

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



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

*Submitted by*

Kushal Naidu N (**24BECS408**)

*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 **Kushal Naidu N (24BECS408)**, 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.

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

## Index

<b>Sl. No.</b>	<b>Date</b>	<b>Experiment Title</b>	<b>Page No.</b>
1	30/09/2024	Quadratic Equation	04-09
2	07/10/2024	SGPA calculation	10-14
3	14/10/2024	Book Details	15-19
4	21/10/2024	Area of shapes using abstract methods	20-24
5	28/10/2024	Bank class using Inheritance	25-37
6	12/11/2024	Packages	38-46
7	28/11/2024	User defined Exceptions	47-51
8	28/11/2024	Multithreading	52-55
9	28/11/2024	User Interface for division program	56-61
10	30/11/2024	Demonstration of IPC and Deadlock	62-73

Github Link:

<https://github.com/KushalBMSCE/ooj-lab-programs>

### 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:

30/9/24

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

→ Pgm 1 : Quadratic Expression  
WAP that prints all real solutions to the quad eqn  $ax^2+bx+c=0$   
Read  $a, b \& c$  & use the quad formula. If the discriminant  
 $b^2 - 4ac$  is ' $-$ 've, display a msg stating that there are no real

```
import java.util.Scanner;  
  
class Equation{  
    int a=0, b, c;  
    double r1, r2, d;  
    void add(){  
        Scanner s = new Scanner(System.in);  
        while (a==0){  
            System.out.println("Enter coefficient of a :");  
            a = s.nextInt();  
            if (a==0){  
                System.out.println("No possible solution when a is 0");  
            }  
        }  
        System.out.println("Enter coefficient of b :");  
        b = s.nextInt();  
        System.out.println("Enter coefficient of c :");  
        c = s.nextInt();  
        d = (b*b) - (4*a*c);  
        if (d==0){  
            r1 = (-b) / (2*a);  
            System.out.println("Roots are real & equal i.e " + r1);  
        }  
        else if (d>0){  
            r1 = ((-b)+(Math.sqrt(d))) / (double)(2*a);  
            r2 = ((-b)-(Math.sqrt(d))) / (double)(2*a);  
            System.out.println("Roots are real");  
        }  
    }  
}
```

```

System.out.println("Root 1" + r1);
System.out.println("Root 2" + r2);
}
else {
    r1 = (-b) / (2 * a);
    r2 = Math.sqrt(-d) / (2 * a);
    System.out.println("Roots are real & imaginary i.e.");
    System.out.println("Root 1" + r1);
    System.out.println("Root 2" + r2);
}
}
}

```

```

public class QuadEq {
public static void main(String args[]) {
    Equation e1 = new Equation();
    e1.ged();
    System.out.println("Kushal Naidu N");
    System.out.println("USN : 24BEC5408");
}
}

```

~~Output:~~

Enter coefficient of a : 3  
 Enter coefficient of b : 6  
 Enter coefficient of c : 9  
 Roots are real & imaginary i.e,  
 Root 1 : -1.0  
 Root 2 : 1.414913569373095  
 Kushal Naidu .N.  
 USN: 24BEC5408.

Enter coefficient of  $a$ : 1  
 $a = 1$        $b = 8$   
 $c = 4$

Roots are real

Root 1: -0.535898384

Root 2: -7.46410168137754

Enter coefficient of  $a$ : 0

No possible solution when  $a$  is 0

BT

~~(10/10)~~ ~~30/10/24~~

Code:

```
import java.util.Scanner;
class Equation{
    int a=0,b,c;
    double r1,r2,d;
    void getd(){
        Scanner s=new Scanner(System.in);
        while (a==0){
            System.out.println("Enter coefficient of a");
            a=s.nextInt();
            if (a==0){
                System.out.println("No possible solutions when a is 0");
            }
        }
        System.out.println("Enter coefficient of b");
        b=s.nextInt();
        System.out.println("Enter coefficient of c");
        c=s.nextInt();
        d=(b*b)-(4*a*c);
        if (d==0){
            r1=(-b)/(2*a);
            System.out.println("Roots are real and equal i.e root 1 and root 2
is"+ r1);
        } else if (d>0){
            r1=(((-b) + (Math.sqrt(d)))/(double)(2*a));
            r2=(((-b) - (Math.sqrt(d)))/(double)(2*a));
            System.out.println("Roots are real ");
            System.out.println("Root 1 "+r1);
            System.out.println("Root 2 "+r2);
        } else if(d<0){
            r1=(-b)/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println("Roots are real and imaginary i.e ");
            System.out.println("Root 1 : "+ r1);
            System.out.println("Root 2 : "+ r2);
        }
    }
}
public class Quadeq{
    public static void main(String args[]){

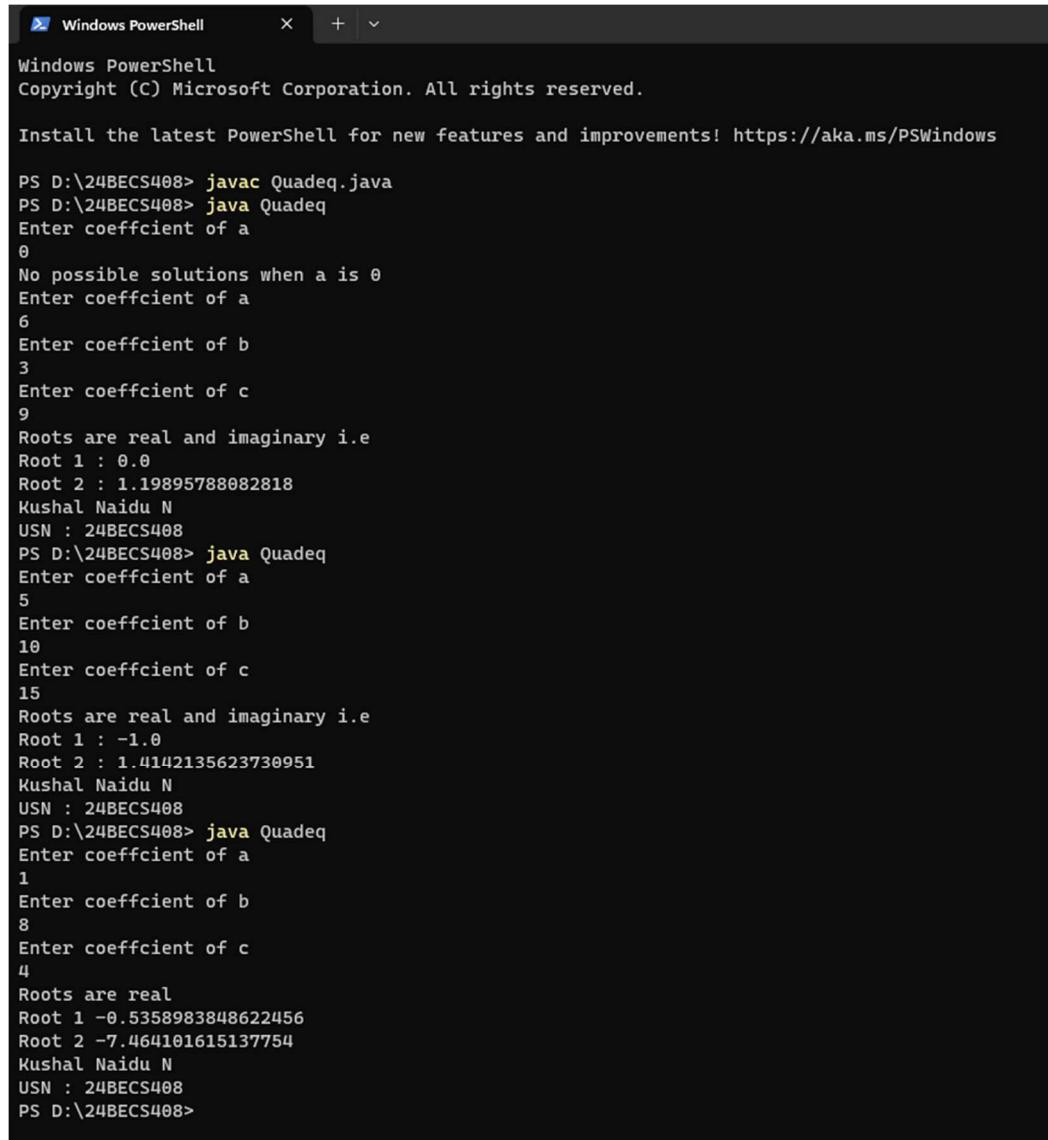
```

```

        Equation e1=new Equation();
        e1.getd();
        System.out.println("Kushal Naidu N");
        System.out.println("USN : 24BECS408");
    }
}

