[] args) {  
        double a,b,c,d,roots,root1,root2;  
        Scanner input = new Scanner(System.in);  
        System.out.print("Enter value of a: ");  
        a = input.nextDouble();  
        System.out.print("Enter value of b: ");  
        b = input.nextDouble();  
        System.out.print("Enter value c: ");  
        c = input.nextDouble();  
        d = b * b - 4 * a * c;  
        System.out.println("Discriminant is : " + d);  
        if (d < 0) {  
            System.out.println("No real solution");  
        }  
        else if (d == 0) {  
            roots = -b / (2 * a);  
            System.out.println("Roots are equal, root 1=root2=" + roots);  
        }  
        else {  
            root1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);  
            root2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);  
            System.out.println("Root 1 is: " + root1);  
            System.out.println("Root 2 is: " + root2);  
        }  
    }  
}
```

Output

```
C:\Users\Gopika\Desktop\java programs>javac Main.java  
  
C:\Users\Gopika\Desktop\java programs>java Main  
Enter value of a: 1  
Enter value of b: 5  
Enter value c: 6  
Discriminant is : 1.0  
Root 1 is: -2.0  
Root 2 is: -3.0  
  
C:\Users\Gopika\Desktop\java programs>java Main  
Enter value of a: 4  
Enter value of b: 23  
Enter value c: 54  
Discriminant is : -335.0  
No real solution  
  
C:\Users\Gopika\Desktop\java programs>javac Main.java  
Enter value of a: 1  
Enter value of b: 2  
Enter value c: 1  
Discriminant is : 0.0  
Roots are equal, root 1=root2=-1.0  
  
C:\Users\Gopika\Desktop\java programs>
```

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:

Bafna Gold
Date: _____
Page: _____
Connected program!

```
import java.util.*;
class Student{
    String USN;
    String name;
    int credits[];
    int marks[];
    int i, n;

    void inputdetails()
    {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter name:");
        name = input.nextLine();
        System.out.println("Enter USN:");
        USN = input.nextLine();
        System.out.println("Enter no of subjects");
        n = input.nextInt();
        marks = new int[n];
        credits = new int[n];
        System.out.println("Enter marks:");
        for (i = 0; i < n; i++)
        {
            marks[i] = input.nextInt();
        }
        for (i = 0; i < n; i++)
        {
            credits[i] = input.nextInt();
        }
    }
}
```

void displaydetails()
{
 System.out.println("Student name: " + name);
 System.out.println("Student USN: " + USN);
 System.out.println("Marks scored");
 for (i = 0; i < marks.length; i++)
 {
 System.out.println(marks[i]);
 }
 System.out.println("credits for subject");
 for (i = 0; i < credits.length; i++)
 {
 System.out.println(credits[i]);
 }
 System.out.println("SGPA is " +
 sgpacalculation());
}

double sgpacalculation()
{
 int totalSum = 0;
 int sumCredits = 0;
 for (i = 0; i < n; i++)
 {
 totalSum = totalSum + (marks[i] *
 credits[i]);
 sumCredits = sumCredits + credits[i];
 }
 return (double) totalSum / (sumCredits * 10);
}

Bafna Gold
Date: _____ Page: _____

```

Student()
{
    this.name = name;
    this.USN = USN;
}

public static void main(String[] args)
{
    Student obj = new Student();
    obj.inputDetails();
    obj.displayDetails();
}

→ OUTPUT
Enter name:
Gopika Pusphparajan
Enter USN:
IBM23CS101
Enter no. of subjects:
3
Enter marks:
89
99
87
Enter credits
4
4
4
Student name: Gopika Pusphparajan
Student USN: IBM23CS101
Marks scored,
89
99

```

87

credits for subject

4

4

4

~~SGPA is 9.166666666666666~~

~~89 99 87~~

~~9.10.24~~

Code:

```

import java.util.Scanner;

class Student {
    String name;
    String USN;
    int marks[];
    int credits[];
    int n;

    void inputDetails() {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter name: ");
        name = input.nextLine();
    }
}

```

```

System.out.print("Enter USN: ");
USN = input.nextLine();
System.out.println("Enter no. of subjects: ");
n = input.nextInt();
marks = new int[n];
credits = new int[n];
System.out.println("Enter marks: ");
for (int i = 0; i < n; i++) {
    marks[i] = input.nextInt();
}
System.out.println("Enter credits: ");
for (int i = 0; i < n; i++) {
    credits[i] = input.nextInt();
}
}
class Student {

String name;
String USN;
int marks[];
int credits[];
int n;

void displayDetails() {
    System.out.println("Student Name: " + name);
    System.out.println("Student USN: " + USN);
    System.out.println("Marks Scored:");
    for (int i = 0; i < marks.length; i++) {
        System.out.println(marks[i]);
    }
    System.out.println("Credits for Subjects:");
    for (int i = 0; i < credits.length; i++) {
        System.out.println(credits[i]);
    }
    System.out.println("SGPA is: " + sgpaCalculation());
}

double sgpaCalculation() {
    int total_sum = 0;
    int sum_credits = 0;
    for (int i = 0; i < n; i++) {
        total_sum += marks[i] * credits[i];
        sum_credits += credits[i];
    }
    return (double) total_sum / sum_credits * 100;
}
}
class Student {
    String name;
    String USN;

```

```

Student(String name, String USN) {
    this.name = name;
    this.USN = USN;
}

public static void main(String[] args) {
    Student obj = new Student("Student Name", "Student USN");
    obj.inputDetails();
    obj.displayDetails();
}
}

```

Output:

```

Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gopika\Desktop\java programs>javac Studentdetails.java

C:\Users\Gopika\Desktop\java programs>java Studentdetails
Enter Student Name: Gopika Pushparajan
Enter Student USN: 1BM23CS101
Enter the number of subjects:
3
Enter marks for each subject:
89
99
87
Enter credits for each subject:
4
4
4

