

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



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

Submitted by

Crevan Neil Fernandes (**1BM23CS082**)

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 **Crevan Neil Fernandes (1BM23CS082)**, 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.

Geetha N 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	4
2	8/10/24	Student Class	7
3	15/10/24	Book Class	15
4	22/10/24	Abstract Shapes	21
5	29/10/2 4	Bank Program	27
6	12/11/2 4	Packages Program	39
7	19/11/24	Polygon Interface	47
8	26/11/24	WrongAge Exception program	51
9	3/12/24	Threads program	56
10	3/12/24	GUI-Java Swing	59

Github Link:

<https://github.com/Crevan-Fernandes/OOJ>

Program 1

Develop a Java program that prints all real solutions to the quadratic equation. Read a, b, c and use the quadratic formula. If the discriminant is negative, display a message stating that there are no real solutions.

Algorithm:

3. Java program for solving a quadratic equation

```
import java.util.*;
public class Quadratic{
    public static void (String [] args){
        Scanner scanner = new Scanner (System.in);
        System.out.println ("Enter the first coefficient:");
        double a = scanner.nextDouble();
        System.out.println ("Enter the second coefficient:");
        double b = scanner.nextDouble();
        System.out.println ("Enter the third coefficient:");
        double c = scanner.nextDouble();
        double d = b * b - 4 * a * c;
        if (d == 0) {
            double root1 = -b / (2 * a);
            System.out.println ("Roots are real and equal"
                + root1);
        }
    }
}
```

```

else if (d > 0) {
    double root1 = (-b + Math.sqrt(d)) / (2 * a); 
$$(-b + \sqrt{d}) / 2a$$

    double root2 = (-b - Math.sqrt(d)) / (2 * a); 
$$(-b - \sqrt{d}) / 2a$$

    System.out.println("Root 1 is: " + root1);
    System.out.println("Root 2 is: " + root2);
}

else {
    System.out.println("Roots are imaginary.");
}

scanner.close();
System.out.println("Crewan Neil Fernandes, 18M23CS082");
}

Output
double real = -b / 2 * a;
double img = Math.sqrt(-d) / 2 * a;
System.out.println("Roots are " + real + " + " + img + "i");
System.out.println("Roots are " + real + " - " + img + "i");
}

```

Scrn Scrn
off

Output

Enter the first coefficient 3
 Enter the second coefficient 7
 Enter the third coefficient 2
 Roots are : -0.33333 and -2.0

Enter the first coefficient 1
 Enter the second coefficient 2
 Enter the third coefficient 1
 Roots are : There is one real root : -1.0

QF
01/10/24

Code:

```
import java.util.*;
public class Quadratic{
public static void main(String args[]){
Scanner scanner=new Scanner(System.in);
System.out.println("Enter the first coefficient:");
double a=scanner.nextDouble();
System.out.println("Enter the second coefficient:");
double b=scanner.nextDouble();
System.out.println("Enter the third coefficient:");
double c=scanner.nextDouble();
double d=b*b-4*a*c;
if (d==0){
double root1=-b/(2*a);
System.out.println("Roots are real and equal" +root1);
}
else if (d>0){
double root1=(-b+Math.sqrt(d))/(2*a);
double root2=(-b-Math.sqrt(d))/(2*a);
System.out.println("Root 1 is:" +root1);
System.out.println("Root 2 is:" +root2);
}
else{
double real=-b/2*a;
double imag=Math.sqrt(-d)/(2*a);
System.out.println("Roots are:" + real+ "+" +imag+ "i");
System.out.println(real+ "-" +imag+ "i");
}
System.out.println("Name:Crevan Neil Fernandes");
System.out.println("USN:1BM23CS082");
}
}
```

```
C:\Users\Dell\Desktop>java Quadratic
Enter the first coefficient:
3
Enter the second coefficient:
7
Enter the third coefficient:
2
Root 1 is:-0.3333333333333333
Root 2 is:-2.0
```

```
C:\Users\Dell\Desktop>java Quadratic
Enter the first coefficient:
1
Enter the second coefficient:
2
Enter the third coefficient:
1
Roots are real and equal-1.0
```

```
C:\Users\Dell\Desktop>java Quadratic
Enter the first coefficient:
5
Enter the second coefficient:
1
Enter the third coefficient:
6
Roots are:-2.5+1.0908712114635715i
-2.5-1.0908712114635715i
```

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:

Date _____
Page _____

Write a Java program to create a class Student with members usn, name, an array credits, array marks and include methods to accept and display details

```
import java.util.*;  
class Student { /* Constructor statement  
String usn;  
String name;  
int [] credits;  
int [] marks;  
int numSubjects;  
Student (int numSubjects){  
Scanner scanner = new Scanner (System.in);  
credits = new int [numSubjects];  
marks = new int [numSubjects];  
}  
Student()  
public static void main (String [] args) {  
Scanner scanner = new Scanner (System.in);  
System.out.print ("Enter the number of subjects:");  
int numSubjects = scanner.nextInt();  
for (i=0; i < numSubjects; i++) {  
System.out.print ("Enter the usn:");  
usn = scanner.nextLine();  
System.out.print ("Enter the name:");  
name = scanner.nextLine();  
Student student = new Student();  
public void display () {  
Scanner scanner = new Scanner (System.in);  
System.out.print ("Enter the usn:");  
usn = scanner.nextLine();  
System.out.print ("Enter the name:");  
name = scanner.nextLine();
```

```

System.out.println("Enter the number of subjects:");
numSubjects = scanner.nextInt();
for (i=0; i < numSubjects; i++) {
    System.out.println("Enter credits for subject" + (i+1) +
        ":");

    credits[i] = scanner.nextInt();
    System.out.println("Enter marks for subject" + (i+1) +
        ":");

    marks[i] = scanner.nextInt();
}

public void display() {
    System.out.println("USN: " + usn);
    System.out.println("Name: " + name);
    for (i=0; i < numSubjects; i++) {
        System.out.println("Subject" + (i+1) + "-credits: " + credits[i]);
        System.out.println("Subject" + (i+1) + "-marks: " + marks[i]);
    }
}

System.out.println("SGPA")
public int sgpa(int marks) {
    if (marks >= 90) {
        return 10;
    }
    else if (marks >= 80 && marks < 90) {
        return 9;
    }
    else if (marks >= 70 && marks < 80) {
        return 8;
    }
    else if (marks >= 60 && marks < 70) {
        return 7;
    }
}

```

```

else if (marks >= 50 && marks < 60) {
    return 6;
}
else if (marks >= 40 && marks < 50) {
    return 5;
}
else {
    return 0;
}

double sgpa() {
    int totalcredits = 0;
    double totalgp = 0.0;

    for (i=0; i<n; i++) {
        int resgp = sgpa(marks[i]);
        totalgp += resgp * credits[i];
        totalcredits += credits[i];
    }
    double sgpa = totalgp / totalcredits;
    return sgpa;
}
}

public class Second {
    public static void main (String [] args) {
        Student st = new Student();
        st.input();
        st.display();
        st.sgpa();
        double res = st.sgpa();
        System.out.println ("SGPA: " + res);
    }
}

```

Page _____

*Seen
Proved*

Output

USN-434.