```

## Output



The screenshot shows a Windows PowerShell window titled "Windows PowerShell". The session starts with the standard PowerShell welcome message. It then executes three separate Java commands. The first command, "javac Quadeq.java", compiles a Java program named Quadeq. The second command, "java Quadeq", runs the program with coefficients a=0, b=6, c=3, which results in two complex roots: -1.0 + 1.4142135623730951i and -1.0 - 1.4142135623730951i. The third command, "java Quadeq" again, runs the program with coefficients a=1, b=8, c=4, resulting in two real roots: -0.5358983848622456 and -7.464101615137754.

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\24BECS408> javac Quadeq.java
PS D:\24BECS408> java Quadeq
Enter coefficient of a
0
No possible solutions when a is 0
Enter coefficient of a
6
Enter coefficient of b
3
Enter coefficient of c
9
Roots are real and imaginary i.e
Root 1 : 0.0
Root 2 : 1.19895788082818
Kushal Naidu N
USN : 24BECS408
PS D:\24BECS408> java Quadeq
Enter coefficient of a
5
Enter coefficient of b
10
Enter coefficient of c
15
Roots are real and imaginary i.e
Root 1 : -1.0
Root 2 : 1.4142135623730951
Kushal Naidu N
USN : 24BECS408
PS D:\24BECS408> java Quadeq
Enter coefficient of a
1
Enter coefficient of b
8
Enter coefficient of c
4
Roots are real
Root 1 -0.5358983848622456
Root 2 -7.464101615137754
Kushal Naidu N
USN : 24BECS408
PS D:\24BECS408>

```

```
Windows PowerShell      X + ▾
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\BMSCE\Desktop> javac QuadEq.java
PS C:\Users\BMSCE\Desktop> java QuadEq
Error: Could not find or load main class QuadEq
Caused by: java.lang.NoClassDefFoundError: QuadEq (wrong name: QuadEq)
PS C:\Users\BMSCE\Desktop> javac QuadEq.java
PS C:\Users\BMSCE\Desktop> java QuadEq
Enter coeffcient of a

0
No possible solutions when a is 0
Enter coeffcient of a
2
Enter coeffcient of b
4
Enter coeffcient of c
2
Roots are real and equal i.e root 1 and root 2 is -1.0
Kushal Naidu N
USN : 24BEC5408
PS C:\Users\BMSCE\Desktop> java QuadEq
Enter coeffcient of a
1
Enter coeffcient of b
-3
Enter coeffcient of c
2
Roots are real
Root 1 2.0
Root 2 1.0
Kushal Naidu N
USN : 24BEC5408
PS C:\Users\BMSCE\Desktop> java QuadEq
Enter coeffcient of a
1
Enter coeffcient of b
2
Enter coeffcient of c
5
Roots are real and imaginary i.e.
Root 1 : -1.0
Root 2 : 2.0
Kushal Naidu N
USN : 24BEC5408
PS C:\Users\BMSCE\Desktop> |
```

## 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:

4/10/24

2. Java Program to create a class Student with members usn, name, an array credits & an array marks. Include methods to accept & display details & a method to calculate SGPA of a student.

→ Main.java

```
import java.util.Scanner;
class Subject{
    int marks, credits, grade;
}
class Student{
    Subject subject[];
    String name, usn;
    double SGPA;
    Scanner s = new Scanner(System.in);
    Student(){
        int i;
        subject = new Subject[9];
        for(i=0; i<9; i++){
            subject[i] = new Subject();
        }
    }
    void getStudentDetails(){
        System.out.println("Enter Student name : ");
        this.name = s.nextLine();
        System.out.println("Enter Student usn : ");
        this.usn = s.nextLine();
    }
    void getMarks(){
        for(int i=0; i<9; i++){
            System.out.print("Enter marks of subject " + (i+1) + ": ");
            subject[i].credits = s.nextInt();
            subject[i].grade = (subject[i].marks / 10) + 1;
            if(subject[i].grade == 11)
                subject[i].grade = 10;
            else if(subject[i].grade == 4)
                subject[i].grade = 4;
        }
    }
}
```

```
void computeSGPA() {
    double points = 0;
    double totalcredits = 0;
    for (int i = 0; i < 9; i++) {
        int sub = subject[i].credits * subject[i].grade;
        points = points + sub;
        totalcredits = totalcredits + subject[i].credits;
    }
    SGPA = points / totalcredits;
    System.out.println("SGPA of the student is: " + SGPA);
}
```

```
public class Main {
    public static void main (String args[]) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        System.out.println("Name : " + s1.name);
        System.out.println("USN : " + s1.usn);
        s1.computeSGPA();
    }
}
```

55  
19/10/24

(10)  
/10

Code:

```
import java.util.Scanner;
class Subject{
    int marks, credits, grade;
}

class Student{
    Subject subject[];
    String name, usn;
    double SGPA;
    Scanner s=new Scanner(System.in);
    Student(){
        int i;
        subject = new Subject[9];
        for(i=0;i<9;i++)
            {subject[i] = new Subject();}
    }
    void getStudentDetails() {
        System.out.println("Enter name :");
        this.name=s.nextLine();
        System.out.println("Enter usn :");
        this.usn=s.nextLine();
    }
    void getMarks() {
        for(int i=0;i<9;i++) {
            System.out.println("Enter marks of subject "+(i+1)+":");
            subject[i].marks=s.nextInt();
            System.out.println("Enter credits of subject "+(i+1)+":");
            subject[i].credits=s.nextInt();
            subject[i].grade=(subject[i].marks/10)+1;
            if (subject[i].grade==11)
                subject[i].grade=10;
            else if(subject[i].grade==4)
                subject[i].grade=0;
        }
    }
    void computeSGPA(){
        double points=0;
        double totalcredits=0;
```