Student Details:
Name: Gopika Pushparajan
USN: 1BM23CS101

Marks Scored:
Subject 1: 89
Subject 2: 99
Subject 3: 87

Credits for Subjects:
Subject 1: 4
Subject 2: 4
Subject 3: 4

SGPA: 91.66666666666667

```

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:

LAB 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 Student {  
    private String name;  
    private String author;  
    private double price;  
    private int numPages;  
    public void student(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 :  
            name : " + name + ",  
            Author name : " + author + ",  
            Price : " + price + ",  
            Number of pages : " + numPages);  
}
```

```

public class Bookstore {
    public static void main (String [] args) {
        Scanner input = new Scanner (System
            .in);
        System.out.println ("Enter no. of Books");
        int n = input.nextInt ();
        input.nextLine ();
        Book [] books = new Book [n];
        for (int i=0; i<n; i++) {
            System.out.print ("Enter Book name:");
            name = input.nextLine ();
            System.out.print ("Enter Author name:");
            author = input.nextLine ();
            System.out.print ("Enter Price:");
            price = input.nextDouble ();
            System.out.print ("Enter No. of Pages");
            num_pages = input.nextInt ();
            books [i] = new Book (name, author,
                price, num_pages);
        }
        for (int i=0; i<books.length; i++) {
            System.out.print (books [i]);
        }
    }
}
```

→ Output
 Enter number of Books
 1
 Enter Book Name!
~~Gopika~~ abc 1BM23CS101
 Enter Author name!
 Gopika Pushparajen
 Enter Price!
 34
 Enter No. of Pages:
 21
 Book name: 1BM23CS101
 Author: Gopika Pushparajen
 Price: 34.0
 Number of Pages: 21
 15.10

Code:

```

import java.util.*;
class Book
{
    private String name;
    private String author;
    private double price;
    private int num_pages;
    public Book(String name, String author, double price, int num_pages)
    {
        this.name=name;
        this.author=author;
        this.price=price;
        this.num_pages=num_pages;
    }
}

```

```

public String getName()
{
    return name;
}
public String getAuthor()
{
    return author;
}
public double getPrice()
{
    return price;
}
public int getNum_pages()
{
    return num_pages;
}

public void setName()
{
    this.name=name;
}
public void setAuthor()
{
    this.author=author;
}
public void setPrice()
{
    this.price=price;
}
public void setNum_pages()
{
    this.num_pages=num_pages;
}

@Override
public String toString()
{
    return "(Book name:" + name + "\nAuthor:" + author + "\nPrice:" + price +
    "\nNumber of pages:" + num_pages+ ")";
}
}

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

```

```

Scanner input=new Scanner(System.in);
System.out.println("Enter number of books: ");
int n=input.nextInt();
input.nextLine();
Book[] books=new Book[n];

for(int i=0;i<n;i++)
{
    System.out.println("Enter Book Name: ");
    String name=input.nextLine();
    System.out.println("Enter Author name: ");
    String author=input.nextLine();
    System.out.println("Enter Price: ");
    double price=input.nextDouble();
    System.out.println("Enter No. of Pages: ");
    int num_pages=input.nextInt();

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

for(int i=0;i<books.length;i++)
{
    System.out.println(books[i]);
}
}

```

Output:

```

Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\bms 101>javac BookStore.java

C:\Users\Admin\Desktop\bms 101>java BookStore
Enter number of books:
1
Enter Book Name:
1BM23CS101
Enter Author name:
GopikaPushparajan
Enter Price:
34
Enter No. of Pages:
21
(Book name:1BM23CS101
Author:GopikaPushparajan
Price:34.0
Number of pages:21)

C:\Users\Admin\Desktop\bms 101>

```

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:

22/10/24

```
import java.util.*;
abstract class shape
{
    int side1;
    int side2;
    abstract void printArea(int side1, int side2);
    public shape (int side1, int side2)
    {
        this.side1 = side1;
        this.side2 = side2;
    }
    class rectangle extends shape
    {
        public rectangle (int length, int breadth)
        {
            super(length,breadth);
        }
        void printArea(int side1, int side2)
        {
            int area;
            area = side1 * side2;
            System.out.println ("Area of rectangle = " + area);
        }
    }
    class triangle extends shape
    {
        public triangle (int height, int base)
        {
            super(height,base);
        }
    }
}
```

```
void printArea (int side1, int side2)
{
    double area;
    area = 0.5 * side1 * side2;
    System.out.println ("Area of triangle = " + area);
}

class circle extends shape
{
    int radius;
    public circle (int radius)
    {
        super(radius,radius);
    }
    void printArea (int side1, int side2)
    {
        double area;
        area = Math.PI * side1 * side2;
        System.out.println ("Area of circle = " + area);
    }
}

public class labFourTwo
{
    public static void main (String [] args)
    {
        Scanner input = new Scanner (System.in);
        System.out.print ("Enter length and breadth of rectangle: ");
        int length = input.nextInt();
        int breadth = input.nextInt();
        shape rectangle = new Rectangle (length, breadth);
    }
}
```

```

rectangle.printArea(length, breadth);
System.out.println("Enter height and
base of triangle");
int height = input.nextInt();
int base = input.nextInt();
Shape triangle = new Triangle(height,
base);
triangle.printArea(height, base);
System.out.println("Enter radius of circle");
int radius = input.nextInt();
Shape circle = new Circle(radius);
circle.printArea(radius, 0);

```

3

→ Output

Enter length and breadth of rectangle.

5

2

Area of rectangle = 10

Enter height and base of triangle.

6

6

Area of triangle = 18.0

Enter radius of circle.