Enter the number of students (minimum 2): 2

Enter the number of subjects : 2

Enter details for student 1:

Enter USN: 4341BM23CS082

Enter Name: Aravan

Enter credits for subject 1: 2

Enter marks for subject 1: 80

Enter credits for subject 2: 3

Enter marks for subject 2: 70

Enter details for student 2:

* Enter USN: 1BM23CS090

Enter Name: John

Enter credits for subject 1: 1

Enter marks for subject 1: 65

Enter credits for subject 2: 3

Enter marks for subject 2: 90

Student 1 Details

USN: 1BM23CS082

Name: Aravan

Subject 1 - Credits: 2, Marks: 80

Subject 2 - Credits: 3, Marks: 70

SGPA for student 1: 8.840

Student 2 Details

USN: 1BM23CS090

→ Seen

Name: John

Subject 1 - Credits: 1, Marks: 65

Subject 2 - Credits: 3, Marks: 90

SGPA for student 2: 9.25

C/
8/10

Code:

```
import java.util.*;
class Student {
    String usn;
    String name;
    int[] credits;
    int[] marks;

    Student(int numSubjects) {
        credits = new int[numSubjects];
        marks = new int[numSubjects];
    }

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

        System.out.print("Enter the number of students (minimum 2): ");
        int numStudents = sc.nextInt();
        while (numStudents < 2) {
            System.out.print("Please enter at least 2 students: ");
            numStudents = sc.nextInt();
        }

        System.out.print("Enter the number of subjects: ");
        int numSubjects = sc.nextInt();
        sc.nextLine(); // Consume newline

        Student[] students = new Student[numStudents];

        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nEnter details for student " + (i + 1) + ":");
            students[i] = new Student(numSubjects);

            System.out.print("Enter USN: ");
            students[i].usn = sc.nextLine();

            System.out.print("Enter Name: ");
            students[i].name = sc.nextLine();

            for (int j = 0; j < numSubjects; j++) {
                System.out.print("Enter credits for subject " + (j + 1) + ": ");
                students[i].credits[j] = sc.nextInt();
                System.out.print("Enter marks for subject " + (j + 1) + ": ");
                students[i].marks[j] = sc.nextInt();
            }
            sc.nextLine(); // Consume newline after reading marks
        }
    }
}
```

```

for (int i = 0; i < numStudents; i++) {
    System.out.println("\nStudent " + (i + 1) + " Details:");
    System.out.println("USN: " + students[i].usn);
    System.out.println("Name: " + students[i].name);

    for (int j = 0; j < numSubjects; j++) {
        System.out.println("Subject " + (j + 1) + " - Credits: " + students[i].credits[j] + ", Marks: "
+ students[i].marks[j]);
    }

    int totalCredits = 0;
    double totalPoints = 0.0;

    for (int j = 0; j < numSubjects; j++) {
        int mark = students[i].marks[j];
        int gradePoint;

        if (mark >= 90) gradePoint = 10;
        else if (mark >= 80) gradePoint = 9;
        else if (mark >= 70) gradePoint = 8;
        else if (mark >= 60) gradePoint = 7;
        else if (mark >= 50) gradePoint = 6;
        else if (mark >= 40) gradePoint = 5;
        else gradePoint = 0;

        totalPoints += gradePoint * students[i].credits[j];
        totalCredits += students[i].credits[j];
    }

    double sgpa = totalCredits == 0 ? 0.0 : totalPoints / totalCredits;
    System.out.printf("\nSGPA for student %d: %.2f\n", i + 1, sgpa);
}

System.out.println("Name: Crevan Neil Fernandes");
System.out.println("USN: 1BM23CS082");
}
}

```

```
C:\Users\DELL\Desktop>java Student
Enter the number of students (minimum 2): 2
Enter the number of subjects: 4

Enter details for student 1:
Enter USN: 77
Enter Name: John
Enter credits for subject 1: 3
Enter marks for subject 1: 87
Enter credits for subject 2: 4
Enter marks for subject 2: 90
Enter credits for subject 3: 2
Enter marks for subject 3: 70
Enter credits for subject 4: 3
Enter marks for subject 4: 66

Enter details for student 2:
Enter USN: 80
Enter Name: Austin
Enter credits for subject 1: 3
Enter marks for subject 1: 80
Enter credits for subject 2: 4
Enter marks for subject 2: 92
Enter credits for subject 3: 2
Enter marks for subject 3: 77
Enter credits for subject 4: 3
Enter marks for subject 4: 80

Student 1 Details:
USN: 77
Name: John
Subject 1 - Credits: 3, Marks: 87
Subject 2 - Credits: 4, Marks: 90
Subject 3 - Credits: 2, Marks: 70
Subject 4 - Credits: 3, Marks: 66

SGPA for student 1: 8.67
```

```
Student 2 Details:  
USN: 80  
Name: Austin  
Subject 1 - Credits: 3, Marks: 80  
Subject 2 - Credits: 4, Marks: 92  
Subject 3 - Credits: 2, Marks: 77  
Subject 4 - Credits: 3, Marks: 80
```

```
SGPA for student 2: 9.17  
Name: Crevan Neil Fernandes  
USN: 1BM23CS082
```

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:

Create a 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. Write a Java program to create n books.

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

```
class Book {  
    private String name;  
    private String author;  
    private double price;  
    private int numPages;
```

```
Book (String name, String author, double price, int numPages)  
{
```

```
    this.name = name;  
    this.author = author;  
    this.price = price;  
    this.numPages = numPages;
```

```
}
```

```
void setName (String name) {  
    this.name = name;
```

```
}
```

```
void setAuthor (String author) {  
    this.author = author;
```

```
}
```

```
void setPrice (double price) {  
    this.price = price;
```

```
}
```

```
void setNumPages (int numPages) {  
    this.numPages = numPages;
```

```
}
```

```

    getName() {
        return name;
    }
    getAuthor() {
        return author;
    }
    getPrice() {
        return price;
    }
    getNumPages() {
        return numPages;
    }
}

@Override
String toString() {
    return "Book [ name = " + name + " author = " + author +
        " price = " + price + " No. of pages = " + numPages];
}
}

class BookDemo {
    public static void main (String [] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.println ("Enter the no. of books");
        int n = scanner.nextInt();
        Book [] books = new Book [n];
        ArrayList <Book> book = new ArrayList <> ();
        for (i=0; i < n; i++) {
            System.out.println ("Enter the details for book" + (i+1));
            System.out.print ("Name : ");
            String name = scanner.nextLine();
            System.out.print ("Author : ");
            String author = scanner.nextLine();
            System.out.print ("Price : ");
            double price = scanner.nextDouble();
            books[i] = new Book (name, author, price);
            book.add (books[i]);
        }
        System.out.println ("Books added successfully");
        System.out.println ("Books details are as follows");
        for (i=0; i < n; i++) {
            System.out.println ("Book " + (i+1) + " Details : " + books[i]);
        }
        System.out.println ("Books details using ArrayList");
        for (i=0; i < book.size(); i++) {
            System.out.println ("Book " + (i+1) + " Details : " + book.get(i));
        }
    }
}