```

        for (int i=0;i<9;i++) {
            int sub=subject[i].credits*subject[i].grade;
            points=points+sub;
            totalcredits=totalcredits+subject[i].credits;
        }
        SGPA=points/totalcredits;
        System.out.println("SGPA of the student is : "+SGPA);
    }
}

public class Main{
    public static void main(String args[]) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        System.out.println("Name : "+s1.name);
        System.out.println("USN : "+s1.usn);
        s1.computeSGPA();
        System.out.println("Name: Kushal Naidu N \nUSN: 24BECS408");
    }
}

```

## Output

```
PS D:\SGPA> javac Main.java
PS D:\SGPA> java Main
Enter name :
Kushal
Enter usn :
24BECS408
Enter marks of subject 1:
65
Enter credits of subject 1:
4
Enter marks of subject 2:
78
Enter credits of subject 2:
3
Enter marks of subject 3:
45
Enter credits of subject 3:
4
Enter marks of subject 4:
73
Enter credits of subject 4:
4
Enter marks of subject 5:
84
Enter credits of subject 5:
3
Enter marks of subject 6:
95
Enter credits of subject 6:
4
Enter marks of subject 7:
68
Enter credits of subject 7:
2
Enter marks of subject 8:
96
Enter credits of subject 8:
4
Enter marks of subject 9:
84
Enter credits of subject 9:
3
Name : Kushal
USN : 24BECS408
SGPA of the student is : 8.129032258064516
PS D:\SGPA> |
```

### 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:

1) Book

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 & get the details of the objects. Include a `toString()` method that could display the details.

→ Main.java

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

```

public class Main {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        int n, price, numPages;
        String name, author;
        System.out.println("Enter number of Books : ");
        n = s.nextInt();
        Books b[];
        b = new Books[n];
        for (int i=0; i<n; i++) {
            System.out.println("\nEnter Book " + (i+1) + " Details");
            System.out.println("Enter book name : ");
            name = s.next();
            System.out.println("Enter book author : ");
            author = s.next();
            System.out.println("Enter book price : ");
            price = s.nextInt();
            System.out.println("Enter number of pages : ");
            numPages = s.nextInt();
            b[i] = new Books(name, author, price, numPages);
        }
        for (int j=0; j<n; j++) {
            System.out.println("\nBook " + (j+1) + " Details");
            System.out.println(b[j].toString());
        }
    }
}

```

10/10

Output:

Enter Book name : Java	Book name : Java
Enter Book author : Ramesh	Book author : Ramesh
Enter Book price : 955	Book price : 955
Enter number of pages : 798	Number of pages : 798
	Kushal Naidu.N
	24 BECS 408

Code:

```
import java.util.Scanner;
class Books{
    String name;
    String author;
    int price;
    int numPages;
    Books(String name, String author, int price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString()
    {
        String name,author,price,numPages;
        name = "Book name: " + this.name + "\n";
        author = "Author name: " + this.author + "\n";
        price = "Price: " + this.price + "\n";
        numPages = "Number of pages: " + this.numPages + "\n";
        return name + author + price + numPages;
    }
}
public class Main{
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        int n, price, numPages;
        String name, author;
        System.out.println("Enter number of Books : \n");
        n = s.nextInt();
        Books b[];
        b=new Books[n];
        for(int i=0;i<n;i++){
            System.out.println("\n Enter Book "+(i+1)+" Details");
            System.out.println("Enter book name :");
            name = s.next();
            System.out.println("Enter book author :");
            author = s.next();
        }
    }
}
```

```
        System.out.println("Enter book price :");
        price = s.nextInt();
        System.out.println("Enter number of pages :");
        numPages = s.nextInt();
        b[i]=new Books(name,author,price,numPages);
    }
    for(int j=0;j<n;j++){
        System.out.println("\n Book "+(j+1)+" Details");
        System.out.println(b[j].toString());
    }
}
System.out.println("Kushal Naidu N \n 24BECS408");
}
```

## Output

```
PS D:\Books toString> javac Main.java
PS D:\Books toString> java Main
Enter number of Books :

2

    Enter Book 1 Details
Enter book name :
Java
Enter book author :
Ramesh
Enter book price :
255
Enter number of pages :
798

    Enter Book 2 Details
Enter book name :
DS
Enter book author :
Suresh
Enter book price :
195
Enter number of pages :
654

    Book 1 Details
Book name: Java
Author name: Ramesh
Price: 255
Number of pages: 798

    Book 2 Details
Book name: DS
Author name: Suresh
Price: 195
Number of pages: 654

Kushal Naidu N
24BECS408
PS D:\Books toString>
```

#### **Program 4**

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

Algorithm:

21/10/24

→ LP - 4

Develop a Java Pgm to create an abstract class named Shape that contains 2 integers & an empty method named printArea(). Provide 3 classes named Rectangle, Triangle & 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.

→ Main.java

```
import java.util.Scanner;  
abstract class Shape  
{  
    Scanner s = new Scanner(System.in);  
    double dim1;  
    double dim2;  
    abstract void printArea();  
}  
class Rectangle extends Shape  
{  
    Rectangle()  
    {  
        System.out.println("Enter the dimensions of rectangle  
        (Length, Breadth) :");  
        dim1 = s.nextDouble();  
        dim2 = s.nextDouble();  
    }  
    void printArea()  
    {  
        System.out.println("Area of Rectangle : " + dim1 * dim2)  
    }  
}
```

```
class Triangle extends Shape
```

```
{
```

```
Triangle () {
```

```
System.out.println("Enter the dimensions of Triangle");
dim1 = s.nextDouble();
dim2 = s.nextDouble();
```

```
}
```

```
void printArea () {
```

```
System.out.println("Area of Triangle : " + (dim1 * dim2) / 2);
```

```
}
```

```
}
```

```
class Circle extends Shape
```

```
{
```

```
Circle () {
```

```
System.out.println("Enter the radius of Circle : ");
dim1 = s.nextDouble();
```