5

Area of circle = 78.53981633974483

Ques

Code:

```

import java.util.*;
abstract class Shape
{
    int side1;
    int side2;
    abstract void printArea(int side1,int side2);
    public Shape(int side1,int side2)
    {
        this.side1=side1;
        this.side2=side2;
    }
}
class Rectangle extends Shape
{
    public Rectangle(int length,int breadth)
    {
        super(length,breadth);
    }
    void printArea(int side1,int side2)

```

```

{
    int area;
    area=side1*side2;
    System.out.println("Area of rectangle=" + area);
}
}

class Triangle extends Shape
{

    public Triangle(int height,int base)
    {
        super(height,base);
    }

    void printArea(int side1,int side2)
    {
        double area;
        area=0.5*side1*side2;
        System.out.println("Area of triangle=" + area);
    }
}

class Circle extends Shape
{
    int radius;
    public Circle(int radius)
    {
        super(radius,radius);
    }

    void printArea(int side1,int side2)
    {
        double area;
        area=Math.PI*side1*side1;
        System.out.println("Area of circle=" + area);
    }
}

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

        System.out.println("Enter length and breadth of rectangle: ");
        int length=input.nextInt();
        int breadth=input.nextInt();
        Shape rectangle=new Rectangle(length,breadth);
    }
}

```

```

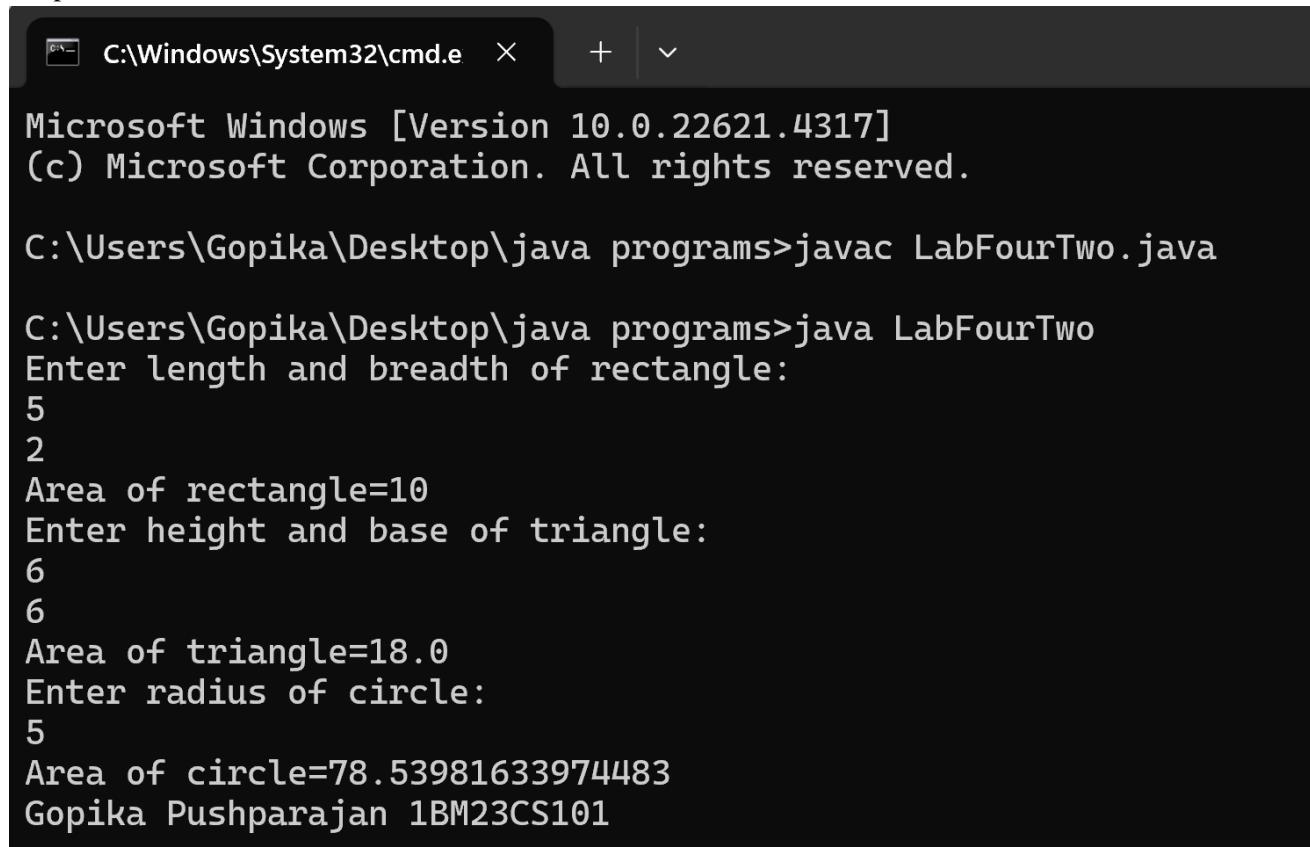
rectangle.printArea(length,breadth);

System.out.println("Enter height and base of triangle:");
int height=input.nextInt();
int base=input.nextInt();
Shape triangle=new Triangle(height,base);
triangle.printArea(height,base);

System.out.println("Enter radius of circle:");
int radius=input.nextInt();
Shape circle=new Circle(radius);
circle.printArea(radius,0);
System.out.println("Gopika Pushparajan 1BM23CS101");
}
}

```

Output:



```

C:\Windows\System32\cmd.e  X  +  ▾
Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gopika\Desktop\java programs>javac LabFourTwo.java

C:\Users\Gopika\Desktop\java programs>java LabFourTwo
Enter length and breadth of rectangle:
5
2
Area of rectangle=10
Enter height and base of triangle:
6
6
Area of triangle=18.0
Enter radius of circle:
5
Area of circle=78.53981633974483
Gopika Pushparajan 1BM23CS101