```

Date _____
Page _____

```

System.out.println("No. of pages:");
int numPages = scanner.nextInt();
}

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

for (Book book : books) {
    System.out.println(book);
}
scanner.close();
}
}

```

seen execute

Output

Enter the no. of books
2

Enter the details for book 1:

Name:
Harry Potter

Author:
JK Rowling

Price:
2500

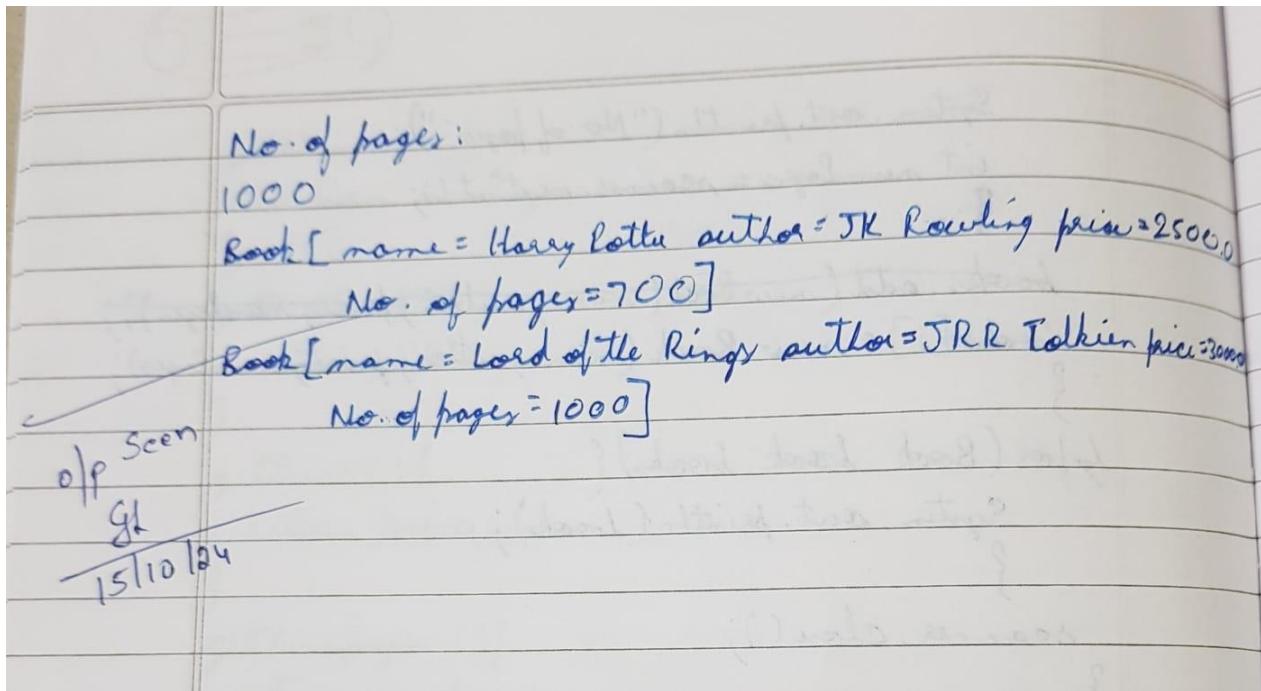
No. of pages
700

Enter the details for book 2:

Name:
Lord of the Rings

Author:
JRR Tolkien

Price:
3000



Code:

```

import java.util.*;
class Book{
    private String name;
    private String author;
    private double price;
    private int numPages;
    Book(String name, String author, double price, int numPages)
    {
        this.name=name;
        this.author=author;
        this.price=price;
        this.numPages=numPages;
    }
    void setName(String name){
        this.name=name;
    }
    void setAuthor(String author){
        this.author=author;
    }
    void setPrice(double price){
        this.price=price;
    }
    void setnumPages(int numPages){
        this.numPages=numPages;
    }
    String getName(){
        return name;
    }
}

```

```

}

String getAuthor(){
return author;
}
Double getPrice(){
return price;
}
int getnumPages(){
return numPages;
}
@Override
public String toString(){
return "Book [name= " + name+ " author= " + author + " price= " + price + " No. of pages= " +
numPages + "]";
}
}

class BookDemo{
public static void main (String[]args){
Scanner scanner=new Scanner(System.in);
System.out.println("Enter the no. of books");
int n = scanner.nextInt();
scanner.nextLine();
Book[] books= new Book[n];
for(int i=0;i<n;i++) {
System.out.println("Enter the details for book" +(i+1) +":");
System.out.println("Name:");
String name=scanner.nextLine();
System.out.println("Author:");
String author=scanner.nextLine();
System.out.println("Price:");
double price=scanner.nextDouble();
System.out.println("No. of pages:");
int numPages=scanner.nextInt();
scanner.nextLine();
books[i]=new Book(name, author, price, numPages);
}
for(Book book: books){
System.out.println(book);
}
System.out.println("Name: Crevan Neil Fernandes");
System.out.println("USN: 1BM23CS082");
scanner.close();
}
}

```

```
C:\Users\DELL\Desktop>java BookDemo
Enter the no. of books
2
Enter the details for book1:
Name:
Harry Potter
Author:
J.K Rowling
Price:
500
No. of pages:
800
Enter the details for book2:
Name:
Lord of the Rings
Author:
JRR Tolkien
Price:
1000
No. of pages:
900
Book [name= Harry Potter author= J.K Rowling price= 500.0 No. of pages= 800]
Book [name= Lord of the Rings author= JRR Tolkien price= 1000.0 No. of pages= 900]
Name: Crevan Neil Fernandes
USN: 1BM23CS082
```

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 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 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 contains only the method printArea() that prints the area of the given shape

```
import java.util.*;
abstract class Shape {
    int dim1, dim2;
    Shape (int dim1, int dim2) {
        this.dim1 = dim1;
        this.dim1 = 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 {
    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 * dim3;  
    System.out.println("Area of the Circle: " + area);  
}
```