```
}
```

```
void printArea () {
```

```
System.out.println("Area of Circle : " + 3.14 * dim1 * dim1);
```

```
}
```

```
}
```

```
class Main {
```

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

```
Rectangle r = new Rectangle ();
```

```
Triangle t = new Triangle ();
```

```
Circle c = new Circle ();
```

```
r.printArea ();
```

```
t.printArea ();
```

```
c.printArea ();
```

```
System.out.println ("Kushal Naidu N \n 24BEC5408");
```

```
% % %
```

```
3 3
```

```
3
```

Output:-

Enter the dimensions of Rectangle (length, breadth) :

6

8

Enter the dimensions of Triangle (base, height) :

9

4

Enter the radius of Circle :

10.5

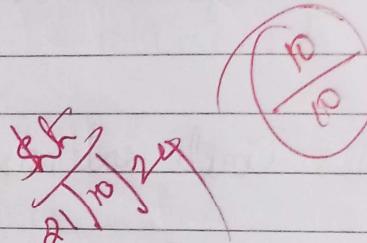
Area of Rectangle : 48.0

Area of Triangle : 4.0

Area of Circle : 346.185

Kushal Naidu. N.

24BEC5408



Code:

```
import java.util.Scanner;
abstract class Shape
{
    Scanner s=new Scanner(System.in);
    double dim1;
    double dim2;
    abstract void printArea();
}
class Rectangle extends Shape
{
    Rectangle()
    {
        System.out.println("Enter the dimensions of rectangle (length, breadth) :");
        dim1=s.nextDouble();
        dim2=s.nextDouble();
    }
    void printArea()
    {
        System.out.println("Area of Rectangle : "+ dim1*dim2);
    }
}
class Triangle extends Shape
{
    Triangle()
    {
        System.out.println("Enter the dimensions of Triangle (base, height) :");
        dim1=s.nextDouble();
        dim2=s.nextDouble();
    }
    void printArea()
    {
        System.out.println("Area of Triangle : "+(dim1*dim2)/2 );
    }
}
class Circle extends Shape
```

```

{
    Circle(){
        System.out.println("Enter the dimensions of Circle (radius) :");
        dim1=s.nextDouble();
    }
    void printArea()
    {
        System.out.println("Area of Circle : "+ 3.14*dim1*dim1);
    }
}
class Main{
    public static void main(String args[]){
        Rectangle r=new Rectangle();
        Triangle t=new Triangle();
        Circle c=new Circle();
        r.printArea();
        t.printArea();
        c.printArea();
        System.out.println("Kushal Naidu N \n 24BECS408");
    }
}
Output

```

```

PS D:\shape> java Main
Enter the dimensions of rectangle (length, breadth) :
6
8
Enter the dimensions of Triangle (base, height) :
2
4
Enter the dimensions of Circle (radius) :
10.5
Area of Rectangle : 48.0
Area of Triangle : 4.0
Area of Circle : 346.185
Kushal Naidu N
24BECS408
PS D:\shape> |

```

## Program 5

Create a Java program to manage bank accounts with a base class Account that stores customer name, account number, account type, and balance. Derive two classes: SavAcct for savings accounts, which supports compound interest and withdrawals but lacks cheque book facilities, and CurAcct for current accounts, which provides cheque book facilities but requires maintaining a minimum balance and imposes penalties if the balance falls below the required amount. Implement methods to deposit money, display balance, compute and add interest (for savings accounts), and handle withdrawals while enforcing minimum balance rules (for current accounts).

Algorithm:

10/24  
→ Program 5 : Bank.java

```
import java.util.Scanner;
class Account {
    String customerName;
    int accNum;
    String accType;
    double balance;
    Account(String customerName, int accNum, String accType,
            double balance) {
        this.customerName = customerName;
        this.accNum = accNum;
        this.accType = accType;
        this.balance = balance;
    }
    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Successfully deposited :" + amount);
        } else {
            S.O.P("Invalid Amount !");
        }
    }
    void displayDetails() {
        S.O.P("Name :" + customerName);
        S.O.P("Account Number :" + accNum);
        S.O.P("Account Type :" + accType);
        S.O.P("Account Balance :" + balance);
    }
}
class SavAcct extends Account {
    private static final double interest = 0.04;
```

LIAISON  
Date \_\_\_\_\_  
Page \_\_\_\_\_

```
Saver (String customerName, int accNum, double balance){  
    super(customerName, accNum, "Savings", balance);  
    System.out.println("Cheque Book facility not available");  
}
```

```
void computeInterest(){  
    double interestAmount = balance * interest;  
    deposit(interestAmount);  
    S.O.P("Interest deposited is " + interestAmount);  
}  
void withdraw(double amount){  
    if (amount > 0 && amount < balance){  
        balance -= amount;  
        S.O.P("Successfully withdrew : " + amount);  
        S.O.P("Updated Balance : " + balance);  
    } else {  
        S.O.P("Insufficient funds!");  
    }  
}
```

```
class Current extends Account{  
    private static final double minimumBalance = 500.0;  
    private static final double penalty = 50.0;  
    Current (String customerName, int accNum, double balance){  
        super(customerName, accNum, "Current", balance);  
        S.O.P("Cheque Book Facility available");  
    }
```

```
void checkMinimumBalance(){  
    if (balance < minimumBalance){  
        balance -= penalty;  
        S.O.P("Less than min. balance " + penalty);  
        S.O.P("Penalty levied : " + penalty);  
    }  
}
```

22/8

```

    void withdraw (double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            S.O.P ("Successfully withdrew : " + amount);
            checkMinimumBalance ();
        } else {
            S.O.P ("Insufficient funds ! ");
        }
    }
}

```

```

public class Bank {
    public static void main (String args) {
        Scanner s = new Scanner (System.in);
        S.O.P ("Enter Name");
        String name = s.next();
        S.O.P ("Enter Acc Num");
        int accNum = s.nextInt();
        S.O.P ("Enter Balance");
        double balance = s.nextDouble();
        S.O.P ("Select Type \n 1. Savings \n 2. Current");
        int opt = s.nextInt();
        boolean exit = false;
        if (opt == 1) {
            Savings sav = new Savings (name, accNum, balance);
            while (!exit) {
                S.O.P ("--- MENU ---");
                S.O.P ("1. Deposit \n 2. Withdraw \n 3. Interest");
                S.O.P ("4. Display \n 5. Exit");
                S.O.P ("Enter choice");
                int choice = s.nextInt();
                switch (choice) {
                    case 1:
                        S.O.P ("Enter amount");
                        double amount = s.nextDouble();

```

```

def sav_deposit(amount);
break;
case 2:
    S.O.P("Enter amount");
    double w_amount = s.nextDouble();
    sav_withdraw(w_amount);
    break;
case 3:
    sav_computInterest();
    break;
case 4:
    sav_displayDetails();
    break;
case 5:
    exit = true;
}
}

else if (opt == 2) {
    choice cur = new Curacc(name, accNum, balance);
    while (!exit) {
        // Same Menu
        switch (choice) {
            case 1:
                S.O.P("Enter amount");
                double amount = s.nextDouble();
                cur.deposit(amount);
                break;
            case 2:
                S.O.P("Enter amount");
                double w_amount = s.nextDouble();
                cur.withdraw(w_amount);
                break;
        }
    }
}

```

case 3:

SOP("Can't compute interest for current");  
break;

case 4:

cur.displayDetails();  
break;

case 5:

Isit = True;

333

--- MENU ---

- 1. Deposit
- 2. Withdraw
- 3. Compute Interest
- 4. Display Details
- 5. Exit
- 4

(10)  
10

Name: Rohan

Account Number: 102

Account Type: Current

Account Balance: 375.5

→ 3

Can't compute Interest for Current Account

→ 4 2

Enter the Withdrawal amount

900

Successfully withdrew: 900.0

Updated balance: 2325.65

Code:

```
import java.util.Scanner;
class Account
{
    String customerName;
    int accNum;
    String accType;
    double balance;
    Account(String customerName, int accNum, String accType, double balance){
        this.customerName=customerName;
        this.accNum=accNum;
        this.accType=accType;
        this.balance=balance;
    }
    void deposit(double amount){
        if(amount>0){
            balance += amount;
            System.out.println("Successfully deposited :" +amount);
        }
        else{
            System.out.println("Invalid Amount !");
        }
    }
    void displayDetails(){
        System.out.println("Name : "+customerName);
        System.out.println("Account Number : "+accNum);
        System.out.println("Account Type : "+accType);
        System.out.println("Account Balance : "+balance);
    }
}
class Savacc extends Account
{
    private static final double interest=0.04;
    Savacc(String customerName, int accNum, double balance){
```

```

super(customerName, accNum, "Savings", balance);
System.out.println("Cheque Book Facility not available");

}

void computeInterest(){
    double interest_amount=balance*interest;
    deposit(interest_amount);
    System.out.println("Interest Deposited is "+interest_amount);
}

void withdraw(double amount){
    if(amount>0 && amount<=balance){
        balance -= amount;
        System.out.println("Successfully withdrew : "+amount);
        System.out.println("Updated balance : "+balance);
    }else{
        System.out.println("Insufficient Funds or Invalid entry !");
    }
}

class Curacc extends Account
{
    private static final double minimum_balance=500.0;
    private static final double penalty=50.0;
    Curacc(String customerName, int accNum, double balance){
        super(customerName, accNum, "Current", balance);
        System.out.println("Cheque Book Facility available");
    }

    void checkMinimumBalance(){
        if(balance<minimum_balance){
            balance -= penalty;
            System.out.println("Your account balance is less than minimum
balance : "+minimum_balance);
            System.out.println("Penalty levied : "+penalty);
        }
    }
}

```

```

void withdraw(double amount){
    if(amount>0 && amount<=balance){
        balance -= amount;
        System.out.println("Successfully withdrew : "+amount);
        checkMinimumBalance();
    }else{
        System.out.println("Insufficient Funds or Invalid entry !");
    }
}

public class Bank{
    public static void main(String []args){
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the customer name :");
        String name=s.next();
        System.out.println("Enter the Account Number :");
        int accNum=s.nextInt();
        System.out.println("Enter the Account Balance :");
        double balance=s.nextDouble();
        System.out.println("Select the Account type \n1.Savings \n2.Current");
        int opt=s.nextInt();
        boolean exit=false;
        if(opt==1){
            Savacc sav=new Savacc(name, accNum, balance);
            while(!exit){
                System.out.println("-----MENU-----");
                System.out.println("1.Deposit \n2.Withdraw \n3.Compute
Interest \n4.Display Details \n5.Exit");
                int choice=s.nextInt();
                switch(choice){
                    case 1:
                        System.out.println("Enter the deposit amount");
                        double amount=s.nextDouble();
                        sav.deposit(amount);
                        break;
                }
            }
        }
    }
}

```

```

        case 2:
            System.out.println("Enter the withdrawal
amount");
            double w_amount=s.nextDouble();
            sav.withdraw(w_amount);
            break;
        case 3:
            sav.computeInterest();
            break;
        case 4:
            sav.displayDetails();
            break;
        case 5:
            exit=true;
    }
}
}

else if (opt==2){
    Curacc cur=new Curacc(name, accNum, balance);
    while(!exit){
        System.out.println("-----MENU-----");
        System.out.println("1.Deposit \n2.Withdraw \n3.Compute
Interest \n4.Display Details \n5.Exit");
        int choice=s.nextInt();
        switch(choice){
            case 1:
                System.out.println("Enter the deposit amount");
                double amount=s.nextDouble();
                cur.deposit(amount);
                break;
            case 2:
                System.out.println("Enter the withdrawal
amount");
                double w_amount=s.nextDouble();
                cur.withdraw(w_amount);

```

```
        break;
    case 3:
        System.out.println("Cannot Compute Interest
for Current Account");
        break;
    case 4:
        cur.displayDetails();
        break;
    case 5:
        exit=true;
    }
}
}
System.out.println("Kushal Naidu N \n 24BECS408");
}
}
```

## Output

```
PS D:\24BECS408\bank> javac Bank.java
PS D:\24BECS408\bank> java Bank
Enter the customer name :
Kushal
Enter the Account Number :
124
Enter the Account Balance :
2000
Select the Account type
1.Savings
2.Current
1
Cheque Book Facility not available
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
1
Enter the deposit amount
525.65
Successfully deposited :525.65
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
4
Name : Kushal
Account Number : 124
Account Type : Savings
Account Balance : 2525.65
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
2
Enter the withdrawal amount
200
Successfully withdrew : 200.0
Updated balance : 2325.65
```

```
Updated Balance : 29287.00
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
3
Successfully deposited :93.02600000000001
Interest Deposited is 93.02600000000001
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
5
PS D:\24BECs408\bank> java Bank
Enter the customer name :
Rohan
Enter the Account Number :
102
Enter the Account Balance :
6000
Select the Account type
1.Savings
2.Current
2
Cheque Book Facility available
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
1
Enter the deposit amount
225.5
Successfully deposited :225.5
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
2
Enter the withdrawal amount
5800
Successfully withdrew : 5800.0
Your account balance is less than minimum balance : 500.0
Penalty levied : 50.0
```

```
PS D:\24BECS408\bank> javac Bank.java
PS D:\24BECS408\bank> java Bank
Enter the customer name :
Kushal
Enter the Account Number :
124
Enter the Account Balance :
2000
Select the Account type
1.Savings
2.Current
1
Cheque Book Facility not available
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
1
Enter the deposit amount
525.65
Successfully deposited :525.65
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
4
Name : Kushal
Account Number : 124
Account Type : Savings
Account Balance : 2525.65
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
2
Enter the withdrawal amount
200
Successfully withdrew : 200.0
Updated balance : 2325.65
```

## **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:

11/24

→ Program 6: Create a package CIE which has 2 classes - Student & Internals. The class Student has members like usn, name, sem. The class Internals derived from Student has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student.

→ CIE/Student.java

```
package CIE;
import java.util.Scanner;
public class Student {
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails () {
        Scanner s = new Scanner(System.in);
        usn = s.nextLine();
        System.out.println("Enter USN : ");
        usn = s.nextLine();
        System.out.println("Enter name : ");
        name = s.nextLine();
        System.out.println("Enter Semester : ");
        sem = s.nextInt();
    }
    public void displayStudentDetails () {
        System.out.println("USN :" + usn);
        System.out.println("Name :" + name);
        System.out.println("Sem :" + sem);
    }
}
```

→ CIE/internals.java