```

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:

```

class Account
{
protected String customer-name;
String account-no;
double balance;

public Account (String customer-name,
                String account-no)
{
    this.customer-name = customer-name;
    this.account-no = account-no;
    this.balance = 0.0;
}

public void deposit (amount)
{
    if (amount > 0)
        balance = balance + amount;
    System.out.println ("Deposited: "
                        + amount);
    else
        System.out.println ("Invalid amount");
}

public void withdraw (amount)
{
    if (amount > 0 & < amount < balance)
        balance = balance - amount;
    System.out.println ("Amount withdrawn"
                        + amount);
}
}

```

```

class Bafna Gold extends Account
{
    public void display()
    {
        System.out.println ("The account
                            balance is " + balance);
    }
}

class SavAcc extends Account
{
    double interest-rate;

    public SavAcc (String customer-name,
                  String account-no,
                  double interest-rate)
    {
        super (customer-name,
               account-no);
        this.interest-rate = interest-rate;
    }

    public void depositInterest ()
    {
        double interest = balance * interest-rate / 100;
        deposit (interest);
        System.out.println ("Interest deposited"
                            + interest);
    }
}

```

```

class Current extends Account {
    double minimum_balance;
    double service_charge;

    public void imposeServiceCharge();
    public Current (String customer_name,
                   String account_no,
                   double minimum_balance,
                   double service_charge) {
        super (customer_name, account_no);
        this.minimum_balance = minimum_balance;
        this.service_charge = service_charge;
    }

    public void withdraw (double amount) {
        super.withdraw (amount);
        if (amount > 0 && amount < balance)
            if (balance < minimum_balance)
                imposeServiceCharge();
        else
            System.out.println ("Insufficient funds");
    }

    public void imposeServiceCharge () {
        balance = balance - service_charge;
    }
}

```

Bafna Gold
Date: _____
Page: _____

```

System.out.println ("Imposed service charge");

public class Bank {
    public static void main (String [] args) {
        SavingsAccount sa = new SavingsAccount ("Rahul", 549, 2000, 5.0);
        CurrentAccount ca = new CurrentAccount ("Jane", "203", 1500, 500);

        sa.deposit (1200);
        sa.withdraw (500);
        sa.display ();
        sa.depositInterest ();

        ca.deposit (2500);
        ca.withdraw (200);
        ca.withdraw (200);
        ca.display ();
    }
}

Scanner input = new Scanner (System.in);
System.out.println ("Enter following services are available in:");
1. Savings In 2. Current;

System.out.println ("Enter Account type");
1. Savings In 2. Current;

```

```

int type = input.nextInt();
input.nextLine();
System.out.println ("Enter name");
String name = input.nextLine();
System.out.println ("Enter Account no");
String accno = input.nextLine();
System.out.print

System.out.println ("Enter service provided");
System.out.println ("1. Deposit");
System.out.println ("2. withdraw");
System.out.println ("3. Check Interest");
System.out.println ("4. withdraw from savings account");
System.out.println ("5. Current account");

if (type == 1)
    System.out.println ("Enter interest interest rate");
    double interestRate = input.nextDouble();
    account = new SavingsAccount (name, accno, interestRate);
else
    System.out.println ("Enter minimum balance");
    double minBal = input.nextDouble();
    System.out.println ("Enter service charge");
    double serviceCharge = input.nextDouble();
    account = new CurrentAccount (name, accno, minBal, serviceCharge);

do {
    System.out.println ("Services available");
    System.out.println ("1. Deposit");
    System.out.println ("2. withdraw");
    System.out.println ("3. Display Balance");
} while (true);

```

Bafna Gold
Date: _____
Page: _____

```

System.out.println ("1. Check Interest (for savings only)");
System.out.println ("2. Exit");

while (choice != 5) {
    System.out.println ("Enter choice");
    choice = input.nextInt();

    switch (choice) {
        case 1: {
            account.deposit (amount);
            System.out.println ("Enter amount to deposit");
            double amount = input.nextDouble();
            account.deposit (amount);
            break;
        }
        case 2: {
            System.out.println ("Enter amount to withdraw");
            double amount = input.nextDouble();
            account.withdraw (amount);
            break;
        }
        case 3: {
            account.display ();
            break;
        }
        case 4: {
            account.depositInterest ();
            break;
        }
        default case 5: {
            System.out.println ("Thank you");
            break;
        }
    }
}

```

default : Byteman.out.println("Invalid
in put");

3

→ Output
Enter account type
1. Savings
2. Current
1
Enter name:
Gopika
Enter account no:
123456
Enter interest rate:
5
Services available
1. Deposit
2. withdraw
3. Display Balance
4. credit Interest (for savings only)
5. Exit
Enter choice: 1
Enter amount to deposit: 1000
Deposited: 1000.0
Services available:
1. Deposit
2. withdraw
2. Display Balance

Bafna Gold
Date: Page:

4.
5.
Enter choice: 3
Account Balance: 1000.0
services available:
1. Deposit
2. withdraw
3. credit Int Display Balance
4. credit Interest (for savings only)
5. Exit
Enter choice: 4
Interest deposited: 50.0
services available:
1. Deposit
2. withdraw
3. Display Balance
4. credit Interest (for savings only)
5. Exit
Enter choice: 5
Thankyou!