}

```
public class Shapes {
```

```
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the length of rectangle: ");  
        int len = scanner.nextInt();  
        System.out.print("Enter the breadth of the rectangle: ");  
        int breth = scanner.nextInt();  
        System.out.print("Enter the base of the triangle: ");  
        int b = scanner.nextInt();  
        System.out.print("Enter the height of the triangle: ");  
        int h = scanner.nextInt();  
        System.out.print("Enter the radius of the circle: ");  
        int r = scanner.nextInt();  
        Shape rectangle = new Rectangle(len, breth);  
        Shape triangle = new Triangle(b, h);  
        Shape circle = new Circle(r);  
    }
```

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

{
}

{

Output

Enter the length of the rectangle

5

Enter the breadth of the rectangle

6

Enter the base of the triangle

3

Enter the height of the triangle

4

Enter the radius of the circle

7

Area of Rectangle: 30

Area of Triangle: 6.0

Area of Circle: 153.93804

Q8

22/10/24

Code:

```
import java.util.*;  
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 Shapes {
    public static void main(String[] args) {
        Scanner scanner= new Scanner(System.in);
        System.out.println("Enter the length of the rectangle");
        int len=scanner.nextInt();
        System.out.println("Enter the breadth of the rectangle");

        int breth=scanner.nextInt();
        System.out.println("Enter the base of the triangle");

        int b=scanner.nextInt();
        System.out.println("Enter the height of the triangle");

        int h=scanner.nextInt();
        System.out.println("Enter the radius of the circle ");

        int r= scanner.nextInt();
        Shape rectangle = new Rectangle(len, breth);
        Shape triangle = new Triangle(b, h);
        Shape circle = new Circle(r);

        rectangle.printArea();
        triangle.printArea();
        circle.printArea();
        System.out.println("Name:Crevan Neil Fernandes");
        System.out.println("USN:1BM23CS082");
    }
}

```

```
C:\Users\DELL\Desktop>javac Shapes.java

C:\Users\DELL\Desktop>java Shapes
Enter the length of the rectangle
4
Enter the breadth of the rectangle
5
Enter the base of the triangle
12
Enter the height of the triangle
9
Enter the radius of the circle
7
Area of Rectangle: 20
Area of Triangle: 54.0
Area of Circle: 153.93804002589985
Name:Crevan Neil Fernandes
USN:1BM23CS082
```

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:

Develop a Java program to create a class Bank that maintains two kinds of accounts 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 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) Allow withdrawal and update the balance
- Check for minimum balance, impose penalty if not necessary and update the balance
import java.util.*;

class Account {

String customer_name;

String account_number;

double balance;

String account_type)

Account & (String customer_name, String account_number,

String account_type) {

this.customer_name = customer_name;

this.account_number = account_number;

this.account_type = account_type;

this.balance = balance 0.0;

}

```

void deposit (double amount) {
    balance += amount;
    print();
    System.out.println("Amount " + amount + " has been deposited");
}

double getBalance () {
    return balance;
}

void displayBalance () {
    System.out.println("Balance is: " + balance);
}

class SavingsAccount extends Account {
    double interest_rate;

    SavingsAccount (String customer_name, String account_number,
                    double interest_rate) {
        super (customer_name, account_number, "Savings");
        this.interest_rate = interest_rate;
    }

    void calculateAndDeposit_interest (double) {
        double interest = balance * interest_rate / 100;
        deposit (interest);
        System.out.println("Interest " + interest + " has been deposited");
    }

    void withdraw (double amount) {
        if (amount > balance) {
            System.out.println("Balance insufficient");
        }
        else {
            balance -= amount;
            System.out.println(amount + " has been withdrawn");
        }
    }
}

```

Date _____
Page _____

```

class CurrentAccount extends Account {
    int minimumBalance = 1000;
    int penalty = 50;

    CurrentAccount (String customer_name, String account_number) {
        super (customer_name, account_number, "Current");
    }

    void withdraw (double amount) {
        if (amount > balance) {
            System.out.println ("Balance insufficient");
        } else {
            balance -= amount;
            checkMinimumBalance ();
            System.out.println (* amount + " has been withdrawn");
        }
    }

    void checkMinimumBalance () {
        if (balance < minimum_balance) {
            balance -= penalty;
            System.out.println ("50 has been withdrawn from balance as
penalty");
        }
    }
}

class Bank {
    public static void main (String [] args) {
        Scanner scanner = new Scanner (System.in);
        SavingsAccount savings = new SavingsAccount ();
        System.out.print ("Enter customer name for Savings Account");
        String cust = scanner.nextLine ();
        System.out.print ("Enter account number for Savings Account");
        String acc = scanner.nextLine ();
        System.out.print ("Enter interest rate for Savings Account");
        double rate = scanner.nextDouble ();
        SavingsAccount savings = new SavingsAccount (cust, acc, rate);
    }
}

```

```

System.out.print("Enter customer name for current account");
String customer = scanner.nextLine();
System.out.print("Enter account number for current account");
String num = scanner.nextLine();
CurrentAccount current = new CurrentAccount(customer, num);
System.out.print("Enter the
int i=1;
while (i==1){
    System.out.print("Enter the choice");
    int choice = scanner.nextInt();
    switch (choice){
        case 1:
            System.out.print("Enter amount to be deposited");
            double amount = scanner.nextDouble();
            savings.double deposit(amount);
            break;
        case 2:
            System.out.print("Enter the amount to be deposited");
            double amount = scanner.nextDouble();
            current.deposit(amount);
            break;
        case 3:
            savings.calculateAndDepositInterest();
            break;
        case 4:
            System.out.print("Enter the amount to be withdrawn");
            double amount = scanner.nextDouble();
            savings.withdraw(amount);
            break;
        case 5:
            System.out.print("Enter the amount to be withdrawn");
            double amount = scanner.nextDouble();
            current.withdraw(amount);
            break;
    }
}

```

case 6:
 saving.displaybalance();
 break;

case 7: current
 current.displaybalance();
 break;

* case 8:

i = 0;

break;

default:

System.out.println("Invalid choice");

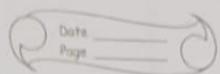
break;

}

{

{

{



~~Code~~ Output

Enter customer name for Savings Account : Augustine

Enter account number for Savings Account : 1

Enter interest rate for Savings Account : 5

Enter customer name for Current Account : Bernard

Enter account number for Current Account : 2

1. Deposit into Savings Account

2. Deposit into Current account

3. Deposit interest

4. Withdraw from Savings account

5. Withdraw from Current account

6. Display balance in Savings account

7. Display balance in Current account

8. Exit

Enter the choice

1

Enter amount to be deposited

3000

Deposited : 3000.0

Enter the choice

3

Deposited: 250

Interest of 250 deposited

Enter the choice

4

Enter the amount to be withdrawn

2950

Withdrawn: 2950.0

Minimum balance not maintain. Service charge of 50.0 imposed

Enter the choice: 5

Current balance: 3250

Enter the choice: 7

Current balance: 0.0

Enter the choice

8

Name: Caesar Neil Fernandes

USN: 1BM23CS082

Code:

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

```
class Account {  
    String customerName;  
    String accountNumber;  
    double balance;  
    String accountType;
```

```
Account(String customerName, String accountNumber, String accountType) {  
    this.customerName = customerName;
```

```

        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0.0; // Initial balance is set to zero
    }

    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }

    double getBalance() {
        return balance;
    }

    void displayBalance() {
        System.out.println("Current balance: " + balance);
    }

}

class SavingsAccount extends Account {
    private double interestRate;

    SavingsAccount(String customerName, String accountNumber, double interestRate) {
        super(customerName, accountNumber, "Savings");
        this.interestRate = interestRate;
    }

    void computeAndDepositInterest() {
        double interest = balance * interestRate / 100;
        deposit(interest);
        System.out.println("Interest of " + interest + " deposited.");
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance for withdrawal.");
        }
    }
}

class CurrentAccount extends Account {
    private static final double MINIMUM_BALANCE = 1000.0;
    private static final double SERVICE_CHARGE = 50.0;
}

```

```

CurrentAccount(String customerName, String accountNumber) {
    super(customerName, accountNumber, "Current");
}

void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrawn: " + amount);
    } else {
        System.out.println("Insufficient balance for withdrawal.");
    }
    checkMinimumBalance();
}

void checkMinimumBalance() {
    if (balance < MINIMUM_BALANCE) {
        balance -= SERVICE_CHARGE;
        System.out.println("Minimum balance not maintained. Service charge of " +
SERVICE_CHARGE + " imposed.");
    }
}
}

class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Welcome to the Bank!");

        // Create a savings account
        System.out.print("Enter customer name for Savings Account: ");
        String savCustomerName = scanner.nextLine();
        System.out.print("Enter account number for Savings Account: ");
        String savAccountNumber = scanner.nextLine();
        System.out.print("Enter interest rate for Savings Account: ");
        double interestRate = scanner.nextDouble();
        SavingsAccount savingsAccount = new SavingsAccount(savCustomerName,
savAccountNumber, interestRate);

        // Create a current account
        scanner.nextLine(); // Consume newline
        System.out.print("Enter customer name for Current Account: ");
        String currCustomerName = scanner.nextLine();
        System.out.print("Enter account number for Current Account: ");
        String currAccountNumber = scanner.nextLine();
        CurrentAccount currentAccount = new CurrentAccount(currCustomerName,
currAccountNumber);
    }
}

```

```

int i=1;
while(i==1){
    System.out.println("1. Deposit into Savings account");
    System.out.println("2. Deposit into Current account");
    System.out.println("3. Deposit interest");
    System.out.println("4. Withdraw from Savings account");
    System.out.println("5. Withdraw from Current account");
    System.out.println("6. Display balance in Savings account");
    System.out.println("7. Display balance in Current account");
    System.out.println("8. Exit");
    System.out.println("Enter the choice");
    int choice= scanner.nextInt();
    switch(choice){
        case 1:
            System.out.println("Enter amount to be deposited");
            double amt= scanner.nextDouble();
            savingsAccount.deposit(amt);
            break;
        case 2:
            System.out.println("Enter amount to be deposited");
            double a= scanner.nextDouble();
            currentAccount.deposit(a);
            break;
        case 3:
            savingsAccount.computeAndDepositInterest();
            break;
        case 4:
            System.out.println("Enter amount to be withdrawn");
            double am= scanner.nextDouble();
            savingsAccount.withdraw(am);
            break;
        case 5:
            System.out.println("Enter amount to be withdrawn");
            double amot= scanner.nextDouble();
            currentAccount.withdraw(amot);
            break;
        case 6:
            savingsAccount.displayBalance();
            break;
        case 7:
            currentAccount.displayBalance();
            break;
        case 8:
            i=0;
            break;
        default:
            System.out.println("Invalid choice");
    }
}

```

```

        break;
    }
}
System.out.println("Name: Crevan Neil Fernandes");
System.out.println("USN: 1BM23CS082");

scanner.close();
}
}

C:\Users\Dell\Desktop>javac Bank.java

C:\Users\Dell\Desktop>java Bank
Welcome to the Bank!
Enter customer name for Savings Account: David
Enter account number for Savings Account: 1
Enter interest rate for Savings Account: 5
Enter customer name for Current Account: Henry
Enter account number for Current Account: 2
1. Deposit into Savings account
2. Deposit into Current account
3. Deposit interest
4. Withdraw from Savings account
5. Withdraw from Current account
6. Display balance in Savings account
7. Display balance in Current account
8. Exit
Enter the choice
1
Enter amount to be deposited
500
Deposited: 500.0
1. Deposit into Savings account
2. Deposit into Current account
3. Deposit interest
4. Withdraw from Savings account
5. Withdraw from Current account
6. Display balance in Savings account
7. Display balance in Current account
8. Exit
Enter the choice
2
Enter amount to be deposited
3000
Deposited: 3000.0
1. Deposit into Savings account
2. Deposit into Current account
3. Deposit interest

```

```
Enter amount to be deposited
3000
Deposited: 3000.0
1. Deposit into Savings account
2. Deposit into Current account
3. Deposit interest
4. Withdraw from Savings account
5. Withdraw from Current account
6. Display balance in Savings account
7. Display balance in Current account
8. Exit
Enter the choice
3
Deposited: 25.0
Interest of 25.0 deposited.
1. Deposit into Savings account
2. Deposit into Current account
3. Deposit interest
4. Withdraw from Savings account
5. Withdraw from Current account
6. Display balance in Savings account
7. Display balance in Current account
8. Exit
Enter the choice
5
Enter amount to be withdrawn
2500
Withdrawn: 2500.0
Minimum balance not maintained. Service charge of 50.0 imposed.
1. Deposit into Savings account
2. Deposit into Current account
3. Deposit interest
4. Withdraw from Savings account
5. Withdraw from Current account
6. Display balance in Savings account
7. Display balance in Current account
8. Exit
Enter the choice
6
Current balance: 525.0
```

```
Enter the choice
6
Current balance: 525.0
1. Deposit into Savings account
2. Deposit into Current account
3. Deposit interest
4. Withdraw from Savings account
5. Withdraw from Current account
6. Display balance in Savings account
7. Display balance in Current account
8. Exit
Enter the choice
7
Current balance: 450.0
1. Deposit into Savings account
2. Deposit into Current account
3. Deposit interest
4. Withdraw from Savings account
5. Withdraw from Current account
6. Display balance in Savings account
7. Display balance in Current account
8. Exit
Enter the choice
8
Name: Crevan Neil Fernandes
USN: 1BM23CS082
```