```
package CIE;
import java.util.Scanner;
public class internals extends student {
    public int marks[] = new int [5];
    public void inputCIEmarks () {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the internal marks out of 50 :");
        for (int i=0; i<5; i++) {
            System.out.println ("Course " +(i+1) + ":");
            marks[i] = s.nextInt();
        }
    }
}
```

→ SEE/externals.java

```
package SEE;
import CIE.internals;
import java.util.Scanner;
public class externals extends internals {
    protected int externalmarks[];
    protected int finalMarks[];
    public externals () {
        externalmarks = new int [5];
        finalMarks = new int [5];
    }
    public void inputSEEmarks () {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter External marks for 100 :");
        for (int i=0; i<5; i++) {
            System.out.println ("Enter marks for course " +(i+1) + ":");
            externalmarks[i] = s.nextInt();
        }
    }
}
```

```

public void calculateFinalMarks() {
    for (int i=0; i<5; i++) {
        finalMarks[i] = marks[i] + externalmarks[i]/2;
    }
}

public void displayFinalMarks() {
    displayStudentDetails();
    System.out.println("Final Marks : ");
    for (int i=0; i<5; i++) {
        System.out.println("Course " + (i+1) + ":" + finalMarks[i]);
    }
}

```

→ Main.java

```

import SEE.externalmarks;
import java.util.Scanner;
public class Main {
    public static void main(String [] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter number of students : ");
        int n = s.nextInt();
        externalmarks [] students = new externalmarks[n];
        for (int i=0; i<n; i++) {
            students[i] = new externalmarks();
            students[i].inputStudentDetails();
            students[i].inputFEmarks();
            students[i].inputSEEmarks();
            students[i].calculateFinalMarks();
        }
        for (int i=0; i<n; i++) {
            students[i].displayFinalMarks();
        }
        System.out.println("Kushal Naik N. 1n24BECS408");
    }
}

```

O/P:- Enter no. of students: 1

Enter Student USN : CS408

Enter Student Name: Kushal

Enter Student Semester: 3

Enter the Internal Marks for the following subjects out of 50  
Course 1:

24

37 Course 2: 37

Course 3: 39

Course 4: 41

Course 5: 37

Enter external marks out of 100

Course 1: 87

Course 2: 74

Course 3: 68

Course 4: 69

Course 5: 8

USN of the Student : CS408

Name of the Student: Kushal

Semester of the Student: 3

Final Marks

Course 1: 67

Course 2: 74

Course 3: 73

Course 4: 79

Course 5: 77

Kushal Nairdu N

24 BECS408

10  
10  
2/12/19

Code:

CIE/internals.java

```
package CIE;
import java.util.Scanner;
public class internals extends student{
    public int marks[] = new int[5];
    public void inputCIEmarks()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the Internal Marks for the following Subjects
out of 50 :");
        for (int i=0;i<5;i++)
        {
            System.out.println("Course "+(i+1)+":");
            marks[i]=s.nextInt();
        }
    }
}
```

CIE/student.java

```
package CIE;
import java.util.Scanner;
public class student{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter Student USN :");
        usn=s.next();
        System.out.println("Enter Student Name :");
```

```

        name=s.next();
        System.out.println("Enter Student Semester :");
        sem=s.nextInt();
    }
    public void displayStudentDetails()
    {
        System.out.println("USN of the Student : "+ usn);
        System.out.println("Name of the Student : "+ name);
        System.out.println("Semester of the Student : "+ sem);
    }
}

```

### SEE/externals.java

```

package SEE;
import CIE.internals;
import java.util.Scanner;
public class externals extends internals {
    protected int externalmarks[];
    protected int finalMarks[];
    public externals(){
        externalmarks=new int[5];
        finalMarks = new int[5];
    }
    public void inputSEEmarks() {
Scanner s = new Scanner(System.in);
System.out.println("Enter External marks for 5 courses for 100: ");
for (int i = 0; i < 5; i++) {
    System.out.print("Enter marks for course " +(i + 1) + ": ");
    externalmarks[i] = s.nextInt();
}
}
    public void calculateFinalMarks(){
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + externalmarks[i]/2;
        }
    }
}

```

```

        }
    }

    public void displayFinalMarks(){
        displayStudentDetails();
        System.out.println("Final Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course "+(i+1)+":"+finalMarks[i]);
        }
    }
}

```

Main.java

```

import SEE.externals;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter number of students: ");
        int n = s.nextInt();
        externals[] students = new externals[n];
        for (int i = 0; i < n; i++) {
            students[i] = new externals();
            students[i].inputStudentDetails();
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
            students[i].calculateFinalMarks();
        }
        for (int i = 0; i < n; i++) {
            students[i].displayFinalMarks();
        }
        System.out.println("Kushal Naidu N \n24BECS408");
    }
}

```

## Output

```
Enter number of students:  
2  
Enter Student USN :  
CS408  
Enter Student Name :  
Kushal  
Enter Student Semester :  
3  
Enter the Internal Marks for the following Subjects out of 50 :  
Course 1:  
24  
Course 2:  
37  
Course 3:  
39  
Course 4:  
41  
Course 5:  
37  
Enter External marks for 5 courses for 100:  
Enter marks for course 1: 87  
Enter marks for course 2: 74  
Enter marks for course 3: 68  
Enter marks for course 4: 62  
Enter marks for course 5: 80  
Enter Student USN :  
CS420  
Enter Student Name :  
Sachit  
Enter Student Semester :  
3  
Enter the Internal Marks for the following Subjects out of 50 :  
Course 1:  
31  
Course 2:  
37  
Course 3:  
30  
Course 4:  
44  
Course 5:  
42  
Enter External marks for 5 courses for 100:  
Enter marks for course 1: 87  
Enter marks for course 2: 81  
Enter marks for course 3: 76  
Enter marks for course 4: 68
```

```
USN of the Student : CS408
Name of the Student : Kushal
Semester of the Student : 3
Final Marks:
Course 1:67
Course 2:74
Course 3:73
Course 4:72
Course 5:77
USN of the Student : CS420
Name of the Student : Sachit
Semester of the Student : 3
Final Marks:
Course 1:74
Course 2:77
Course 3:68
Course 4:78
Course 5:89
Kushal Naidu N
24BECS408
PS D:\package> |
```

## Program 7

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

Algorithm:

→ Program 7: WAP that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" & derived class called "Son" which extends the base class. In father class, implement a constructor which takes the age & throws a the exception WrongAge() when the input age < 0. In Son class, implement a constructor that uses both father & son's age & throws an exception if son's age is  $\geq$  father's age.

→ ExceptionHandling.java

```
import java.util.Scanner;  
  
class WrongAge extends Exception {  
    public WrongAge() {  
        super("Age error occurred");  
    }  
    public WrongAge(String Message) {  
        super(Message);  
    }  
}  
  
class Father {  
    protected int fatherAge;  
    public Father() throws WrongAge {  
        Scanner s = new Scanner(System.in);  
        System.out.print("Enter father's age");  
        fatherAge = s.nextInt();  
        if (fatherAge  $\leq$  0) {  
            throw new WrongAge("Age can't be negative");  
        }  
    }  
    public void display() {  
        System.out.println("Father's age :" + fatherAge);  
    }  
}
```

```

class Son extends Father {
    private int sonAge;
    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter son's age: ");
        sonAge = s.nextInt();
        if (sonAge < 0)
            throw new WrongAge("Age can't be negative");
        else if (sonAge >= fatherAge)
            throw new WrongAge("Son's age can't be greater than
                or equal to father's age");
    }
    public void display() {
        super.display();
        System.out.println("Son's age: " + sonAge);
    }
}

public class ExceptionHandling {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}

```

O/p → Enter father's age: 56  
 Enter son's age: -20  
 Exception: Age can't be negative

Enter father's age: 29  
 Enter son's age: 39  
 Exception: Son's age can't be greater than or equal to father's age.

Enter father's age: 45  
 Enter son's age: 21  
 Father's age: 45  
 Son's age: 21

Code:

```
import java.util.Scanner;
class WrongAge extends Exception {
    public WrongAge() {
        super("Age error occurred");
    }
    public WrongAge(String message) {
        super(message);
    }
}
class Father {
    protected int fatherAge;
    public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter father's age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
    public void display() {
        System.out.println("Father's age: " + fatherAge);
    }
}
class Son extends Father {
    private int sonAge;
    public Son() throws WrongAge {
        super();
        Scanner s = new Scanner(System.in);
        System.out.print("Enter son's age: ");
        sonAge = s.nextInt();
        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        } else if (sonAge >= fatherAge) {
```

```

        throw new WrongAge("Son's age cannot be greater than or equal to father's
age");
    }
}
public void display() {
    super.display();
    System.out.println("Son's age: " + sonAge);
}
}

public class ExceptionHandling {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
        System.out.println("Kushal Naidu N \n24BECS408");
    }
}

```

## Output

```
C:\Users\DELL\Desktop\exception\exception>java ExceptionHandling
Enter father's age: 56
Enter son's age: 25
Father's age: 56
Son's age: 25
Kushal Naidu N
24BECS408

C:\Users\DELL\Desktop\exception\exception>java ExceptionHandling
Enter father's age: 42
Enter son's age: 48
Exception: Son's age cannot be greater than or equal to father's age
Kushal Naidu N
24BECS408

C:\Users\DELL\Desktop\exception\exception>java ExceptionHandling
Enter father's age: -65
Exception: Age cannot be negative
Kushal Naidu N
24BECS408

C:\Users\DELL\Desktop\exception\exception>java ExceptionHandling
Enter father's age: 54
Enter son's age: -30
Exception: Age cannot be negative
Kushal Naidu N
24BECS408
```

## Program 8

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

Algorithm:

8/11/24  
→ Program-8: WAP which creates 2 threads, 1 thread displaying "BMS College of Engineering" once every 10 seconds & another displaying "CSE" once every 2 seconds

→ Main.java

```
class BMS extends Thread {
    public void run() {
        try {
            while(true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("BMS College Thread Interrupted");
        }
    }
}

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

public class Main {
    public static void main(String [] args) {
        BMS bmsThread = new BMS();
        CSE cseThread = new CSE();
        bmsThread.start();
        cseThread.start();
    }
}
```

Q.P:- BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

10  
10

8x  
21/12/20

Code:

```
class BMS extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("BMSCollegeThread interrupted");  
        }  
    }  
}  
class CSE extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("CSE");  
                Thread.sleep(2000);  
            }  
        } catch (InterruptedException e) {  
            System.out.println("CSEThread interrupted");  
        }  
    }  
}  
public class Main {  
    public static void main(String[] args) {  
        BMS bmsThread = new BMS();  
        CSE cseThread = new CSE();  
        bmsThread.start();  
        cseThread.start();  
        System.out.println("Kushal Naidu N \n24BECS408");  
    }  
}
```

## Output

```
C:\Users\DELL\Desktop\exception>java Main
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
```

## **Program 9**

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

Algorithm:

→ Program 9:  
WAP that creates a user interface to perform integer divisions.  
The user enters 2 nos in the text fields, Num1 & Num2. The  
division of Num1 & Num2 is displayed in the Result field when  
the Divide button is clicked. If Num1 or Num2 weren't an  
integer, the pgm would throw a NumberFormatException. If  
Num2 were zero, the pgm would throw an Arithmetic  
Exception. Display the exception in a message dialog box.

```
import java.awt.*;  
import java.awt.event.*;  
  
class SwingDemo {  
    SwingDemo () {  
        JFrame jfrm = new JFrame("Divider App");  
        jfrm.setSize(975, 150);  
        jfrm.setLayout(new FlowLayout());  
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        JLabel jlbl = new JLabel("Enter the divisor & dividend:");  
        JTextField aitf = new JTextField(8);  
        JTextField btf = new JTextField(8);  
        JButton button = new JButton("Calculate");  
        JLabel err = new JLabel();  
        JLabel ablk = new JLabel();  
        JLabel bblk = new JLabel();  
        JLabel ansblk = new JLabel();  
        jfrm.add(blk);  
        jfrm.add(jlbl);  
        jfrm.add(aitf);  
        jfrm.add(btf);  
        jfrm.add(button);  
    }  
}
```

```
ifrm.add(alab);
ifrm.add(blab);
ifrm.add(anslab);
```

```
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            int a = Integer.parseInt(alif.getText());
            int b = Integer.parseInt(blif.getText());
            int ans = a / b;
            err.setText("");
            alab.setText("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + ans);
        } catch (NumberFormatException e) {
            alab.setText(" ");
            blab.setText(" ");
            anslab.setText(" ");
            err.setText("Enter only Integers !");
        } catch (ArithmaticException e) {
            alab.setText(" ");
            blab.setText(" ");
            anslab.setText(" ");
            err.setText("B should be NON zero !");
        }
    }
});
```

```
ifrm.setVisible(true);
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDems();
        }
    });
}
```

Divider App

Enter the divisor and dividend:

20

9

Calculate

$$A=20 \quad B=9 \quad Ans=10$$

10  
10

~~10~~  
~~2 | 12 | 04~~

Code:

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

class SwingDemo {
    SwingDemo() {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divisor and dividend:");
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        JButton button = new JButton("Calculate");
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);

        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                try {
                    int a = Integer.parseInt(ajtf.getText());

```

```

        int b = Integer.parseInt(bjtf.getText());
        int ans = a / b;
        err.setText("");
        alab.setText("A = " + a);
        blab.setText("B = " + b);
        anslab.setText("Ans = " + ans);
    } catch (NumberFormatException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("Enter Only Integers!");
    } catch (ArithmaticException e) {
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be NON zero!");
    }
}
});

jfrm.setVisible(true);
}

public static void main(String args[]) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
    System.out.println("Kushal Naidu N \n24BECS408");
}
}

```

## Output

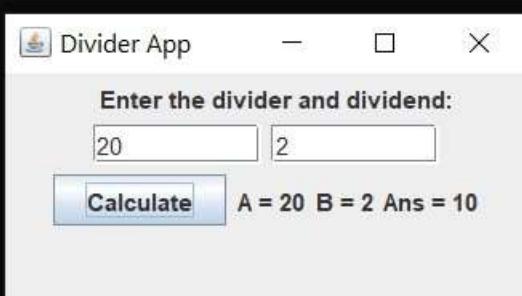
```
Administrator:~\new_actionlistener>
Administrator:~\new_actionlistener> Command Prompt - java SwingDemo
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>cd Desktop

C:\Users\DELL\Desktop>cd exception

C:\Users\DELL\Desktop\exception>javac SwingDemo.java

C:\Users\DELL\Desktop\exception>java SwingDemo
```



## Program 10(a)

Demonstrate Inter process Communication

Algorithm:

01/11/24 → Program 10: WAP to demonstrate inter process communication & deadlock.

(i) Inter Process Communication (IPC)

```
class A {
    int n;
    boolean valueset = false;
    Synchronized int get() {
        while (!valueset) {
            try {
                System.out.println("In consumer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
            }
            System.out.println("GOT = " + n);
            valueset = true;
            System.out.println("In Intimate producer\n");
            notify();
        }
        return n;
    }

    Synchronized void put(int n) {
        while (valueset)
            try {
                System.out.println("In Producer Waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
            }
        this.n = n;
        valueset = true;
    }
}
```

```

System.out.println("Put: " + n);
System.out.println("In Intimate Consumer\n");
notify();
}
}
}

```

```
class producer implements Runnable {
```

```

    Queue q;
    producer(Q q);
    this.q = q;
    new Thread(this, "Producer").start();
}

public void run() {
    int i = 0;
    while (i < 5) {
        q.put(i + 1);
    }
}
```

```
class consumer implements Runnable {
```

```

    Queue q;
    consumer(Q q);
    this.q = q;
    new Thread(this, "consumer").start();
}

public void run() {
    int i = 0;
    while (i < 5) {

```

```

        int r = q.get();
        System.out.println("Consumed: " + r);
    }
}
```

class Pc Fixed {

public static void main (String [] args) {

Qq = new BQ();

new producer (Qq);

new consumer (Qq);

System.out.println ("press ~~as~~ ctrl+c to stop");

}

Output:-

~~press ctrl+c to stop~~

put : \* 0

~~Intimate consumer  
producer waiting~~

got : 0

~~Intimate producer  
producer waiting~~

put : 1

~~Intimate consumer  
producer waiting  
consumed : 0~~

got : ~~1~~

~~Intimate producer~~

~~consumed : 1~~

put : 2

~~Intimate consumer  
producer waiting~~

got : 2

~~Intimate producer~~

~~consumed : 2~~

put : 3

~~Intimate Consumer  
producer waiting~~

got : 3

~~Intimate producer  
consumed : 3~~

put : 4

~~Intimate consumer  
got : 4~~

~~Intimate producer  
consumed : 4.~~

Code:

```
class Q {  
    int n;  
    boolean valueSet = false;  
  
    synchronized int get() {  
        while (!valueSet) {  
            try {  
                System.out.println("\nConsumer waiting\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        }  
        System.out.println("Got: " + n);  
        valueSet = false;  
        System.out.println("\nNotify Producer\n");  
        notify();  
        return n;  
    }  
  
    synchronized void put(int n) {  
        while (valueSet) {  
            try {  
                System.out.println("\nProducer waiting\n");  
                wait();  
            } catch (InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        }  
        this.n = n;  
        valueSet = true;  
        System.out.println("Put: " + n);  
        System.out.println("\nNotify Consumer\n");  
    }  
}
```

```

        notify();
    }
}

class Producer implements Runnable {
    Q q;

    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }

    public void run() {
        int i = 0;
        while (i < 15) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;

    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }

    public void run() {
        int i = 0;
        while (i < 15) {
            int r = q.get();
            System.out.println("Consumed: " + r);
            i++;
        }
    }
}

```

```
    }
}

class PCFixed {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
        System.out.println("Kushal Naidu \n24BECS408");
    }
}
```

## Output

```
PS C:\Users\STUDENT\Desktop\10(i) IPC> java PCFixed
Press Control-C to stop.
Kushal Naidu
24BECS408
Put: 0

Notify Consumer

Producer waiting

Got: 0

Notify Producer

Put: 1

Notify Consumer

Producer waiting

Consumed: 0
Got: 1

Notify Producer

Consumed: 1
Put: 2

Notify Consumer

Producer waiting

Got: 2

Notify Producer

Consumed: 2
Put: 3

Notify Consumer

Producer waiting

Got: 3

Notify Producer
```

Consumed: 7

Put: 8

Notify Consumer

Producer waiting

Got: 8

Notify Producer

Consumed: 8

Put: 9

Notify Consumer

Producer waiting

Got: 9

Notify Producer

Consumed: 9

Put: 10

Notify Consumer

Producer waiting

Got: 10

Notify Producer

Consumed: 10

Put: 11

Notify Consumer

Producer waiting

Got: 11

Notify Producer

Consumed: 11

Put: 12

### Program 10(b)

Demonstrate deadlock

Algorithm:

| Deadlock

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

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

```
void last() {
```

```
    System.out.println("Inside A.last");
```

```
}
```

```
class Deadlock implements Runnable {
```

```
    A a = new A();
```

```
    B b = new B();
```

```
    Deadlock() {
```

```
        Thread currentThread = Thread.currentThread().setName("Main Thread");
```

```
        Thread t = new Thread(this, "Racing Thread");
```

```
        t.start();
```

```
        a.feed(b);
```

```
        System.out.println("Back in main thread");
```

```
}
```

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

```
    b.bark(a);
```

```
    System.out.println("Back in other thread");
```

```
}
```

```
}
```

O/P:-

Main Thread entered A for

Racing Thread entered B. for

Racing Thread trying to call A.last()

Inside A.last

Back in other thread

), Main Thread trying to call B.last()

Inside A.last

Back in main thread.

(10/10)

2/12/2\*

Code:

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

```

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this,"RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }
    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String[] args) {
        new Deadlock();
        System.out.println("Kushal Naidu N \n24BECS408");
    }
}

```

## Output

```

PS C:\Users\STUDENT\Desktop\10(ii) Deadlock> javac Deadlock.java
PS C:\Users\STUDENT\Desktop\10(ii) Deadlock> java Deadlock
RacingThread entered B.bar
MainThread enteredA.foo
MainThread trying to call B.last()
Inside A.last
Back in main thread
Kushal Naidu N
24BECS408
RacingThread trying to call A.last()
Inside A.last
Back in other thread
PS C:\Users\STUDENT\Desktop\10(ii) Deadlock> |

```