Code:

```
import java.util.*;  
class Account {  
    String customer_name;  
    String account_no;  
    double balance;  
    public Account(String customer_name, String account_no, double initialBalance) {  
        this.customer_name = customer_name;  
        this.account_no = account_no;  
        this.balance = initialBalance;  
    }  
    public void deposit(double amount) {  
        if (amount > 0)  
        {  
            balance = balance + amount;  
            System.out.println("Deposited: " + amount);  
        }  
    }  
}
```

```

        }
    else {
        System.out.println("Invalid Amount");
    }
}

public void withdraw(double amount){
    if(amount>0 && amount<=balance)
    {
        balance=balance-amount;
        System.out.println("Amount withdrawn: " + amount);
    }
    else
    {
        System.out.println("Insufficient funds");
    }
}
public void displayBalance() {
    System.out.println("The account balance is: " + balance);
}
}

class SavAcc extends Account
{
    double interestrate;

    public SavAcc(String customer_name,String account_no,double initialBalance,double interestrate)
    {
        super(customer_name,account_no,initialBalance);
        this.interestrate=interestrate;
    }
    public void depositinterest()
    {
        double interest=balance*interestrate/100;
        deposit(interest);
        System.out.println("Interest deposited" + interest);
    }
}

class CurAcc extends Account
{
    double minimum_balance;
    double service_charges;

    public CurAcc(String customer_name,String account_no,double initialBalance, double
minimum_balance,double service_charges)
    {

```

```

super(customer_name,account_no,initialBalance);
this.minimum_balance=minimum_balance;
this.service_charges=service_charges;
}
public void withdraw(double amount)
{
    super.withdraw(amount);
    if(amount>0 && amount<=balance)
    {
        if (balance<minimum_balance)
        {
            imposeServiceCharge();
        }
    }
    else
    {
        System.out.println("Insufficient funds");
    }
}
public void imposeServiceCharge()
{
    balance=balance-service_charges;
    System.out.println("Imposed service charges");
}
}
public class Bank {
    public static void main(String[] args) {
        Scanner input=new Scanner(System.in);
        System.out.println("Enter Account type\n 1.Savings\n 2.Current");
        int type = input.nextInt();
        input.nextLine();

        System.out.print("Enter name: ");
        String name = input.nextLine();

        System.out.print("Enter account no: ");
        String accno = input.nextLine();
        Account account=null;
        if (type == 1) {
            System.out.println("Enter initial balance: ");
            double initialBalance=input.nextDouble();
            System.out.print("Enter interest rate: ");
            double interestRate = input.nextDouble();
            account = new SavAcc(name, accno,initialBalance, interestRate);
        }
    }
}

```

```

else {
    System.out.println("Enter initial balance: ");
    double initialBalance=input.nextDouble();

    System.out.print("Enter minimum balance: ");
    double minBal = input.nextDouble();

    System.out.print("Enter service charge: ");
    double serviceCharge = input.nextDouble();
    account = new CurAcc(name, accno,initialBalance,minBal, serviceCharge);
}

do {
    System.out.println("\nServices available:");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Display Balance");
    System.out.println("4. Credit Interest (for savings only)");
    System.out.println("5. Exit");

    System.out.print("Enter choice: ");
    int choice = input.nextInt();

    switch (choice) {
        case 1:
            System.out.print("Enter amount to deposit: ");
            double amount = input.nextDouble();
            account.deposit(amount);
            break;

        case 2:
            System.out.print("Enter amount to withdraw: ");
            double amount = input.nextDouble();
            account.withdraw(amount);
            break;

        case 3:
            account.displayBalance();
            break;

        case 4:
            account.depositInterest();
            break;

        case 5:
    }
}

```

```

        System.out.println("Thank you!");
        break;

    default:
        System.out.println("Invalid input");
        break;
    }
} while (choice != 5);
}
}

```

Output

```

Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gopika\Desktop\java programs>javac Bank.java

C:\Users\Gopika\Desktop\java programs>java Bank
Enter Account type
1. Savings
2. Current
1
Enter name: Gopika
Enter account no: 123456
Enter initial balance:
2000
Enter interest rate: 5

Services available:
1. Deposit
2. Withdraw
3. Display Balance
4. Credit Interest (for savings only)
5. Exit
Enter choice: 1
Enter amount to deposit: 1000
Deposited: 1000.0

Services available:
1. Deposit
2. Withdraw
3. Display Balance
4. Credit Interest (for savings only)
5. Exit
Enter choice: 3
The account balance is: 3000.0

Services available:
1. Deposit
2. Withdraw
3. Display Balance
4. Credit Interest (for savings only)
5. Exit

```

Services available:

1. Deposit
2. Withdraw
3. Display Balance
4. Credit Interest (for savings only)
5. Exit

Enter choice: 4
Deposited: 150.0
Interest deposited: 150.0

Services available:

1. Deposit
2. Withdraw
3. Display Balance
4. Credit Interest (for savings only)
5. Exit

Enter choice: 5
Thank you!

C:\Users\Gopika\Desktop\java programs>

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:

10/11/24

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;
import java.util.Scanner;
public class Student {
    protected String usn;
    protected String name;
    protected int sem;

    private void inputStudent() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter USN: ");
        usn = this.usn = input.nextLine();
        System.out.println("Enter name: ");
        this.name = input.nextLine();
        System.out.println("Enter semester: ");
        this.sem = input.nextInt();
    }

    public void getdetails() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter USN: ");
        usn = this.usn = input.nextLine();
        System.out.println("Enter name: ");
        this.name = input.nextLine();
        System.out.println("Enter semester: ");
        this.sem = input.nextInt();
    }
}

```

Bafna Gold
Date: _____
Page: _____

```

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

package CIE;
import java.util.Scanner;

public class Internals extends Student {
    protected int marks[5] = new int[5];

    public void inputCIEmarks() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter Internal Marks for 5 courses");
        marks[0] = input.nextInt();
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) +
                ": ");
            marks[i] = input.nextInt();
        }
    }

    public void displayCIEmarks() {
        System.out.println("Internal Marks:");
        for (int i = 0; i < 5; i++) {
            System.out.println(marks[i]);
        }
    }
}

```

```

package SEE;
import java.util.Scanner;
public class External extends Student {
    protected int externalMarks[] = new int[5];
    protected int finalMarks[] = new int[5];
    public External() {
        externalMarks = new int[5];
        finalMarks = new int[5];
    }
    public void getSEEMarks() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter External Marks for 5 courses:");
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter External Marks for course " + i);
            externalMarks[i] = input.nextInt();
        }
    }
    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + externalMarks[i];
        }
    }
}

```