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:

Create a package CIE which has two classes - Student and Internals. The class student has members like name, marks, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
package CIE;
```

```
public class Student {
```

```
    public String name;
```

```
    public String marks;
```

```
    public int sem;
```

```
}
```

```
package CIE;
```

```
public class Internals extends Student {
```

```
    public int[] internalMarks = new int[5];
```

```
    public void setInternalMarks(int[] marks) {
```

```
        for (int i = 0; i < 5; i++) {
```

```
            internalMarks[i] = marks[i];
```

```
}
```

```
    public int[] getInternalMarks() {
```

```
        return internalMarks;
```

```
}
```

```
}
```

~~package SEE;~~~~import CIE.Student;~~~~public class External extends Student {~~ ~~public int[] reMarks = new int[5];~~

```

public void setSEEMarks (int [] marks) {
    for (int i = 0; i < 5; i++) {
        seeMarks[i] = marks[i];
    }
}

public int [] getSEEMarks () {
    return seeMarks;
}

import CIE.*;
import SEE.*;
import java.util.Scanner;

public class FinalMarks {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the number of students:");
        int n = sc.nextInt ();
        Student [] students = new Student [n];
        Internal [] internalMarks = new Internal [n];
        for (int i = 0; i < n; i++) {
            students[i] = new Student ();
            internalMarks[i] = new Internal ();
            seeMarks[i] = new External ();
            System.out.println ("Enter USN for student " + (i+1) + ":");
            students[i].usn = sc.next ();
            sc.nextLine ();
            System.out.println ("Enter Name for Student " + (i+1) + ":");
            students[i].name = sc.nextLine ();
            int [] internalMarks = new int [5];
            System.out.println ("Enter Internal Marks for Student " + (i+1));
        }
    }
}

```

```

for (int j=0; j<5; j++) {
    internMarks[j] = sc.nextInt();
}
internalMarks[i].setInternalMarks(internalMarks);
int[] see = new int[5];
System.out.println("Enter SEE marks for Student");
for (int j=0; j<5; j++) {
    see[j] = sc.nextInt();
}
seeMarks[i].setSEEMarks(see);
}
System.out.println("In Final Marks of Students:");
for (int i=0; i<n; i++) {
    System.out.println("In Student "+(i+1)+": "+student[i].usn
        + " (USN: "+student[i].usn+")");
}
System.out.println("Course "+internalMarks[i].getInternalMarks()[j]+
    seeMarks[i].getSEEMarks()[j]);
System.out.println("Course "+(j+1)+": "+internalMarks[i].
    getInternalMarks()[j]+ " "+ seeMarks[i].
    getSEEMarks()[j]+ " "+ finalMark);
}
}
sc.close();
}

```

~~Output~~

Enter the number of students: 2

Enter USN for Student 1: 1

Enter name for Student 1: Sean

Enter Semester for Student 1: 2

Enter Internal Marks for Student 1:

33 35 31 32 38

Enter SEE Marks for Student 1:

90 95 88 85 87

Enter USN for Student 2: 2

Enter Name for Student 2: James

Enter Semester for Student 2: 3 3

Enter Internal Marks for Student 2:

35 37 40 30 33

Enter SEE Marks for Student 2:

99 100 95 96 92

Final Marks of Students:

Student 1: Sean (USN: 1)

Course	Internal	SEE	Final Marks
Course 1:	33	90	123
Course 2:	35	95	129
Course 3:	31	88	119
Course 4:	32	85	117
Course 5:	38	87	123

Student 2: James (USN: 2)

Course	Internal	SEE	Final Marks
Course 1:	35	99	134
Course 2:	37	100	137
Course 3:	40	95	135
Course 4:	30	96	126
Course 5:	33	92	125

Seen
10/10/24

Code:

```
package CIE;
```

```
public class Student {  
    public String usn;  
    public String name;  
    public int sem;  
}
```

```
package CIE;
```

```
public class Internals extends Student {  
    public int[] internalMarks = new int[5];  
  
    public void setInternalMarks(int[] marks) {  
        for (int i = 0; i < 5; i++) {  
            internalMarks[i] = marks[i];  
        }  
    }  
}
```

```
public int[] getInternalMarks() {  
    return internalMarks;  
}  
}  
package SEE;
```

```
import CIE.Student;
```

```
public class External extends Student {  
    public int[] seeMarks = new int[5];  
  
    public void setSEEMarks(int[] marks) {  
        for (int i = 0; i < 5; i++) {  
            seeMarks[i] = marks[i];  
        }  
    }  
}
```

```
public int[] getSEEMarks() {  
    return seeMarks;  
}  
}
```

```
import CIE.*;  
import SEE.*;
```

```
import java.util.Scanner;
```

```

public class FinalMarks {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();

        Student[] students = new Student[n];
        Internals[] internalMarks = new Internals[n];
        External[] seeMarks = new External[n];

        for (int i = 0; i < n; i++) {
            students[i] = new Student();
            internalMarks[i] = new Internals();
            seeMarks[i] = new External();

            System.out.print("Enter USN for Student " + (i + 1) + ": ");
            students[i].usn = sc.nextInt();
            sc.nextLine(); // Consume the newline left by next()

            System.out.print("Enter Name for Student " + (i + 1) + ": ");
            students[i].name = sc.nextLine();

            System.out.print("Enter Semester for Student " + (i + 1) + ": ");
            students[i].sem = sc.nextInt();

            int[] internals = new int[5];
            System.out.println("Enter Internal Marks (5 courses) for Student " + (i + 1) + ": ");
            for (int j = 0; j < 5; j++) {
                internals[j] = sc.nextInt();
            }
            internalMarks[i].setInternalMarks(internals);

            int[] see = new int[5];
            System.out.println("Enter SEE Marks (5 courses) for Student " + (i + 1) + ": ");
            for (int j = 0; j < 5; j++) {
                see[j] = sc.nextInt();
            }
            seeMarks[i].setSEEMarks(see);
        }

        System.out.println("\nFinal Marks of Students:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1) + ": " + students[i].name + " (USN: " +
                students[i].usn + ")");
            System.out.println("Course\tInternal\tSEE\tFinal Marks");
            for (int j = 0; j < 5; j++) {

```

```

        int finalMark = internalMarks[i].getInternalMarks()[j] + seeMarks[i].getSEEMarks()[j];
        System.out.println("Course " + (j + 1) + ":\t" + internalMarks[i].getInternalMarks()[j] +
"\t\t" + seeMarks[i].getSEEMarks()[j] + "\t" + finalMark);
    }
}
sc.close();
}
}

```

