

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



LAB REPORT
on
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

Ayush Aditya(**1BM23CS057**)

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 **Ayush Aditya(1BM23CS057)**, 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.

Lab faculty Incharge Name:Dr. Prasad G R Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE
--	---

Index

Sl. No.	Date	Experiment Title	Page No.
1	25-9-24	Implement Quadratic Equation	1-5
2	2-10-24	Develop a Java program to create a class student with members usn,name and array marks. Include methods to accept and display details and a method to calculate SGPA of student.	6-12
3	9-10-24	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 <code>toString()</code> method that could display the complete details of the book. Develop a Java program to create n book objects.	13-17
4	23-10-24	.Develop a Java program to create an abstract class name shape that contains two integers and an empty method named <code>printArea()</code> .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 <code>printArea()</code> that prints the area of the given Shape.	18-22
5	30-10-24	Create a class Account that stores customer name, account number and type of account. From this	23-36

		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:	
6	13-11-24	Create a package CIE which has two classes- Student and 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 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.	37-44
7	20-11-24	Write a program that demonstrate handling of exception in inheritance String.Create a base class called Father and a derived class called son which extends the base class in Father's class implement a constructor that takes the age and throws exception Wrong age when input age is less than 0.In son's class implements constructor uses both father's and son's age and throws an exception when son's age >= father's age.	45-51
8	27-11-24	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.	52-53
9	27-11-24	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.	54-55
10	27-11-24	Write a Java program to create two threads, one displays "Computer Science" and another display "Information Science" five times.	55-56

Github Link:

https://github.com/Ayush-bmsce/Java_Lab

Program 1

Implement Quadratic Equation

Code:

```
import java.util.Scanner;
import java.util.lang.Math;
public class quadEq{
    public static void main(String[] args){
        System.out.println("Enter coefficients a,b and c");
        Scanner scanner = new Scanner(System.in);
        int a = scanner.nextInt();
        int b = scanner.nextInt();
        int c = scanner.nextInt();
        if(a == 0)
        {
            System.out.println("Not a quadratic Equation");
        }
        else{
            int d = b*b-4*a*c;
            if(d > 0){
                float r1 = (-b+Math.sqrt(d))/(2*a);
                float r2 = (-b-Math.sqrt(d))/(2*a);
                System.out.println("r1 = " + r1);
                System.out.println("r2 = " + r2);
            }
            else if(d == 0){
                float r = -b/(2*a);
                System.out.println("r = " + r);
            }
            else{
                System.out.println("Roots are imaginary");
            }
        }
    }
}
```

}

}

Output-

The screenshot shows a Microsoft Windows Command Prompt window titled "Administrator: Command Prompt". The window displays the following command-line session:

```
C:\Windows\system32>cd ..
C:\Windows>cd ..
C:\>cd ayush
C:\ayush>javac quadEq.java
error: file not found: quadEq.java
usage: javac <options> <source files>
use -help for a list of possible options
C:\ayush>quadEq.java
'quadEq.java' is not recognized as an internal or external command,
operable program or batch file.
C:\ayush>javac quad.java
quad.java:5: error: ')' or ',' expected
        System.out.println("Enter coefficients a,b and c");
                           ^
quad.java:35: error: reached end of file while parsing
    }
2 errors
C:\ayush>javac quad.java
quad.java:2: error: package java.util.lang does not exist
import java.util.lang.Math;
          ^
1 error
C:\ayush>javac quad.java
C:\ayush>java quad
Enter coefficients a,b and c
1
1
2
r1 = -1.0
r2 = -2.0
C:\ayush>java quad
Enter coefficients a,b and c
1
1
1
Roots are imaginary
C:\ayush>java quad
Enter coefficients a,b and c
1
1
1
Roots are imaginary
C:\ayush>
```

The taskbar at the bottom of the screen shows several pinned icons, including File Explorer, Edge, and File History. The system tray indicates the date as 09-10-2024 and the time as 15:55.

Observation Notebook-

① Develop a Java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a , b , and c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is negative, display a message stating that there are no real solutions.

```
→ public class QuadEq {
    import java.util.Scanner;
    import java.lang.Math;
    public class QuadEq {
        public static void main (String [] args) {
            System.out.println ("Enter coefficients a, b, then c");
            Scanner scanner = new Scanner (System.in);
            int a = scanner.nextInt ();
            int b = scanner.nextInt ();
            int c = scanner.nextInt ();
            if (a == 0)
                System.out.println ("Not a quadratic Eqn");
            else {
                int d = b * b - 4 * a * c;
```

if ($d > 0$)

double
cout. $\pi_1 = -b / 2 * a$

float cout $\pi_1 = (-b + \sqrt{b^2 - 4ac}) / 2 * a$;

double

float cout $\pi_2 = (-b - \sqrt{b^2 - 4ac}) / 2 * a$;

System.out.println(" $\pi_1 = "$

+ π_1);

System.out.println(" $\pi_2 = "$

+ π_2);

} else if ($d == 0$) {

float r = ~~(-b / 2) - b / (2 * a)~~;

System.out.println(" Roots are equal and real ");

System.out.println(" $\pi_1 = "$

+ r);

} else {

System.out.println(" Roots are imaginary ");

double $\pi_1 = (-b / 2) / (2 * a)$;

double $\pi_2 = -d / 2 * a$;

$\sqrt{a} / (2 * a)$;

System.out.println(" $\pi_1 = " + \pi_1$

+ " + (" + π_2));

System.out.println(" $\pi_2 = " + \pi_2$

+ " - (" + π_2));

}}}

② Develop a Java program
to create a class student
with static members using
name, an array credits [any
size].