```

Bafna Gold
Date: Page:
public void displayFinalMarks() {
    getdetails();
    display();
    System.out.println("External Marks:");
    for (int i = 0; i < 5; i++) {
        System.out.print(finalMarks[i]);
    }
}
import SEE.External;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter number of students");
        int n = input.nextInt();
        External[] students = new External[n];
        for (int i = 0; i < n; i++) {
            students[i] = new External();
            students[i].getSEEMarks();
            students[i].calculateFinalMarks();
        }
        for (int i = 0; i < n; i++) {
            students[i].displayFinalMarks();
        }
    }
}

```

→ Output:

```

Enter number of students: 1
Enter USN: 1 BM23C3101
Enter Name: Gopika Pushparajan
Enter Semester: 3
Enter Internal Marks for 5 courses:
Course: 20
Course: 25
Course: 22
Course: 20
Course: 28
Enter the External Marks for 5 courses:
Course: 20
Course: 25
Course: 22
Course: 20
Course: 28
USN: 1 BM23C3101
Name: Gopika Pushparajan
Semester: 3
Internal Marks:
20
25
22
20
28

```

Bafna Gold

Internal Marks
20
25
22
20
28
Final Marks:
50
60
54
68
68

Code:

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

    public void getdetails() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter USN: ");
        this.usn = input.nextLine();
        System.out.println("Enter Name: ");
        this.name = input.nextLine();
        System.out.println("Enter Semester: ");
        this.sem = input.nextInt();
    }
    public void displaydetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    protected int marks[] = new int[5];

    public void getCIEmarks() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter Internal Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.print("Course marks: ");
            marks[i] = input.nextInt();
        }
    }
    public void displayCIEmarks() {
        System.out.println("Internal Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println(marks[i]);
        }
    }
}
```

```

package SEE;
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
    protected int externalmarks[] = new int[5];
    protected int finalmarks[] = new int[5];

    public Externals() {
        externalmarks = new int[5];
        finalmarks = new int[5];
    }

    public void getSEEmarks() {
        Scanner input= new Scanner(System.in);
        System.out.println("Enter External Marks for 5 courses: ");
        for (int i = 0; i < 5; i++) {
            System.out.print("Course: ");
            externalmarks[i] = input.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalmarks[i] = marks[i] + externalmarks[i];
        }
    }

    public void displayFinalMarks() {
        getdetails();
        displayCIEmarks();

        System.out.println("External Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println(externalmarks[i]);
        }

        System.out.println("Final Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println(finalmarks[i]);
        }
    }
}

```

```

import SEE.Externals;

import java.util.Scanner;

```

```

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

        System.out.print("Enter number of students: ");
        int n = input.nextInt();

        Externals[] students = new Externals[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Externals();
            students[i].getdetails();
            students[i].getCIEmarks();
            students[i].getSEEmarks();
            students[i].calculateFinalMarks();
        }
        for (int i = 0; i < n; i++) {
            students[i].displayFinalMarks();

        }
    }
}

```

Output:

```

C:\Users\Gopika\Desktop\java programs>java Program15
Enter the number of students:
2

Enter details for Student 1:
USN: 1BM23CS101
Name: Gopika Pushparajan
Semester: 3
Enter marks for 5 courses (Internal followed by SEE):
Course 1 Internal Marks: 40
Course 1 SEE Marks: 30
Course 2 Internal Marks: 40
Course 2 SEE Marks: 33
Course 3 Internal Marks: 44
Course 3 SEE Marks: 35
Course 4 Internal Marks: 41
Course 4 SEE Marks: 32
Course 5 Internal Marks: 40
Course 5 SEE Marks: 34

Enter details for Student 2:
USN: 400
Name: abc
Semester: 3
Enter marks for 5 courses (Internal followed by SEE):
Course 1 Internal Marks: 40
Course 1 SEE Marks: 30
Course 2 Internal Marks: 20
Course 2 SEE Marks: 34
Course 3 Internal Marks: 34
Course 3 SEE Marks: 35
Course 4 Internal Marks: 36
Course 4 SEE Marks: 45
Course 5 Internal Marks: 40
Course 5 SEE Marks:
42

Final Marks for Student 1:
USN: 1BM23CS101
Name: Gopika Pushparajan
Semester: 3
Course-wise Marks (Internal + SEE):
Course 1: 70
Course 2: 73

```

```

Final Marks for Student 2:
USN: 400
Name: abc
Semester: 3
Course-wise Marks (Internal + SEE):
Course 1: 90
Course 2: 56
Course 3: 74
Course 4: 65
Course 5: 56

```

Program 7

Create an interface named Polygon including default method getPerimeter() and an abstract method getArea(). Implement the body of getPerimeter() in Polygon. getArea() is different for all polygons so it is included without implementation.

Algorithm:

19/11/24

Create an interface named Polygon (including default method getPerimeter() and an abstract method getArea()). Implement the body of getPerimeter() in Polygon. getArea() is different for all polygons so it is included without implementation

```

import java.util.*;
interface Polygon {
    double getPerimeter();
    double getArea();
}
class Triangle implements Polygon {
    double a;
    double b;
    double c;
    public Triangle(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }
    public double getPerimeter() {
        return (a+b+c);
    }
    public double getArea() {
        return (0.5 * (a*b));
    }
}

```

Class Square implements Polygon

```

double a;
public Square(double a) {
    this.a = a;
}
public double getPerimeter() {
    return (a*4);
}
public double getArea() {
    return (a*a);
}

public class Polygons {
    public static void main(String args[]) {
        Scanner input = new Scanner(System.in);
        int ch = 0;
        while (ch != 3) {
            System.out.println("1. Triangle");
            System.out.println("2. Square");
            System.out.println("3. Exit");
            ch = input.nextInt();
            switch (ch) {
                case 1:
                    System.out.println("Enter side:");
                    double q = input.nextDouble();
                    System.out.println("Enter height:");
                    double w = input.nextDouble();
                    System.out.println();
                    Polygon p = new Triangle(q, w);