```

C:\Users\Dell\Desktop\Studentmarks>javac CIE\Student.java

C:\Users\Dell\Desktop\Studentmarks>javac CIE\Internals.java

C:\Users\Dell\Desktop\Studentmarks>javac SEE\External.java

C:\Users\Dell\Desktop\Studentmarks>javac FinalMarks.java

C:\Users\Dell\Desktop\Studentmarks>java FinalMarks
Enter the number of students: 2
Enter USN for Student 1: 5
Enter Name for Student 1: Saul
Enter Semester for Student 1: 2
Enter Internal Marks (5 courses) for Student 1:
35
37
34
33
31
Enter SEE Marks (5 courses) for Student 1:
90
97
88
89
94
Enter USN for Student 2: 6
Enter Name for Student 2: Paul
Enter Semester for Student 2: 2
Enter Internal Marks (5 courses) for Student 2:
39
35
36
38
32
Enter SEE Marks (5 courses) for Student 2:
98
95
94

```

Enter SEE Marks (5 courses) for Student 2:

98
95
94
96
92

Final Marks of Students:

Student 1: Saul (USN: 5)

Course	Internal	SEE	Final Marks
Course 1:	35	90	125
Course 2:	37	97	134
Course 3:	34	88	122
Course 4:	33	89	122
Course 5:	31	94	125

Student 2: Paul (USN: 6)

Course	Internal	SEE	Final Marks
Course 1:	39	98	137
Course 2:	35	95	130
Course 3:	36	94	130
Course 4:	38	96	134
Course 5:	32	92	124

Program 7

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:

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

```
import java.util.*;
interface Polygon{
    default void getPerimeter(int sides[]){
        double perimeter=0;
        if(sides.length==1){
            perimeter=2*3.14*sides[0];
        }
        else
            for(int i=0; i<sides.length; i++){
                perimeter+=sides[i];
            }
        System.out.println("Perimeter of the polygon is:" +perimeter);
    }
    void getArea(double a);
}

class Circle implements Polygon{
    public static void main(String args){
        public void getArea(double a){
            System.out.println("Area of circle is:" + (a*a*3.14));
        }
    }
}

class Square implements Polygon{
    public void getArea(double length){
        System.out.println("Area of rectangle is:" + (length*length));
    }
}

class Main{}
```

```
public static void main (String [] args) {  
    Circle c1 = new Circle();  
    int m[] = {5};  
    c1.getPerimeter (m);  
    c1.getArea (5);  
  
    Square s1 = new Square();  
    int sides[] = {5, 5, 5, 5};  
    s1.getPerimeter (sides);  
    s1.getArea (5);
```

Output

Centre of the polygon is: 31.41593

~~Area of circle is: 78.539825~~

~~Perimeter of the polygon is: 20.0~~

~~Area of rectangle is : 25.0~~

~~191 u 124~~

Code:

```
import java.util.*;  
interface Polygon{  
    default void getPerimeter(int sides[]){
```

double perimeter=0;

```
if(sides.length==1){
```

```
perimeter=2*3.141593*sides[0];
```

}

,
else

```
for(int i=0;i<sides.length;i++){
```

```

        perimeter+=sides[i];
    }

    System.out.println("Perimeter of the polygon is:" + perimeter);
}

void getArea(double a);
}

class Circle implements Polygon{

    public void getArea(double r){
        System.out.println("Area of circle is: " +(r*3.141593*r));
    }
}

class Square implements Polygon{

    public void getArea(double length){
        System.out.println("Area of rectangle is: " +(length*length));
    }
}

class Main{
    public static void main(String [] args){
        Circle c1= new Circle();
        int m[]={5};
        c1.getPerimeter(m);
        c1.getArea(5);
        Square r1= new Square();
        int sides[]={5,5,5,5};
        r1.getPerimeter(sides);
        r1.getArea(5);
        System.out.println("Name:Crevan Neil Fernandes");
        System.out.println("USN:1BM23CS082");
    }
}

```

```
C:\Users\DELL\Desktop>javac Main.java  
C:\Users\DELL\Desktop>java Main  
Perimeter of the polygon is:31.41593  
Area of circle is: 78.539825  
Perimeter of the polygon is:20.0  
Area of rectangle is: 25.0  
Name:Crevan Neil Fernandes  
USN:1BM23CS082
```

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:

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 \geq father's age

```
import java.util.*;  
class WrongAge extends Exception{}  
public static void main (String args []){  
    public WrongAge extends (String message){  
        super (message);  
    }  
    class Father{  
        int age;  
        public Father (int age) throws WrongAge {  
            if (age < 0){  
                throw new WrongAge ("Age of the father cannot be negative");  
            }  
            this . age = age;  
            System.out.println ("The age of the father is " + this . age);  
        }  
        class Son extends Father{  
            int sage;  
            public Son (int age, int sage) throws WrongAge {  
                super (age);  
                if (sage  $\geq$  age){  
                    throw new WrongAge ("Age of the son cannot be greater than or equal to father's age");  
                }  
            }  
        }  
    }  
}
```

```

        throw new WrongAge("son cannot be older than the father");
    }

    this.sage = sage;
    System.out.println("The son's age is " + this.sage);
}

public class Check {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        try {
            System.out.print("Enter the age of the Father : ");
            int a = sc.nextInt();
            Father f = new Father(a);
        }
        catch (WrongAge e) {
            System.out.println(e.getMessage());
        }
        try {
            System.out.print("Enter the father's age : ");
            int b = sc.nextInt();
            System.out.print("Enter the son's age : ");
            int c = sc.nextInt();
            Son s = new Son(b, c);
        }
        catch (WrongAge e) {
            System.out.println(e.getMessage());
        }
    }
}

```

Output

Enter the age of the Father:

-5

Age of the father cannot be negative
 Enter the father's age:
 40
 Enter the son's age:
 45
~~The age of the Father is 40~~
~~Son cannot be older than the father~~

Case 2:
 Enter the age of the Father:
~~20~~
~~The age of the Father is 20~~
 Enter the father's age:
 43
 Enter the son's age:
 22
~~The age of the Father is 43~~
~~The son's age is 22~~

Code:

```

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 of the father cannot be negative");
}
this.age=age;
System.out.println("The age of the Father is "+ this.age);
}
}
class Son extends Father{
    
```

```

int sage;
public Son(int age, int sage) throws WrongAge{
super(age);
if(sage>=age){
throw new WrongAge("Son cannot be older than the father");
}
this.sage=sage;
System.out.println("The son's age is "+this.sage);
}
}

public class Check{
public static void main(String[]args){
Scanner sc=new Scanner(System.in);
try{
System.out.println("Enter the age of the Father: ");
int a=sc.nextInt();
Father f=new Father(a);
}
catch(WrongAge e){
System.out.println(e.getMessage());
}
try{
System.out.println("Enter the father's age: ");
int b=sc.nextInt();
System.out.println("Enter the son's age: ");
int c= sc.nextInt();
Son s= new Son(b,c);
}
catch(WrongAge e){
System.out.println(e.getMessage());
}
System.out.println("Name:Crevan Neil Fernandes");
System.out.println("USN:1BM23CS082");
}
}

```

```
C:\Users\DELL\Desktop>javac Check.java

C:\Users\DELL\Desktop>java Check
Enter the age of the Father:
-5
Age of the father cannot be negative
Enter the father's age:
40
Enter the son's age:
45
The age of the Father is 40
Son cannot be older than the father
Name:Crevan Neil Fernandes
USN:1BM23CS082
```

```
C:\Users\DELL\Desktop>javac Check.java

C:\Users\DELL\Desktop>java Check
Enter the age of the Father:
45
The age of the Father is 45
Enter the father's age:
50
Enter the son's age:
25
The age of the Father is 50
The son's age is 25
Name:Crevan Neil Fernandes
USN:1BM23CS082
```

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:

Lab program

- 1) 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 MyThread extends Thread {  
    String message;  
    int interval;  
    public MyThread (String message, int interval) {  
        this.message = message;  
        this.interval = interval;  
    }  
    public void run () {  
        try {  
            while (true) {  
                System.out.println (message);  
                Thread.sleep (interval);  
            }  
        }  
        catch (InterruptedException e) {  
            System.out.println (e);  
        }  
    }  
}
```

```

class Demo{
    public static void main (String args[]){
        Thread t1 = new Mythread ("BMS College of Engineering", 10000);
        Thread t2 = new Mythread ("CSE", 2000);
        t1.start();
        t2.start();
    }
}

```

Output

BMS College of Engineering

CSE

CSE

CSE

CSE

~~BMS~~ CSE

~~BMS College of Engineering~~

seen

10/

03/12/15

Code:

```

class Mythread extends Thread{
    String message;
    int interval;
    public Mythread(String message,int interval){
        this.message=message;
        this.interval=interval;
    }
    public void run(){
        try{
            while(true){
                System.out.println(message);
                Thread.sleep(interval);
            }
        }
        catch(InterruptedException e){
    
```

```

System.out.println(e);
}
}
}
}
class Demo{
public static void main(String args[]){
Thread t1=new Mythread("BMS College of Engineering",10000);
Thread t2=new Mythread("CSE", 2000);
t1.start();
t2.start();
System.out.println("Name:Crevan Neil Fernandes");
System.out.println("USN:1BM23CS082");
}
}

```

C:\Users\Dell\Desktop>javac Demo.java

```

C:\Users\Dell\Desktop>java Demo
Name:Crevan Neil Fernandes
USN:1BM23CS082
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
^C

```

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:

```

1 Let program
2 import java.awt.*;
import java.awt.event.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
3
4 public class DivisionApp extends JFrame{
5     private class DivisionApp JTextField num1Field, num2Field, resultField;
6     private JButton divideButton;
7
8     public DivisionApp(){
9
10 }
11
12 }

```

```
setTitle("Integer Division App");
setLayout(new FlowLayout());
setSize(300, 200);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JLabel num1Label = new JLabel("Num 1:");
num1Field = new JTextField(10);
JLabel num2Label = new JLabel("Num 2:");
num2Field = new JTextField(10);
JLabel resultLabel = new JLabel("Result:");
resultField = new JTextField(10);
resultField.setEditable(false);
```

```
divideButton = new JButton("Divide");
add(num1Label);
add(num1Field);
add(num2Label);
add(num2Field);
add(divideButton);
add(resultLabel);
add(resultField);
```

divideButton.addActionListener(new ActionListener() {

```
{  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        try {  
            int num1 = Integer.parseInt(num1Field.getText());  
            int num2 = Integer.parseInt(num2Field.getText());  
            int result = num1 / num2;  
            resultField.setText(String.valueOf(result));  
        } catch (NumberFormatException ex) {  
            // Handle exception  
        }  
    }  
}
```

```
catch(NumberFormatException ex) {
```

```

JOptionPane.showMessageDialog(DivisionApp.this, "Please
enter valid integers.", "Input Error", JOptionPane.ERROR_MESSAGE);
}
catch (ArithmeticException ex) {
    JOptionPane.showMessageDialog(DivisionApp.this, "Division by zero
is not allowed.", "Arithmetic Error", JOptionPane.ERROR_MESSAGE);
}
}
}

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

```

Output

~~Integer Division~~

Num1:	<input type="text" value="6"/>	Num2:	<input type="text" value=""/>
Seen	<input type="text" value="3"/>	<input type="button" value="Divide"/>	Result <input type="text" value="2"/>
TG 6/12/21			

Code:

```
import java. swing.*;
import java awt. *
import java awt.event.ActionEvent;
import java awt.event.ActionListener;
public class Division App extends JFrame {
private Jtextfield num1Field, num2Field, resultField;
private JButton divideButton;
public DivisionApp(){
setTitle ("Integer Division App");
setLayout (new FlowLayout ());
setSize (300, 200);
setDefaultCloseOperation (JFrame. EXIT_ON_CLOSE); JLabel num1Label1
= new JLabel ("Num 1:");
num1Field= new JTextField (10);

JLabel num2Label = new JLabel ("Num 2: ");
num2Field = new JTextField (10);
JLabel resultLabel = new JLabel ("Result: "); resultField = new JTextField (10);
resultField.setEditable(false);
divideButton = new JButton ("Divide");
add (num1Label);
add (num1Field);
add (num2Label);
add (num2Field);
add (divideButton);
add (resultLabel);
add (resultfield);
divide Button.addActionListener (new ActionListener () {
@Override
public void actionPerformed (ActionEvent e) {
try {

int num1 = Integer. parseInt (num1Field.getText ());
int num2 = Integer. parseInt (num2Field.getText ());
int result = num1/num2;
resultField.setText (String. valueOf (result));
} catch (NumberFormatException ex) {
 JOptionPane.showMessageDialog (DivisionApp. this, "Please enter valid integers.", "Input Error",
JOptionPane. ERROR_MESSAGE);
}
} catch (ArithmaticException ex) {
JOptionPane.showMessageDialog (DivisionApp. this, "Division by zero is not allowed.", "Arithmatic
Error", JOptionPane. ERROR_MESSAGE);
}
}
});
```

```
}
```

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

```
    Swing Utilities. invokeLater (new Runnable () {
```

```
        @Override
```

```
        public void run(){
```

```
            new Division App ()setVisible(true);
```

```
        }
```

```
    );
```

```
}
```

```
}
```