Output

① Enter coefficients a, b and c.

$$\begin{array}{r} 2 \\ \times 1 \\ \hline 2 \\ 2 \\ \hline 1 \end{array}$$

Roots are real and equal

$$x = -1$$

② Enter coefficients a, b, and c

$$1 \ 3 \ 2$$

$$\sqrt{x_1} = -1, 0$$

$$\sqrt{x_2} = -2, 0$$

$$\begin{array}{r} 1 \\ \times 2 \\ \hline 2 \\ 2 \\ \hline 0 \end{array}$$

③ Enter coefficients a, b, and c

$$\begin{array}{r} 1 \\ \times 3 \\ \hline 3 \\ 3 \\ \hline 0 \end{array}$$

Roots are imaginary.

$$\sqrt{x_1} = (-1 + i\sqrt{3})/2$$

$$\sqrt{x_2} = (-1 - i\sqrt{3})/2$$

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

Code-

```
import java.util.Scanner;
class Subject {
    int subjectMarks, credits, grade;
    Subject subject[8];
}
class Student {
    String name, usn;
    double SGPA;
    Scanner s = new Scanner(System.in);
    Student() {
        subject = new Subject[9];
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
            s = new Scanner(System.in);
        }
    }
    void getStudentDetails() {
        System.out.println("ENTER YOUR NAME:");
        name = s.nextLine();
        System.out.println("ENTER YOUR USN:");
        usn = s.nextLine();
    }
    void getMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.println("enter marks of subject" + (i + 1) + ":");
            subject[i].subjectMarks = s.nextInt();
            System.out.println("enter credits of subject" + (i + 1) + ":");
            subject[i].credits = s.nextInt();
            if (subject[i].subjectMarks >= 90) {
                subject[i].grade = 10;
            } else if (subject[i].subjectMarks >= 80) {
                subject[i].grade = 9;
            } else if (subject[i].subjectMarks >= 70) {
                subject[i].grade = 8;
            } else if (subject[i].subjectMarks >= 60) {
                subject[i].grade = 7;
            } else if (subject[i].subjectMarks >= 50) {
                subject[i].grade = 6;
            } else if (subject[i].subjectMarks >= 40) {
                subject[i].grade = 5;
            } else {
                subject[i].grade = 0;
            }
        }
    }
    void computeSGPA() {
        int totalCredits = 0, total = 0;
        for (int i = 0; i < 8; i++) {
            totalCredits = totalCredits + subject[i].credits;
        }
        for (int i = 0; i < 8; i++) {
            total = total + (subject[i].credits * subject[i].grade);
        }
        SGPA = total / totalCredits;
```

```
}

public static void main(String args[]) {
    Student s1 = new Student();
    s1.getStudentDetails();
    s1.getMarks();
    s1.computeSGPA();
}
}
```

Output-

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window displays the following interaction:

```
Enter credits of subject 3:  
88  
Enter marks of subject 4:  
99  
Enter credits of subject 4:  
56  
Enter marks of subject 5:  
77  
Enter credits of subject 5:  
55  
Enter marks of subject 6:  
88  
Enter credits of subject 6:  
44  
Enter marks of subject 7:  
69  
Enter credits of subject 7:  
78  
Enter marks of subject 8:  
45  
Enter credits of subject 8:  
68  
Enter marks of subject 9:  
98  
Enter credits of subject 9:  
68  
SGPA for Ayush (Aditya): 6.8979591836734695  
C:\Users\AYUSH ADITYA\Desktop>USN : 1BM23CS057|
```

The taskbar at the bottom of the screen shows various pinned icons and system status indicators, including the date and time (10/17/2024, 10:32 PM).

Observation notebook-

16-10-2021
 Develop a Java program to create a class Student with members usm, name, an array marks to marks. Include methods to accept and display details and a method to calculate CGPA of student.

import java.util. Arrays
 Scanner scanner;

class Student { details }

int[] marks = new int[8]

String name, usm;
 int credits;

Scanner scanner =

new Scanner(System.in)

double CGPA;

void getdetails()

{

System.out.println("Enter
 the usm");

usm = scanner.nextLine();

System.out.println("Enter
 the name");

name = scanner.nextLine();

System.out.println("Enter
 marks of all 8 subjects");

for (int i=0; i<8; ++)

3 marks[i] = scanner.nextInt()

3 void SUPA calc()

{ int sum_credits = 0;

int

for (int i = 0; i < 8; i++)

{

sum_credits += (marks[i] / 10);

}

SUPA =

for (int i = 0; i < 8; i++)

{

System.out.println("Enter

Credit for " + i + "th

subject");

Credits = scanner.nextInt();

sum_credits += (marks * credits / 10);

}

SUPA = sum_credits / 20;

3 3

void display()

```
System.out.println("USM" +  
    "USM");
```

```
System.out.println("Name:" +  
    "Name");
```

```
for (int i = 0; i < 8; i++)
```

```
    System.out.println("Marks  
of " + (i + 1) + "th subject is"  
    + marks[i]);
```

{

```
System.out.println("S GPA" +  
    "S GPA");
```

{

```
class Student {
```

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

{

```
Student details[s = new  
student_details[3].
```

```
for (int i = 0; i < 3; i++)
```

```
s[i].getDetails();  
s[i].S GPA.. calculate();  
s[i].display();
```

{

{

Enter the usn

1B123CS055

Enter the name

Aarpit Sharma

Enter the marks of all 8
subjects

58

58

93

91

77

61

66

65.

✓
USN: 1B123CS055

Name: Aarpit Sharma

marks of 1st subject : 58

" " 2nd " : 58

" " 3rd " : 93

" " 4th " : ~~90~~ 91

" " 5th " : 77

" " 6th " : 61

" " 7th " : 66

" " 8th " : 65

SUMPA: 7.5

Output :-

Enter the usn
IBN23CSOS7

Enter the name:
Aayush Aditya

Enter the marks of all 8 subjects

93

95

92

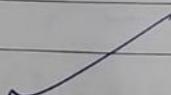
85

86

72

75

89



USN: IBN23CSOS7

Name: Aayush Aditya

Rank of 1st subject: 93

" " 2nd " : 95

" " 3rd " : 92

" " 4th " : 85

" " 5th " : 86

" " 6th " : 72

" " 7th " : 75

" " 8th " : 89

SUMPA = 9.25

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

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 nameStr = "Book name: " + this.name + "\n";
        String authorStr = "Author name: " + this.author + "\n";
        String priceStr = "Price: " + this.price + "\n";
        String numPagesStr = "Number of pages: " + this.numPages + "\n";
        return nameStr + authorStr + priceStr + numPagesStr;
    }

    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int n = s.nextInt();
        s.nextLine(); // Consume the newline after the integer input
        Books[] b = new Books[n];

        for (int i = 0; i < n; i++) {
            String name = s.nextLine();
            String author = s.nextLine();
            int price = s.nextInt();
            int numPages = s.nextInt();
            s.nextLine(); // Consume the newline after the integer input

            b[i] = new Books(name, author, price, numPages);
            String result = b[i].toString();
            System.out.println(result);
        }

        s.close(); // Close the scanner
    }
}
```

Output-

```
C:\Users\bmsce>cd Desktop

C:\Users\bmsce\Desktop>cd javac Books.java
The system cannot find the path specified.

C:\Users\bmsce\Desktop>javac Books.java

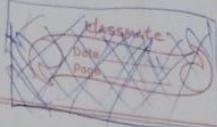
C:\Users\bmsce\Desktop>java Books
2
The iron Wall
Avi Shlaim
809
972
Book name: The iron Wall
Author name: Avi Shlaim
Price: 809
Number of pages: 972

Killing Hope
William Blum
1209
536
Book name: Killing Hope
Author name: William Blum
Price: 1209
Number of pages: 536
```

```
C:\Users\bmsce\Desktop>Name : Ayush Aditya           USN : 1BM23CS057
```

Observation-

23-10-24



Lab Program no 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 `toString()` method that could display the complete details of the book. Develop a Java program to create n book objects.

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

```
class Books {
```

```
    String name, author;  
    int numPages, price;
```

```
    Books (String name,  
           String author,  
           int numPages  
           int price)
```

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

```
    }  
    public String toString()
```

```
        String name, author,  
              price, numPages;
```

price = s.unentInt();

num pages = s.unentInt();

b[i] = new Books(num,
author, price,
num pages);

~~b[i]~~

string res = b[i].toString();

System.out.println(res);

}

3

Output :-

Enter the number of Book Entries:

2

Enter the details

Killing Hope

William Blum

500

436

Book name: Killing Hope

Author name: William Blum

price: 5 00

Number of pages: 436

Enter the ~~the~~ details

The iron wall

Awshawn

809
972

Book name: The Iron Will
Author name: N.W. Sharmin

Price: 809

Number of pages: 256

✓ Hm

3
E

4. Develop a Java program to create an abstract class name 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.

Code-

```
import java.util.Scanner;
abstract class Shape {
    int dim1;
    int dim2;
    public Shape() {
        this.dim1 = 0;
        this.dim2 = 0;
    }
    public Shape(int dim1, int dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }
    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int length, int width) {
        dim1 = length;
        dim2 = width;
    }
    public void printArea() {
        int area = dim1 * dim2;
        System.out.println("Area of Rectangle: " + area);
    }
}
class Triangle extends Shape {
    public Triangle(int base, int height) {
        dim1 = base;
        dim2 = height;
    }
    public void printArea() {
        double area = 0.5 * dim1 * dim2;
        System.out.println("Area of Triangle: " + area);
    }
}
class Circle extends Shape {
    public Circle(int radius) {
        dim1 = radius;
        dim2 = 0;
    }
    public 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 in = new Scanner(System.in);
        System.out.println("This code is written by Ayush Aditya");
        System.out.println("Enter length and width for Rectangle:");
    }
}
```

```

int length = in.nextInt();
int width = in.nextInt();
Shape rectangle = new Rectangle(length, width);
rectangle.printArea();
System.out.println("Enter base and height for Triangle:");
int base = in.nextInt();
int height = in.nextInt();
Shape triangle = new Triangle(base, height);
triangle.printArea();
System.out.println("Enter radius for Circle:");
int radius = in.nextInt();
Shape circle = new Circle(radius);
circle.printArea();
in.close();
}
}

```

Output-

```

Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

```

```
C:\Users\bmsce>cd Desktop
```

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

```
C:\Users\bmsce\Desktop>java Shapes
This code is written by Ayush Aditya
Enter length and width for Rectangle:
```

```
4
```

```
4
```

```
Area of Rectangle: 16
```

```
Enter base and height for Triangle:
```

```
2
```

```
7
```

```
Area of Triangle: 7.0
```

```
Enter radius for Circle:
```

```
3
```

```
Area of Circle: 28.274333882308138
```

```
C:\Users\bmsce\Desktop>3
```

```
'3' is not recognized as an internal or external command,
operable program or batch file.
```

```
C:\Users\bmsce\Desktop>Name : Ayush Aditya
```

```
USN : IBM23CS057
```

Observation-

23-10

Lab Program no 4

classmate

Date _____

Page _____

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

import java.util.Scanner;

abstract class Shape {

int dim1;

int dim2;

public Shape () {

this.dim1 = 0;

this.dim2 = 0;

}

public Shape (int dim1,

int dim2) {

this.dim1 = dim1;

this.dim2 = dim2;

}

public ~~abstract~~ void printArea () {

3

class Rectangle extends Shape {

 public Rectangle (int length,
 int width) {

 dim1 = length;
 dim2 = width;

 }

 public void printArea () {

 int area = dim1 * dim2;

 System.out.println ("Area
 of Triangle" +
 area);

}

3

class Circle extends Shape {

 public Circle (int radius) {

 dim1 = radius;

 dim2 = 0.

 }

 public void printArea () {

 double area = Math.PI *

 dim1 * dim1.

 System.out.println ("Area of
 Circle : " + area);

3

3

triangle.printArea();
System.out.println("15 cm
radius for circle:
int radius = cm.nextInt());

shape.circle = new Circle
(radius);

circle.printArea();
cm.close();

{

{

Output:

This code is written by Ayush
Acharya.

Enter length and width of
rectangle.

4

F



Area of Rectangle is 18

Enter Base and height of
triangle is

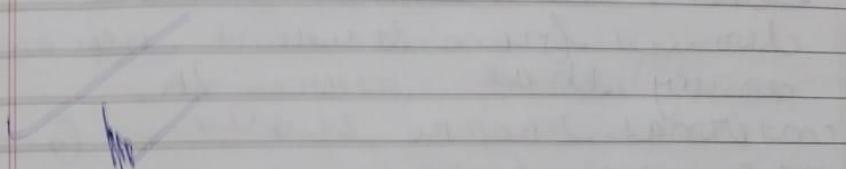
4

9

Area of triangle is 18

Find the radius of circle
2

Area of Circle is 12.56 sq cm



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

```
Code- import java.util.Scanner;
class Account {
    String customerName;
    String accountNumber;
    double balance;
    String accountType;
    public Account(String customerName, String accountNumber, String accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0.0;
    }
    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: $" + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }
    public void displayBalance() {
        System.out.println("Account Balance: $" + balance);
    }
    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: $" + amount);
        } else {
            System.out.println("Insufficient balance or invalid withdrawal amount.");
        }
    }
}
class CurAcct extends Account {
    static final double MIN_BALANCE = 500.0;
```

```

static final double SERVICE_CHARGE = 20.0;

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

public void checkAndImposeServiceCharge() {
    if (balance < MIN_BALANCE) {
        balance -= SERVICE_CHARGE;
        System.out.println("Balance below minimum. Service charge of $" + SERVICE_CHARGE + " imposed.");
    }
}

class SavAcct extends Account {
    static final double INTEREST_RATE = 0.05;

    // Constructor
    public SavAcct(String customerName, String accountNumber) {
        super(customerName, accountNumber, "Savings");
    }

    // Method to compute and deposit interest
    public void computeAndDepositInterest() {
        double interest = balance * INTEREST_RATE;
        balance += interest;
        System.out.println("Interest of $" + interest + " has been added to your savings.");
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Account acc1 = new CurAcct("John Doe", "12345");
        Account acc2 = new SavAcct("Jane Smith", "67890");
        while (true) {
            System.out.println("\n--- Bank Menu ---");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Display Balance");
            System.out.println("4. Compute Interest (Savings Account)");
            System.out.println("5. Check Service Charge (Current Account)");
            System.out.println("6. Exit");
            System.out.print("Enter choice: ");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter account type (Current/Savings): ");
                    String accountType = scanner.next();
                    System.out.print("Enter deposit amount: ");
                    double depositAmount = scanner.nextDouble();
                    if (accountType.equalsIgnoreCase("Current")) {
                        acc1.deposit(depositAmount);
                    } else if (accountType.equalsIgnoreCase("Savings")) {
                        acc2.deposit(depositAmount);
                    } else {
                        System.out.println("Invalid account type.");
                    }
                    break;
                case 2:

```

```

        System.out.print("Enter account type (Current/Savings): ");
        accountType = scanner.next();
        System.out.print("Enter withdrawal amount: ");
        double withdrawalAmount = scanner.nextDouble();
        if (accountType.equalsIgnoreCase("Current")) {
            acc1.withdraw(withdrawalAmount);
        } else if (accountType.equalsIgnoreCase("Savings")) {
            acc2.withdraw(withdrawalAmount);
        } else {
            System.out.println("Invalid account type.");
        }
        break;
    case 3:
        System.out.print("Enter account type (Current/Savings): ");
        accountType = scanner.next();
        if (accountType.equalsIgnoreCase("Current")) {
            acc1.displayBalance();
        } else if (accountType.equalsIgnoreCase("Savings")) {
            acc2.displayBalance();
        } else {
            System.out.println("Invalid account type.");
        }
        break;
    case 4:
        if (acc2 instanceof SavAcct) {
            ((SavAcct) acc2).computeAndDepositInterest();
        } else {
            System.out.println("Interest can only be computed for Savings accounts.");
        }
        break;
    case 5:
        if (acc1 instanceof CurAcct) {
            ((CurAcct) acc1).checkAndImposeServiceCharge();
        } else {
            System.out.println("Service charge can only be applied to Current accounts.");
        }
        break;
    case 6:
        System.out.println("Exiting the program.");
        scanner.close();
        return;
    default:
        System.out.println("Invalid choice. Please try again.");
    }
}
}
}
Output-

```

```
Command Prompt - java Ban + 
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Desktop
C:\Users\Admin\Desktop>javac Bank.java
C:\Users\Admin\Desktop>java Bank

--- Bank Menu ---
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account)
5. Check Service Charge (Current Account)
6. Exit
Enter choice: 1
Enter account type (Current/Savings): Current
Enter deposit amount: 50000
Deposited: $50000.0

--- Bank Menu ---
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account)
5. Check Service Charge (Current Account)
6. Exit
Enter choice: 2
Enter account type (Current/Savings): Current
Enter withdrawal amount: 30000
Withdrew: $30000.0

--- Bank Menu ---
1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account)
5. Check Service Charge (Current Account)
6. Exit
Enter choice: |
```

Observation-

classmate
Date _____
Page _____

```

import java.util.Scanner;

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

    public Account (String customerName,
                    String accountNumber,
                    String accountType) {
        this.customerName =
            customerName;
        this.accountNumber =
            accountNumber;
        this.accountType =
            accountType;
        this.balance = 0;
    }

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

```

class Current extends Account {

 static final double MIN_BALA
 NCE = 500.0;

 static final double

 SERVICE_CHARGE = 20.0;

 public Current (String customer
 Name, String
 accountNumber) {

 super (customerName,
 accountNumber,
 "Current");

}

 public void checkAndImpose
 servicecharge () {

 if (balance < MIN_BALA
 NCE) {

 balance -= SERVICE
 - CHARGE;
 }

✓

System.out.println ("By
 - luru below

minimum - service

charge of \$" +

 SERVICE - CHARGE)

 " imposed.");

}

3

3

class SavAcc extends Account {

 static final double

 INTEREST_RATE = 0.05;

public SavAcct(Strong accountNumber)
 Strong accountNumber;
 Super(cust accountName,
 accountNumber,
 "45 savings");

3

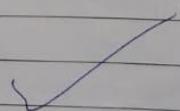
public void computeAndDeposit
 Interest();

double interest = balance
 * INTEREST RATE;

balance += interest;
 System.out.println("Interest
 of \$" + interest +

" has been added to
 your savings");

3



public class Bank {

public static void main(Strong[] args) {

Scanner sc = new
 Scanner(System.in);

Accounts acc = new
 Accounts("John Doe",
 "12345");

Account acc = new SavAcc
 ("Jane Smith",
 "67890");

while(true) {

System.out.println("1. Bank
 Rec - ");

System.out.println("2.
 Deposit");

System.out.println("3.
 Withdraw");

System.out.println("4.
 Display Balance");

System.out.println("5.
 Compute Interest (Savings
 Account)");

System.out.println("6. Exit
 Service Charge (Current
 Account)");

System.out.println("6. Exit");

System.out.println("Enter
 choice: ");

int choice = Scanner.nextInt();

switch(choice) {

case 1:

System.out.println("Enter
 account type
 (Current/Savings): ");

String accountType
 = Scanner.

nextInt();

System.out.println ("Enter
deposit amount");

double depos + Amount
= scanner.nextInt();
if (amountType.equals
("Current")) {
acc1.deposit (deposit
Amount);

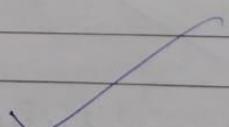
} else if (amountType.equals
("Savings")) {
acc2.deposit (deposit
Amount);

} else {

System.out.println ("Invalid
account type.");

}

break



case 2:

System.out.print ("Enter
amount type (Current/
Savings): ");
amountType = scanner.next();

double withdrawlAmount;
System.out.print ("Enter
withdrawl amount");

double withdrawlAmount
= scanner.nextDouble();

Date _____
Page _____

if (amountType.equals("savings")) {
 acc1.withdraw (withdrawAmount);
}

} else if (amountType.equals("savings")) {
 acc2.withdraw (withdrawAmount);
}

} else {

System.out.println ("Invalid
amount type.");

}

break;

case 3:

System.out.print ("Enter account
type (Current / Savings): ");
amountType = scanner.next();
if (amountType.equals("savings")) {
 case ("current") {
 acc1.displayBalance();
 } else if (amountType.equals("savings")) {
 acc2.displayBalance();
 } else {

System.out.println ("Invalid
amount type.");

}

break;

case 4:

if (acc2 instanceof SavingsAccount) {

((SavAcct) ac1). computeAmr
DepositInterest();

3 else {

System.out.println("Interest
can only be computed
for savings accounts");

}

break;

case 5:

if (ac1 instanceof CurrentAcct)

((CurrentAcct) ac1). check
varImpressService
charge();

3 else {

System.out.println("Service
charge was only
be applied to
current accounts.");

3

break;

case 6:

System.out.println("Exiting
th, program.");
Scanner.close();
return;

default:

System.out.println("Invalid
choice. Please try again.");

3

}

}

}

Output

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account)
5. Check Service Charge (Current Account)
6. EXIT.

Enter choice: 1

Enter amount type (Current/
Savings): current

Enter deposit amount: 50000

Deposited: \$ 50000.0

-- Bank Menu --

1. Deposit
2. Withdraw
3. Display Balance
4. Compute Interest (Savings Account)
5. Check Service Charge / Current Account
6. EXIT.

Enter choice: 2

Enter amount & type (Current / savings): Current

Enter withdraw amount: 3000

Withdraw : \$ 3000.0 .

✓
HHR

6. Create a package CIE which has two classes- Student and 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 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.

Code-

```

package CIE;

public class Student {
    protected String usn;
    protected String name;
    protected int sem;

    // Constructor to initialize student details
    public Student(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }

    // Display student details
    public void displayStudentInfo() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

public class Internals extends Student {
    protected int[] internalMarks;
    public Internals(String usn, String name, int sem, int[] internalMarks) {
        super(usn, name, sem);
        this.internalMarks = internalMarks;
    }
    public void displayInternalMarks() {
        System.out.println("Internal Marks for the student:");
        for (int i = 0; i < internalMarks.length; i++) {
            System.out.println("Course " + (i+1) + ": " + internalMarks[i]);
        }
    }
}

package SEE;
import CIE.Student;
public class External extends Student {
    protected int[] externalMarks;
    public External(String usn, String name, int sem, int[] externalMarks) {
        super(usn, name, sem);
        this.externalMarks = externalMarks;
    }
    public void displayExternalMarks() {
        System.out.println("External Marks for the student:");
        for (int i = 0; i < externalMarks.length; i++) {
            System.out.println("Course " + (i+1) + ": " + externalMarks[i]);
        }
    }
}

```

```

        }
    }
}

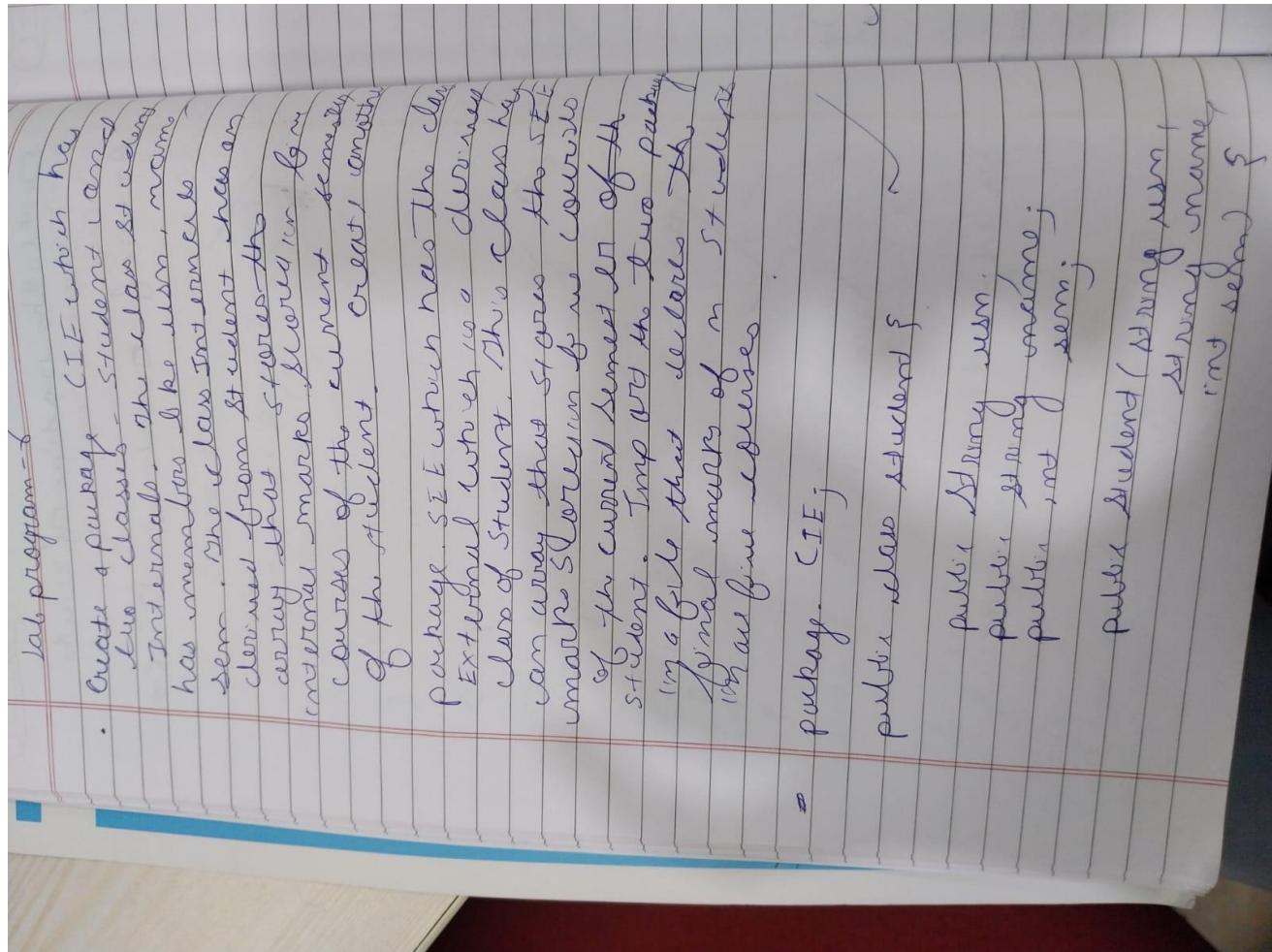
import CIE.Internals;
import SEE.External;
import java.util.Scanner;

public class StudentMarks {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int n = scanner.nextInt();
        scanner.nextLine();
        Internals[] internalsArray = new Internals[n];
        External[] externalsArray = new External[n];
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Student " + (i+1));
            System.out.print("USN: ");
            String usn = scanner.nextLine();
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Semester: ");
            int sem = scanner.nextInt();
            scanner.nextLine();
            int[] internalMarks = new int[5];
            System.out.println("Enter internal marks for 5 courses:");
            for (int j = 0; j < 5; j++) {
                internalMarks[j] = scanner.nextInt();
            }
            scanner.nextLine();
            internalsArray[i] = new Internals(usn, name, sem, internalMarks);
            int[] externalMarks = new int[5];
            System.out.println("Enter external marks (SEE) for 5 courses:");
            for (int j = 0; j < 5; j++) {
                externalMarks[j] = scanner.nextInt();
            }
            scanner.nextLine();
            externalsArray[i] = new External(usn, name, sem, externalMarks);
        }
        System.out.println("\n--- Final Marks of Students ---");
        for (int i = 0; i < n; i++) {
            System.out.println("\nDetails of Student " + (i+1));
            internalsArray[i].displayStudentInfo();
            internalsArray[i].displayInternalMarks();
            externalsArray[i].displayExternalMarks();
            int totalInternalMarks = 0;
            for (int mark : internalsArray[i].internalMarks) {
                totalInternalMarks += mark;
            }
            int totalExternalMarks = 0;
            for (int mark : externalsArray[i].externalMarks) {
                totalExternalMarks += mark;
            }
            int finalMarks = totalInternalMarks + totalExternalMarks;
            System.out.println("Final Marks: " + finalMarks);
        }
        scanner.close();
    }
}

```

}

Observation-



this . name = name;
this . surname = surname;
this . year = year; /

public void displayStudentInfo() {
System.out.println("USN: " +
name);

System.out.println("Name: " +
name);
System.out.println("Year: "
+ year);

}

~~public class ExternalAndInner
Student {~~
public class [] internalMarks {

public class InternalAndExternal
Student {
public InternalAndExternalMarks {
String name, int year,
int [] internalMarks {
int [] externalMarks {

super (name, year);
this . internalMarks :
year; externalMarks :
year;

int totalMarks ;

}

public void displayInternalMarks() {
System.out.println("Internal
Marks after the Student");
for (int i = 0; i < internalMarks;
i++) {

System.out.println(" ");

System.out.println("Loungh");
} +
{
 int[] marks = {
 100,
 100,
 100
 };

}

}

}

package SEE;
import java.util;

public class ExternalMethods

Student &

marks, External (String name,

String marks, int sum,

int totalMarks);

super (name, marks, sum);

this . externalMethods (

marks);

} public void displayExternalMethod()

System.out.println ("External

Works for the Student : ");

for (int i = 0; i < externalMarks

length ; i++) {

System.out . out . println (marks

+ (i + 1) + " : " +

marks [marks.length - 1]);

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

Component CSE. External

Component SEE. External

Component Java. Util. Scanner;

public class Studentmarks{

public static void main (String [] args){

Scanner scanner = new Scanner (System.

in);

System.out.println ("Enter the
number of students : ");

int n = scanner.nextInt ();

int marks [] = new int [n];

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

System.out.print ("Enter the marks
of student " + i + " : ");

marks [i] = scanner.nextInt ();

System.out.println ("USN : ");

String name = scanner.nextLine ();

System.out.println ("Name : ");

String name = scanner.nextLine ();

char sex = scanner.next().charAt (0);

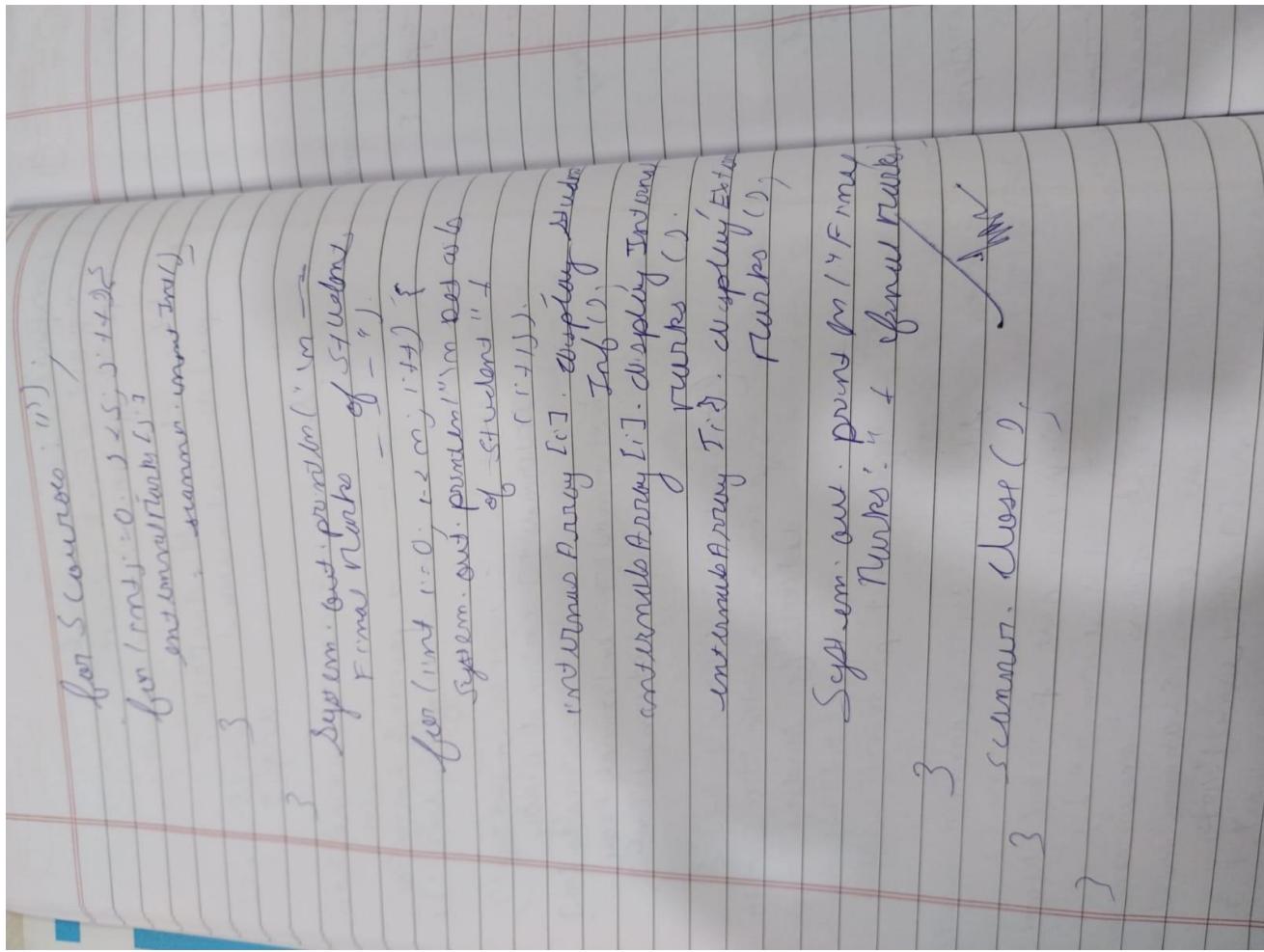
if (sex == 'M') {

System.out.println ("Male");

else {

System.out.println ("Female");

scanner.close ();



7. Write a program that demonstrate handling of exception in inheritance String. Create a base class called Father and a derived class called son which extends the base class in Father's class implement a constructor that takes the age and throws exception Wrong age when input age is less than 0. In son's class implements constructor uses both father's and son's age and throws an exception when son's age \geq father's age.

Code-

```

class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

class Father {
    protected int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Age cannot be less than 0.");
        }
        this.age = age;
        System.out.println("Father's age: " + this.age);
    }
}

class Son extends Father {
    private int sonAge;
}

```

```

public Son(int fatherAge, int sonAge) throws WrongAgeException {
    super(fatherAge);
    if (sonAge < 0) {
        throw new WrongAgeException("Son's age cannot be less than 0.");
    }
    if (sonAge >= fatherAge) {
        throw new WrongAgeException("Son's age cannot be greater than or equal to father's age.");
    }
    this.sonAge = sonAge;
    System.out.println("Son's age: " + this.sonAge);
}
}

public class Main {
    public static void main(String[] args) {
        try {
            Father father = new Father(-5);
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
        try {
            Son son = new Son(40, 45);
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
        try {
            Son son = new Son(40, -10);
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
        try {
            Son son = new Son(40, 20);
        } catch (WrongAgeException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

Output-

```
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Desktop

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

C:\Users\Admin\Desktop>java Main
Error: Age cannot be less than 0.
Father's age: 40
Error: Son's age cannot be greater than or equal to father's age.
Father's age: 40
Error: Son's age cannot be less than 0.
Father's age: 40
Son's age: 20

C:\Users\Admin\Desktop>Ayush Aditya           1BM23CS057|
```

Observation-

2 Write a program that demonstrates handing of exception in

inheritance. Strongly create a base class called Father and a derived class called son which extends the base class. In Father's class implement a constructor that takes the age and throws exception.

Wrong age when input age

less than 0. In son's class implement a constructor that uses both father's and son's age and throws an exception when son's age >= father's age.

> class WrongAgeException extends

Exception {

public WrongAgeException(String
message) {
super(message);
}

3

class Father {

protected int age;

public Father(int age) throws

WrongAgeException {

if (age < 0) {

throw new WrongAge

Exception("Age cannot

be less than 0."),

3

this.age = age;
System.out.println("Father's

age: " + this.age);

3

3

class Son extends Father {

private int SonAge;

public Son (int fatherAge,
int sonAge) throws

wrongAge Exception{

super(fatherAge);

if (sonAge < 0) {

throws new wrongAge

Exception ("Son's age
cannot be less than

3

if (sonAge >= fatherAge) {

throws new

throw new wrongAge

Exception ("Son's age

cannot be greater than

or equal to father's

3

this.SonAge = sonAge;

System.out.println("Son's

age: " + this.son.age);

{

{

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

try {

```
Father father = new
Father(-5).
```

```
} catch (WrongAgeException
e) {
```

```
System.out.println("Error
+ e.getMessage());
```

{

try {

```
Son son = new Son(10, -10)
```

```
} catch (WrongAgeException e)
```

```
System.out.println("Error
+ e.getMessage());
```

{

try {

```
Son son = new Son(10, -10)
```

```
} catch (WrongAgeException e)
```

```
System.out.println("Error
+ e.getMessage());
```

{

try {
 son & son = new Son(40,
 20);

} catch (ArithException e)

System.out.println("Error")

+ e.printStackTrace();

}

}

}

Output -

Error: Age cannot be less than 0

Father's age: 40

Error: Son's age cannot be greater than or equal to

Father's age: 40

Error: Son's age cannot be less than 0.

Father's : 40

Son's age : 20

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.

Code-

```
public class MultiThreadExample {  
    static class DisplayBMS implements Runnable {  
        @Override  
        public void run() {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                try {  
                    Thread.sleep(10000); // Sleep for 10 seconds  
                } catch (InterruptedException e) {  
                    Thread.currentThread().interrupt(); // Handle thread interruption  
                }  
            }  
        }  
    }  
    static class DisplayCSE implements Runnable {  
        @Override  
        public void run() {  
            while (true) {  
                System.out.println("CSE");  
                try {  
                    Thread.sleep(2000); // Sleep for 2 seconds  
                } catch (InterruptedException e) {  
                    Thread.currentThread().interrupt(); // Handle thread interruption  
                }  
            }  
        }  
    }  
    public static void main(String[] args) {  
        // Create two thread objects for each task  
        Thread thread1 = new Thread(new DisplayBMS());  
        Thread thread2 = new Thread(new DisplayCSE());  
        thread1.start();  
        thread2.start();  
    }  
}
```

Output-

```
Microsoft Windows [Version 10.0.22631.4460]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\Admin>cd Desktop  
C:\Users\Admin\Desktop>javac MultiThreadExample.java  
C:\Users\Admin\Desktop>Ayush Aditya IBM23CS057  
'Ayush' is not recognized as an internal or external command,  
operable program or batch file.  
C:\Users\Admin\Desktop>java MultiThreadExample  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
|
```

Lab : program no 3

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

public class MultiThreadExample extends Thread {

static class DisplayBTS implements Runnable {

public void run () {

while (true) {

System.out.println ("BITS
College of Engineering");

try {

Thread.sleep (10000);

} catch (InterruptedException e) {

Thread.currentThread().
interrupt();

static class DisplayCSE implements Runnable {

public void run () {

while (true) {

System.out.println("CSE
try {
 Thread.sleep(2000)

} catch(InterruptedException e)
 Thread.currentThread().
 interrupt();

}

}

public static void main(String[] args) {

 Thread

 DisplayBTS t = new
 DisplayBTS();

 Display CSE thread 2 - new
 Display (S.E());

 thread 1. start();

 thread 2. start();

}

}

Our put

CSE
BTS College of Engineering
CSE
CSE

CS
CS
BM

CSE

CSE

BMS college of Engineering

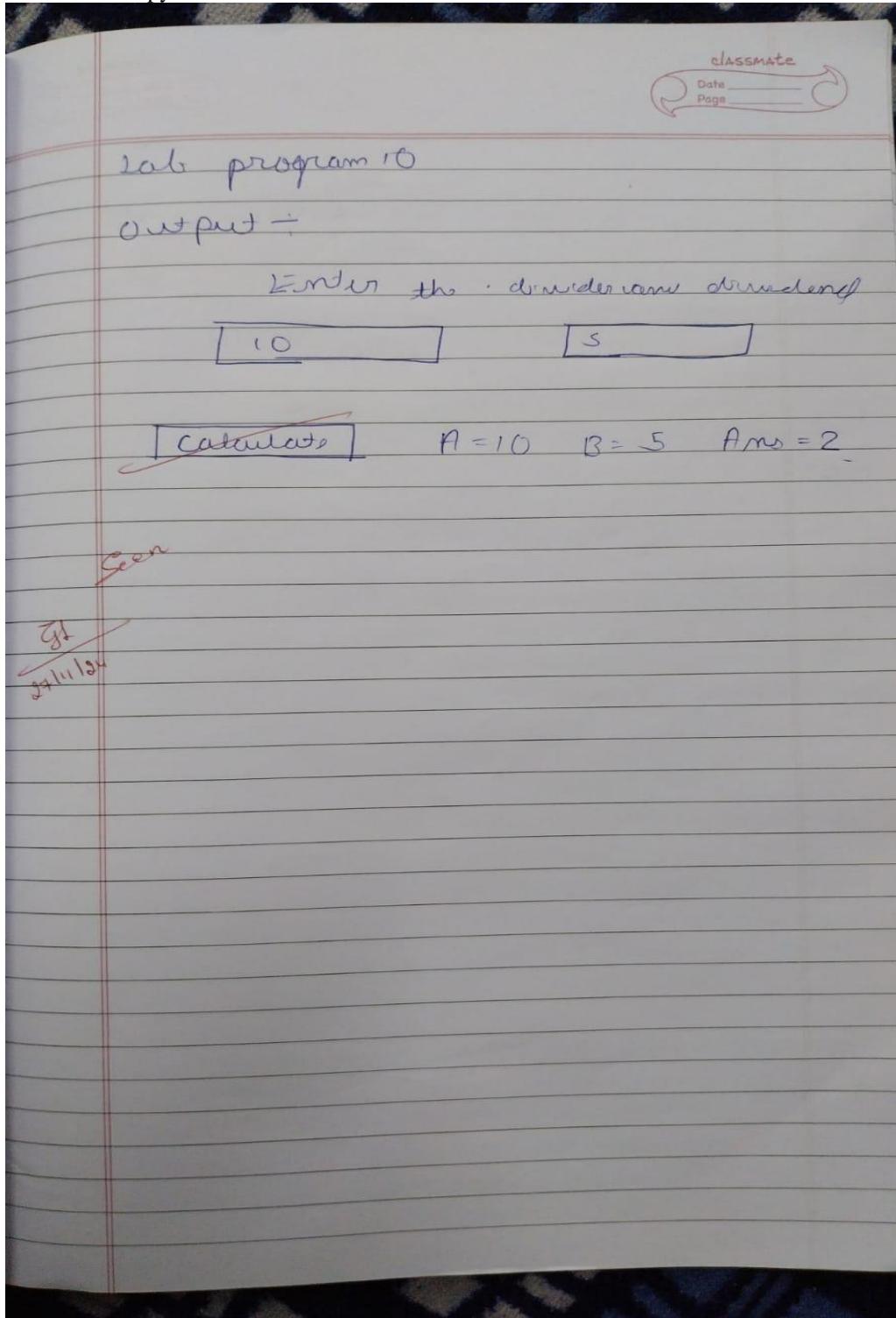
Date _____

Page _____

- 1. down
- 2. sum of numbers
- 3. sum of terms
- 4. sum of consecutive numbers
- 5. sum of first n natural numbers
- 6. sum of first n odd numbers
- 7. sum of first n even numbers
- 8. sum of first n squares
- 9. sum of first n cubes
- 10. sum of first n square numbers
- 11. sum of first n cube numbers

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.

Observation copy-



10. Write a Java program to create two threads, one displays “Computer Science” and another display “Information Science” five times.

Observation-