```

System.out.println("Enter base:");
double e = input.nextDouble();
Polygon p = new Triangle(2, 4, e);
System.out.println(" Perimeter:" + p.getPerimeter() + " Area:" + p.getArea());
break;

case 2:
System.out.println("Enter side:");
double a1 = input.nextDouble();
Polygon p1 = new Square(a1);
System.out.println(" Perimeter:" + p1.getPerimeter() + " Area:" + p1.getArea());
break;

case 3:
System.out.println(" Thank you");
break;

default:
System.out.println(" Invalid");
break;

Output

```

1. Triangle
2. Square
3. Exit
Enter your choice : 1
Enter side:
2.5
Enter base:
3
Enter height:
1.2
Perimeter : 7.0
Area : 18.0
1. Triangle
2. Square
3. Exit
Enter your choice : 2
Enter side:
5
Perimeter : 20.0
Area : 25.0
1. Triangle
2. Square
3. Exit
Enter your choice : 3
Thank you.

```

Code:

```
import java.util.*;
interface Polygon {
    double getPerimeter();
    double getArea();
}

class Triangle implements Polygon {
    double a, b, c;

    public Triangle(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    public double getPerimeter() {
        return a + b + c;
    }

    public double getArea() {
        double s = (a + b + c) / 2;
        return Math.sqrt(s * (s - a) * (s - b) * (s - c));
    }
}

class Square implements Polygon {
    double side;

    public Square(double side) {
        this.side = side;
    }

    public double getPerimeter() {
        return 4 * side;
    }

    public double getArea() {
        return side * side;
    }
}

public class Polygons {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int ch = 0;

        while (ch != 3) {
            System.out.println("1. Triangle\n2. Square\n3. Exit\nEnter your choice:");
            ch = input.nextInt();
        }
    }
}
```

```

switch (ch) {
    case 1:
        System.out.println("Enter side a:");
        double a = input.nextDouble();
        System.out.println("Enter side b:");
        double b = input.nextDouble();
        System.out.println("Enter side c:");
        double c = input.nextDouble();
        Polygon p = new Triangle(a, b, c);
        System.out.println("Perimeter: " + p.getPerimeter());
        System.out.println("Area: " + p.getArea());
        break;
    case 2:
        System.out.println("Enter side:");
        double side = input.nextDouble();
        Polygon p2 = new Square(side);
        System.out.println("Perimeter: " + p2.getPerimeter());
        System.out.println("Area: " + p2.getArea());
        break;
    case 3:
        System.out.println("Thankyou");
        break;
    default:
        System.out.println("Invalid choice.");
}
}
}
}
}

```

Output:

```

Microsoft Windows [Version 10.0.22621.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Gopika\Desktop\java programs>javac Polygons.java

C:\Users\Gopika\Desktop\java programs>java Polygons
1. Triangle
2. Square
3. Exit
Enter your choice:
1
Enter side a:
23
Enter side b:
45
Enter side c:
67
Perimeter: 135.0
Area: 183.82651468164215
1. Triangle
2. Square
3. Exit
Enter your choice:
2
Enter side:
4
Perimeter: 16.0
Area: 16.0
1. Triangle
2. Square
3. Exit
Enter your choice:
3
Thankyou

```

Program 8

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:

Bafna Gold
Date: _____
Page: _____

LAB PROGRAM 8:

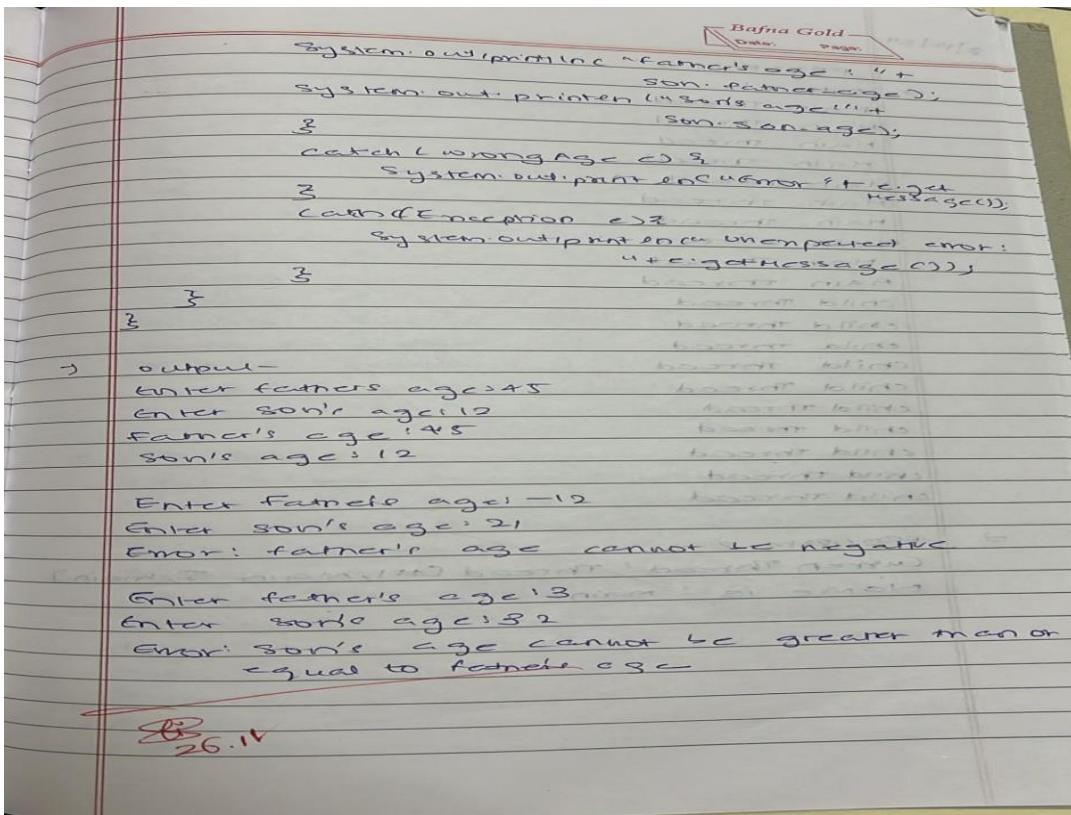
WHITE A PROGRAM THAT DEMONSTRATES HANDLING OF EXCEPTIONS IN INHERITANCE TREE. CREATE A BASE CLASS CALLED "FATHER" AND DERIVED CLASS CALLED "SON" WHICH EXTENDS THE BASE CLASS. IN FATHER CLASS, IMPLEMENT A CONSTRUCTOR WHICH TAKES THE AGE AND THROWS THE EXCEPTION WRONGAGE() WHEN THE INPUT AGE<0. IN SON CLASS, IMPLEMENT A CONSTRUCTOR THAT USES BOTH FATHER AND SON'S AGE AND THROWS AN EXCEPTION IF SON'S AGE IS \geq FATHER'S AGE

```
import java.util.*;  
  
class wrongAge extends Exception{  
    public wrongAge (String message){  
        super(message);  
    }  
}  
  
class father{  
    int father-age;  
    public father (int father-age) throws wrongAge{  
        if (father-age < 0){  
            throw new wrongAge ("father's age cannot be negative.");  
        }  
        this.father-age = father-age;  
    }  
}
```

class son extends father{
 int son-age;
 public son (int father-age, int son-age) throws wrongAge{
 super(father-age);
 if (son-age < 0){
 throw new wrongAge ("son's age cannot be negative.");
 }
 if (son-age >= father-age){
 throw new wrongAge ("son's age cannot be greater than or equal to father's age.");
 }
 this.son-age = son-age;
 }
}

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

 try{
 System.out.println ("Enter father's age: ");
 int father-age = input.nextInt();
 System.out.println ("Enter son's age: ");
 int son-age = input.nextInt();
 son son = new son (father-age, son-age);
 }
 }
}



Code:

```

import java.util.Scanner;
class WrongAge extends Exception {
    public WrongAge(String message) {
        super(message);
    }
}

class Father {
    int father_age;
    public Father(int father_age) throws WrongAgeException {
        if (father_age < 0) {
            throw new WrongAge("Father's age cannot be negative.");
        }
        this.age = age;
    }
}

class Son extends Father {
    int son_age;
    public Son(int father_age, int son_age) throws WrongAge{
        super(father_age);
        if (son_age < 0) {
            throw new WrongAge("Son's age cannot be negative.");
        }
    }
}

```

```

        if (son_age >= father_age) {
            throw new WrongAge("Son's age cannot be greater than or equal to Father's age.");
        }
        this.son_age = son_age;
    }
}

public class ExceptionHandling{
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        try {
            System.out.print("Enter Father's age: ");
            int father_age = input.nextInt();
            System.out.print("Enter Son's age: ");
            int son_age = input.nextInt();
            Son son = new Son(father_age, son_age);
            System.out.println("Father's age: " + son.father_age);
            System.out.println("Son's age: " + son.son_age);
        }
        catch (WrongAge e) {
            System.out.println("Error: " + e.getMessage());
        }
        catch (Exception e){
            System.out.println("Unexpected error: " + e.getMessage());
        }
    }
}

```

Output:

```

Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\bms 101>javac ExceptionHandling.java

C:\Users\Admin\Desktop\bms 101>java ExceptionHandling
Enter Father's age: 45
Enter Son's age: 12
Father's age: 45
Son's age: 12

C:\Users\Admin\Desktop\bms 101>javac ExceptionHandling.java

C:\Users\Admin\Desktop\bms 101>java ExceptionHandling
Enter Father's age: -12
Enter Son's age: 21
Error: Father's age cannot be negative.

C:\Users\Admin\Desktop\bms 101>javac ExceptionHandling.java

C:\Users\Admin\Desktop\bms 101>java ExceptionHandling
Enter Father's age: 3
Enter Son's age: 32
Error: Son's age cannot be greater than or equal to Father's age.

```

Program 9

Q1 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:

LAB PROGRAM

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 BMS Thread extends Thread {
 public void run() {
 try {
 while(true) {
 System.out.println("BMS College
of Engineering");
 Thread.sleep(10000);
 }
 } catch(InterruptedException e) {
 System.out.println("Thread
interrupted:" + e.getMessage());
 }
 }
}

```

class CSEThread extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted:" + e.getMessage());
        }
    }
}

public class DisplayThreads {
    public static void main(String[] args) {
        BMSThread bmsThread = new BMSThread();
        CSEThread cseThread = new CSEThread();
        bmsThread.start();
        cseThread.start();
    }
}

```

→ Output

BMS College of Engineering
CSE
CSE
CSE

BMS COLLEGE OF ENGINEERING
Bafna
Date: 03-12

Code:

```
class BMSThread extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("Thread interrupted: " + e.getMessage());  
        }  
    }  
}  
  
class CSEThread extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("Thread interrupted: " + e.getMessage());  
        }  
    }  
}  
  
public class DisplayThreads {  
    public static void main(String[] args) {  
        BMSThread bmsThread = new BMSThread();  
        CSEThread cseThread = new CSEThread();  
  
        bmsThread.start();  
        cseThread.start();  
    }  
}
```

Output:

```
Microsoft Windows [Version 10.0.22000.2538]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\Admin\Desktop\bms 101>javac DisplayThreads.java  
  
C:\Users\Admin\Desktop\bms 101>java DisplayThreads  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering
```

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 Arithmetic Exception Display the exception in a message dialog box.

Algorithm: