

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



LAB REPORT
on
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

LIKITH.TARAKARAM(1BM23CS171)

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 **StudentName (1BM23CS000)**, 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.

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

Index

Sl. No.	Date	Experiment Title	Page No.
1	30/09/24	QUADRATIC EQUATION	4-6
2	07/10/24	STUDENT SGPA	7-11
3	14/10/24	Book Details	12-15
4	21/10/24	Area of the Shape	16-19
5	28/10/24	Bank	19-29
6	11/11/24	Package	29-32
7	28/11/24	Exception handling	33-36
8	28/11/24	Threads	37-39
9	28/11/24	Swing Demo	40-42
10	28/11/24	10A.Deadlock 10B.PCFixed	43-52

Github Link:

<https://github.com/Likith200504/javalab>

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-

classmate
Date _____
Page _____

Lab Program - 3

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

```

import java.util.*;
class Quadratic
{
    int a,b,c;
    double r1,r2,d;
    void getd()
    {
        System.out.println("Enter three values:");
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
        if (a==0)
            System.out.println("non-blank state value");
        if (a==0)
            System.out.println("Enter non-zero value of a");
        a = s.nextInt();
    }
    b = s.nextInt();
    c = s.nextInt();
}

void compute()
{
    d = b*b - 4*a*c;
    if (d==0)
    
```

classmate
Date _____
Page _____

```

    {
        System.out.println("Input values");
        r1 = (-b) / (2 * a);
        r2 = r1;
        System.out.println("Roots are " + r1 + " and " + r2);
    }
    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 " + r1 + " and " + r2);
    }
    else
    {
        System.out.println("Roots are imaginary");
    }
}
else
{
    System.out.println("Input values");
}

class QuadraticEquation
{
    public static void main (String [] args)
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
        System.out.println("Likith.Taravaram");
        System.out.println("B.M23(S17)");
    }
}

```

classmate
Date _____
Page _____

O/P

→ Enter three values:
 1
 -2
 -3
 roots are 7.4051 and -0.4051
 Likhith.Tarakaram
 1BM23CS171

→ Enter three values:
 1
 -4
 9
 roots are 2.0 & -2.0
 Likhith.Tarakaram
 1BM23CS171

→ Enter three values:
 2
 1
 1
 quadratic equation has imaginary
 roots
 Likhith.Tarakaram
 1BM23CS171

~~Q1~~ ~~Q2~~

1. 1BM23CS171 Moring Bano (Ans) 7/14
 (1) 2nd year 2. M23

CODE-

```
import java.util.*;
class Quadratic
{
    int a,b,c;
    double r1,r2,d;
    void getd()
    {
        System.out.println("enter three values:");
        Scanner s=new Scanner(System.in);
        a=s.nextInt();
        if(a==0)
        {
            System.out.println("enter non zero value of a");
            a=s.nextInt();
        }
        b=s.nextInt();
        c=s.nextInt();
    }
    void compute()
    {
        d=b*b-4*a*c;
        if(d==0)
        {
            r1=(-b)/(2*a);
            r2=r1;
            System.out.println("roots are"+r1+"and"+r2);
        }
        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"+r1+"and"+r2);
        }
        else
        {
            System.out.println("quadratic equation has imaginary roots:");
        }
    }
}
class QuadraticEquation
{
    public static void main(String args[])
    {
        Quadratic q=new Quadratic();
        q.getd();
```

```

        q.compute();
        System.out.println("Likith.Tarakaram");
        System.out.println("1BM23CS171");
    }
}

```

OUTPUT-

Command Prompt

```

D:\1BM23CS171(JAVA)>javac QuadraticEquation.java

D:\1BM23CS171(JAVA)>java QuadraticEquation
enter three values:
1
-7
-3
roots are7.405124837953327and-0.405124837953327
Likith.Tarakaram
1BM23CS171

D:\1BM23CS171(JAVA)>java QuadraticEquation
enter three values:
1
-4
4
roots are2.0and2.0
Likith.Tarakaram
1BM23CS171

D:\1BM23CS171(JAVA)>java QuadraticEquation
enter three values:
2
1
1
quadratic equation has imaginary roots:
Likith.Tarakaram
1BM23CS171

```

Command Prompt

```

D:\1BM23CS171(JAVA)>javac QuadraticEquation.java

D:\1BM23CS171(JAVA)>java QuadraticEquation
enter three values:
1
-7
-3
roots are7.405124837953327and-0.405124837953327
Likith.Tarakaram
1BM23CS171

D:\1BM23CS171(JAVA)>java QuadraticEquation
enter three values:
1
-4
4
roots are2.0and2.0
Likith.Tarakaram
1BM23CS171

D:\1BM23CS171(JAVA)>java QuadraticEquation
enter three values:
2
1
1
quadratic equation has imaginary roots:

```

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-

Program-2

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

```
import java.util.*;
class Subject {
    int subjectmarks, credits, grade;
}
class Student {
    String name, usn;
    double SGPA;
    Subject[] subject;
    Scanner s;
    Student() {
        subject = new Subject[8];
        for (int i=0; i<8; i++) {
            subject[i] = new Subject();
        }
    }
    void getdetails() {
        System.out.println("Enter USN:");
        usn = s.nextLine();
    }
}
```

System.out.println("Enter name:");
name = s.nextLine();
System.out.println("Enter Subject marks and credits");
for (int i=0; i<8; i++) {
 System.out.println("Enter credits for subject " + (i+1) + ":");
 subject[i].credits = s.nextInt();
 System.out.println("Enter marks for subject " + (i+1) + ":");
 subject[i].subjectmarks = s.nextInt();
}

double compute() {
 double totalcredits = 0, totalgradepoints = 0;
 for (int i=0; i<8; i++) {
 totalcredits += subject[i].credits;
 totalgradepoints += (subject[i].subjectmarks / 10) * subject[i].credits;
 }
 return (totalgradepoints / totalcredits);
}

void display() {
 System.out.println("Name: " + name);
 System.out.println("USN: " + usn);
 System.out.println("Marks & Credits:");
 for (int i=0; i<8; i++) {
 System.out.print("Subject " + (i+1) + " Marks - " +
 subject[i].subjectmarks + ", Credits - " +
 subject[i].credits);
 }
}

classmate
Date _____
Page _____

```

System.out.println("SGPA: " + compute());
}

public static void main(String[] args)
{
    int n;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter no. of students:");
    n = sc.nextInt();
    Student std = new Student();
    for (int i = 0; i < n; i++)
    {
        std[i] = new Student();
    }
    for (int i = 0; i < n; i++)
    {
        std[i].getDetails();
        std[i].display();
    }
    System.out.println("Zikith.Tarakaram");
    System.out.println("IBN23CS171");
}

```

~~OP: Zikith.Tarakaram~~

Enter USN
~~IBN23CS172~~

Enter name: Shailesh

Enter subject marks and credits

Enter credits for subject 1: 4

Enter marks for subject 1: 99

classmate
Date _____
Page _____

Enter credits for subject 2: 4

Enter remarks for subject 2: 99

Enter credits for subject 3: 3

Enter marks for subject 3: 99

Enter credits for subject 4: 3

Enter marks for subject 4: 99

Enter credits for subject 5: 3

Enter marks for subject 5: 99

Enter credits for subject 6: 1

Enter marks for subject 6: 99

Enter credits for subject 7: 1

Enter marks for subject 7: 99

Enter credits for subject 8: 1

Enter marks for subject 8: 99

Name: Shailesh
USN: IBN23CS172
SUBJECT 1: Marks - 99, credits - 4
SUBJECT 2: Marks - 99, credits - 4
SUBJECT 3: Marks - 99, credits - 3
SUBJECT 4: Marks - 99, credits - 3

classmate
Date _____
Page _____

SUBJECT	Marks	Credits
5	99	3
6	99	1
7	99	1
8	99	1

SGPA: 9.0

ZIKITH.TARA KARAH
IBN23CS171

~~10/10~~

CODE-

```
import java.util.*;  
  
class Subject  
{  
    int subjectmarks, credits, grade;  
}  
  
class Student  
{  
    String name, usn;  
    double SGPA;  
    Subject[] subject;  
    Scanner s;  
  
    Student()  
    {  
        subject = new Subject[8];  
        for (int i = 0; i < 8; i++) {  
            subject[i] = new Subject();  
        }  
        s = new Scanner(System.in);  
    }  
  
    void getdetails()  
{  
        System.out.println("Enter USN:");  
        usn = s.nextLine();  
        System.out.println("Enter Name:");  
        name = s.nextLine();  
        System.out.println("Enter subject marks and credits");  
        for (int i = 0; i < 8; i++)  
        {  
            System.out.println("Enter credits for subject " + (i + 1) + ":");  
            subject[i].credits = s.nextInt();  
            System.out.println("Enter marks for subject " + (i + 1) + ":");  
            subject[i].subjectmarks = s.nextInt();  
        }  
    }  
  
    double compute()  
{  
        double totalcredits = 0, totalgradepoints = 0;  
        for (int i = 0; i < 8; i++)  
        {  
            totalcredits += subject[i].credits;
```

```

        totalgradepoints += (subject[i].subjectmarks / 10) * subject[i].credits;
    }
    return (totalgradepoints / totalcredits);
}

void display()
{
    System.out.println("Name: " + name);
    System.out.println("USN: " + usn);
    System.out.println("marks and credits:");
    for (int i = 0; i < 8; i++)
    {
        System.out.println("Subject " + (i + 1) + ": Marks - " + subject[i].subjectmarks + ", Credits - "
+ subject[i].credits);
    }
    System.out.println("SGPA: " + compute());
}

public static void main(String[] args)
{
    int n;
    System.out.println("enter the number of students");
    Scanner sc=new Scanner(System.in);
    n=sc.nextInt();
    Student[] std = new Student[2];
    for (int i = 0; i < n; i++)
    {
        std[i] = new Student();
    }
    for (int i = 0; i < n; i++)
    {
        std[i].getdetails();
        std[i].display();
    }
    System.out.println("LIKITH.TARAKARAM");
    System.out.println("1BM23CS171");
}
}

```

OUTPUT-

```
enter the number of students
2
Enter USN:
1BM23CS171
Enter Name:
LIKITH
Enter subject marks and credits
Enter credits for subject 1:
4
Enter marks for subject 1:
99
Enter credits for subject 2:
4
Enter marks for subject 2:
98
Enter credits for subject 3:
3
Enter marks for subject 3:
97
Enter credits for subject 4:
3
Enter marks for subject 4:
97
Enter credits for subject 5:
3
Enter marks for subject 5:
94
Enter credits for subject 6:
1
Enter marks for subject 6:
99
Enter credits for subject 7:
1
Enter marks for subject 7:
99
Enter credits for subject 8:
1
Enter marks for subject 8:
99
Name: LIKITH
USN: 1BM23CS171
marks and credits:
Subject 1: Marks - 99, Credits - 4
Subject 2: Marks - 98, Credits - 4
Subject 3: Marks - 97, Credits - 3
Subject 4: Marks - 97, Credits - 3
Subject 5: Marks - 94, Credits - 3
Subject 6: Marks - 99, Credits - 1
Subject 7: Marks - 99, Credits - 1
Subject 8: Marks - 99, Credits - 1
SGPA: 9.0
```

```
Enter USN:
1BM23CS172
Enter Name:
SHAILESH
Enter subject marks and credits
Enter credits for subject 1:
4
Enter marks for subject 1:
99
Enter credits for subject 2:
4
Enter marks for subject 2:
99
Enter credits for subject 3:
3
Enter marks for subject 3:
99
Enter credits for subject 4:
3
Enter marks for subject 4:
99
Enter credits for subject 5:
3
Enter marks for subject 5:
99
Enter credits for subject 6:
1
Enter marks for subject 6:
99
Enter credits for subject 7:
1
Enter marks for subject 7:
99
Enter credits for subject 8:
1
Enter marks for subject 8:
99
Name: SHAILESH
USN: 1BM23CS172
marks and credits:
Subject 1: Marks - 99, Credits - 4
Subject 2: Marks - 99, Credits - 4
Subject 3: Marks - 99, Credits - 3
Subject 4: Marks - 99, Credits - 3
Subject 5: Marks - 99, Credits - 3
Subject 6: Marks - 99, Credits - 1
Subject 7: Marks - 99, Credits - 1
Subject 8: Marks - 99, Credits - 1
SGPA: 9.0
LIKITH.TARAKARAM
1BM23CS171
```

Program 3

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

ALGORITHM-

LAB PROGRAM - 3

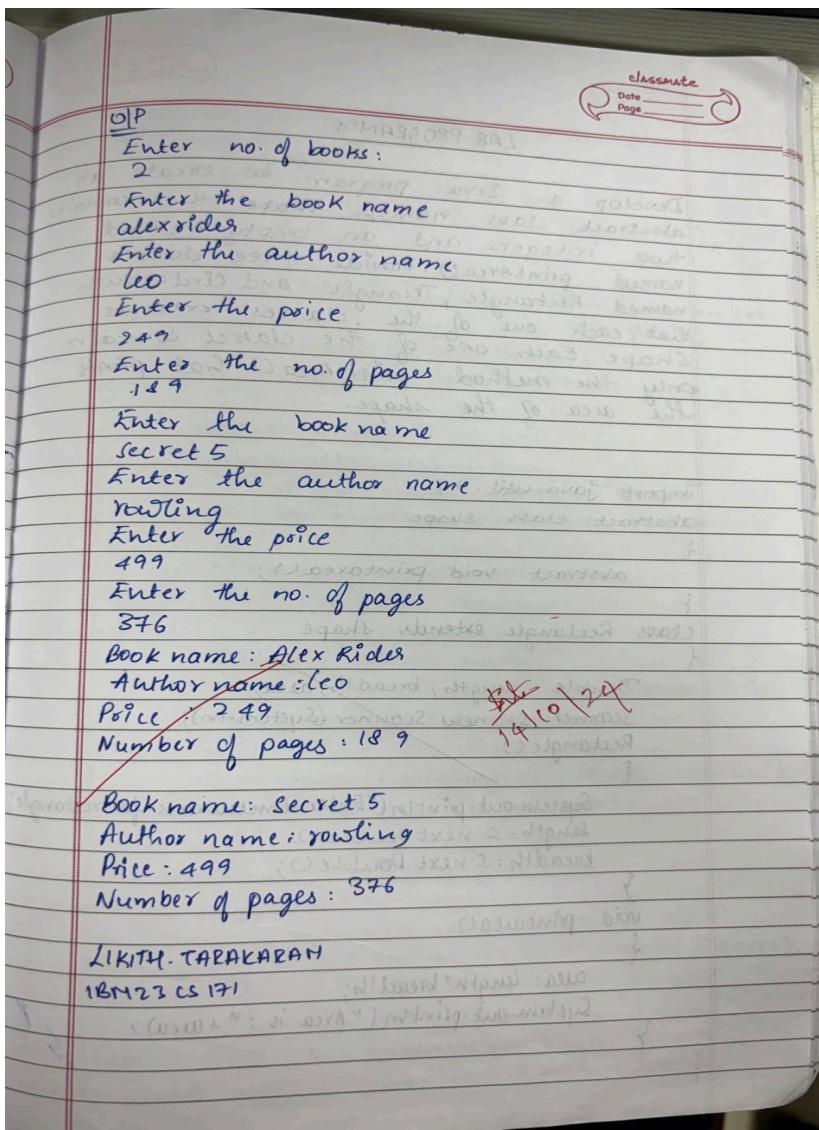
classmate
Date
Page

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

```
import java.util.*;
class 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 = "Book price: " + this.price + "\n";
        numpages = "Book No. of pages: " + this.numPages + "\n";
        return name + author + price + numpages;
    }
}
```

classmate
Date
Page

```
} class @details
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        int n;
        String name;
        String author;
        int price;
        int numpages;
        Books [] b = new Books [n];
        System.out.println("Enter the no. of books");
        n = s.nextInt();
        for(int i=0; i<n; i++)
        {
            System.out.println("Enter Book name:");
            name = s.nextLine();
            System.out.println("Enter author name");
            author = s.nextLine();
            System.out.println("Enter price:");
            price = s.nextInt();
            System.out.println("Enter no. of pages");
            numpages = s.nextInt();
            b[i] = new Books (name, author, price, numpages);
        }
        for(int i=0; i<n; i++)
        {
            System.out.println(b[i]);
            System.out.println("EKKITHI TALARAKARAN");
            System.out.println("IBM23CS171");
        }
    }
}
```



CODE-

```

import java.util.*;
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;
}
}

class details
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        int n;
        String name;
        String author;
        int price;
        int numPages;
        System.out.println("Enter the number of books:");
        n=s.nextInt();
        Books[] b=new Books[n];
        for (int i=0;i<n;i++)
        {
            System.out.println("Enter the book name");
            name=s.next();
            System.out.println("Enter the author name:");
            author=s.next();
            System.out.println("Enter the price:");
            price=s.nextInt();
            System.out.println("Enter the number of pages:");
            numPages=s.nextInt();
            b[i]=new Books(name,author,price,numPages);
        }
        for(int i=0;i<n;i++)
        {
            System.out.println(b[i]);
        }
        System.out.println("LIKITH.TARAKARAM");
        System.out.println("1BM23CS171");
    }
}

```

OUTPUT-

```
Command Prompt
D:\1BM23CS171(JAVA)>java details
Enter the number of books:
2
Enter the book name
alexrider
Enter the author name:
leo
Enter the price:
245
Enter the number of pages:
189
Enter the book name
secret5
Enter the author name:
rowling
Enter the price:
499
Enter the number of pages:
376
Book name:alexrider
Author name: leo
Price: 245
Number of pages: 189

Book name:secret5
Author name: rowling
Price: 499
Number of pages: 376

LIKITH.TARAKARAM
1BM23CS171
```

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.

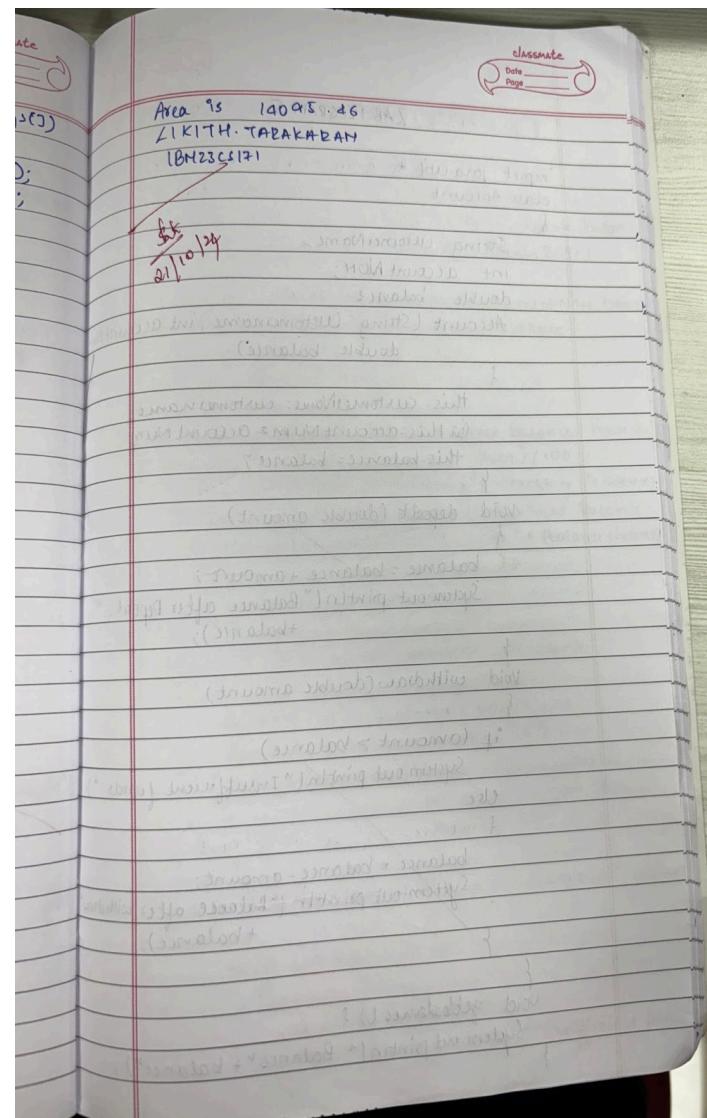
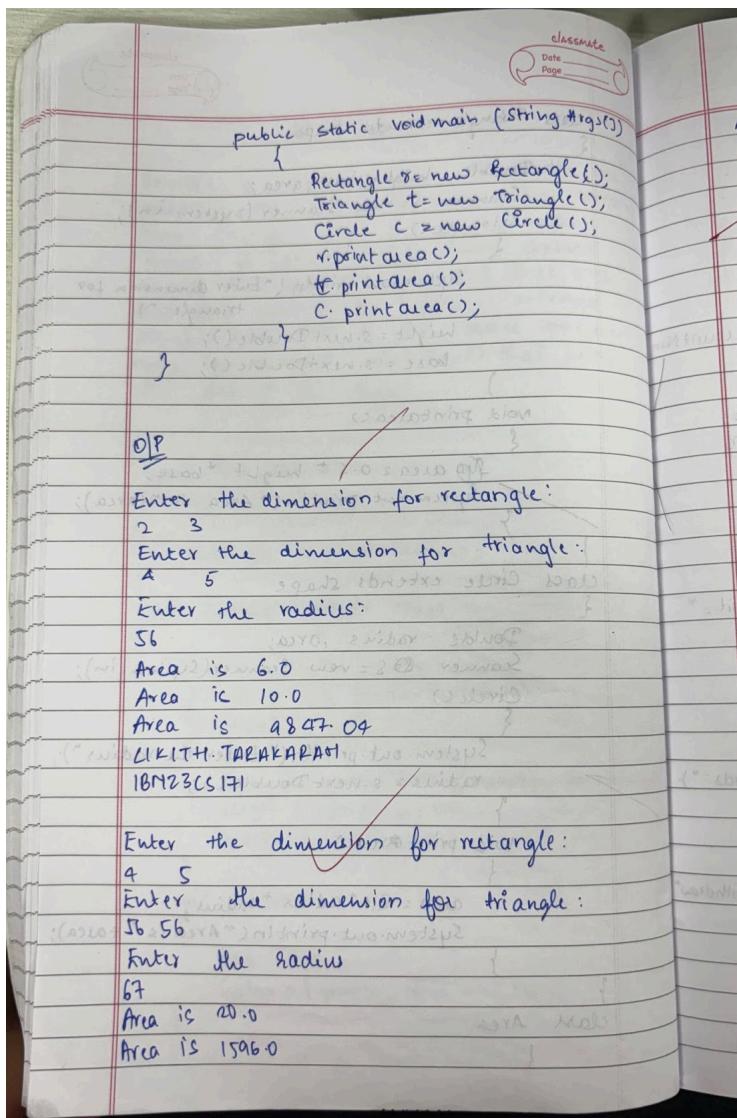
ALGORITHM-

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

```
import java.util.*;
abstract class Shape {
    abstract void printArea();
}
class Rectangle extends Shape {
    Double length, breadth, area;
    Scanner s = new Scanner(System.in);
    Rectangle() {
        System.out.println("Enter dimensions for rectangle");
        length = s.nextDouble();
        breadth = s.nextDouble();
    }
    void printArea() {
        area = length * breadth;
        System.out.println("Area is :" + area);
    }
}
```

class Triangle extends Shape {
 Double height, base, area;
 Scanner s = new Scanner(System.in);
 Triangle() {
 System.out.println("Enter dimension for triangle:");
 height = s.nextDouble();
 base = s.nextDouble();
 }
 void printArea() {
 area = 0.5 * height * base;
 System.out.println("Area is :" + area);
 }
}
class Circle extends Shape {
 Double radius, area;
 Scanner s = new Scanner(System.in);
 Circle() {
 System.out.println("Enter the radius");
 radius = s.nextDouble();
 }
 void printArea() {
 area = 3.14 * radius * radius;
 System.out.println("Area is :" + area);
 }
}
class Area



CODE-

```

import java.util.*;
abstract class shape
{
    abstract void printarea();
}
class Rectangle extends shape
{
    Double length,breadth,area;
    Scanner s=new Scanner(System.in);
    Rectangle()
    {
        System.out.println("Enter the dimensions for rectangle:");
        length=s.nextDouble();
        breadth=s.nextDouble();
    }
}

```

```

        }
    void printarea()
    {
        area=length*breadth;
        System.out.println("Area is "+area);
    }
}
class Triangle extends shape
{
    Double height,base,area;
    Scanner s=new Scanner(System.in);
    Triangle()
    {
        System.out.println("Enter the dimensions for triangle:");
        height=s.nextDouble();
        base=s.nextDouble();
    }
    void printarea()
    {
        area=0.5*base*height;
        System.out.println("Area is "+area);
    }
}
class Circle extends shape
{
    Double radius,area;
    Scanner s=new Scanner(System.in);
    Circle()
    {
        System.out.println("Enter the radius:");
        radius=s.nextDouble();
    }
    void printarea()
    {
        area=3.14*radius*radius;
        System.out.println("Area is "+area);
    }
}
class Area
{
    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("LIKITH.TARAKARAM");
        System.out.println("1BM23CS171");
    }
}

```

OUTPUT-

```

D:\1BM23CS171(JAVA)>javac Area.java
D:\1BM23CS171(JAVA)>java Area
Enter the dimensions for rectangle:
2 3
Enter the dimensions for triangle:
4 5
Enter the radius:
56
Area is 6.0
Area is 10.0
Area is 9847.04
LIKITH.TARAKARAM
1BM23CS171

D:\1BM23CS171(JAVA)>java Area
Enter the dimensions for rectangle:
4 5
Enter the dimensions for triangle:
56 57
Enter the radius:
67
Area is 20.0
Area is 1596.0
Area is 14095.46
LIKITH.TARAKARAM
1BM23CS171

```

Program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

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

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

ALGORITHM-

LAB PROGRAM 5

```
import java.util.*;
class Account {
    String customerName;
    int accountNum;
    double balance;
    Account (String customerName, int accountNum, double balance) {
        this.customerName = customerName;
        if (this.accountNum != accountNum)
            this.balance = balance;
    }
    void deposit (double amount) {
        balance = balance + amount;
        System.out.println ("Balance after Deposit " + balance);
    }
    void withdraw (double amount) {
        if (amount > balance)
            System.out.println ("Insufficient funds");
        else
            balance = balance - amount;
        System.out.println ("Balance after withdraw " + balance);
    }
    void getbalance () {
        System.out.println ("Balance " + balance);
    }
}
```

class SavingsAccount extends Account {
 double interestRate;
 SavingsAccount (String customerName, int accountNum, double balance, double interestRate) {
 super (customerName, accountNum, balance);
 this.interestRate = interestRate;
 }
 void calcInterest () {
 double interest = (super.balance * interestRate) / 100;
 System.out.println ("Interest is " + interest);
 System.out.println ("Your New Balance is " + (balance + interest));
 }
}

class CurrentAccount extends Account {
 double minBalance = 500.0;
 double serviceCharge = 50.0;
 CurrentAccount (String customerName, int accountNum, double balance) {
 super (customerName, accountNum, balance);
 }
 void withdraw (double amount) {
 if (amount > balance)
 System.out.println ("Insufficient funds");
 }
}

classmate
Date _____
Page _____

```

else if ((balance - amount) < MIN_BALANCE)
{
    System.out.println("Service charge will
    be imposed");
    balance = balance - service_charge - amount;
    System.out.println("Balance after
    service charge
    imposed = " + balance);
}
else
{
    balance = balance - amount;
    System.out.println("Balance after withdrawal
    = " + balance);
}

class Bank1
{
    public static void Main(String args[])
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Name, Ac
        No, Balance and
        interestrate");
        String customerName = s.next();
        int accountNum = s.nextInt();
        double balance = s.nextDouble();
        System.out.println("Select account type
        1. Savings
        2. Current");
        int opt = s.nextInt();
        boolean exit = false;
        if (opt == 1)
    }
}

```

classmate
Date _____
Page _____

```

SavingAccount sav = new Saving Account
(customerName,
customerName,
accountNum,
balance)

while (!exit)
{
    System.out.println(" --- MENU ---");
    System.out.println("1. Deposit
    2. Withdraw
    3. Compute Interest
    4. Display details
    5. Exit");
    int choice = s.nextInt();
    switch (choice)
    {
        case 1:
            System.out.println("Enter deposit
            amount");
            double amount = s.nextDouble();
            sav.deposit(amount);
            break;
        case 2:
            System.out.println("Enter withdrawal
            amount");
            double w_amount = s.nextDouble();
            sav.withdraw(w_amount);
            break;
        case 3:
            sav.calcInterest();
            break;
        case 4:
            sav.getBalance();
            break;
        case 5:
            exit = true;
    }
}

```

else if (opt == 2)

```
    cur = new CurrentAccount(  
        customerName, accountNo,  
        balance);
```

while (!exit)

```
        System.out.println("---- MENU ----");  
        System.out.println("1. Deposit 2. Withdraw  
        3. Compute Interest 4.  
        5. Display Details 6.  
        S. Exit");
```

```
        int choice = keyboard.nextInt();
```

```
        switch (choice)
```

```
        {
```

```
            case 1:  
                System.out.println("Enter deposit amount");  
                double amount = keyboard.nextDouble();
```

```
                cur.deposit(amount);
```

```
                break;
```

```
            case 2:
```

```
                System.out.println("Enter withdraw  
                amount");
```

```
                double w_amount = keyboard.nextDouble();
```

```
                cur.withdraw(w_amount);
```

```
                break;
```

```
            case 3:
```

```
                cur.getDetails();
```

```
                break;
```

```
            case 4:
```

```
                System.out.println("Cannot comput  
                interest");
```

```
                break;
```

```
            case 5:
```

```
                System.out.println("Current Account  
                does not support interest computation");
```

```
                break;
```

```
            case 6:
```

```
                System.out.println("Current Account  
                does not support withdrawal");
```

```
                break;
```

```
            case 7:
```

```
                System.out.println("Current Account  
                does not support deposit");
```

```
                break;
```

```
            case 8:
```

```
                System.out.println("Current Account  
                does not support display details");
```

```
                break;
```

```
            case 9:
```

```
                System.out.println("Current Account  
                does not support exit");
```

```
                break;
```

```
        }
```

classmate
Date _____
Page _____

case 4:

cur.getDetails();

break; // exit loop

case 5:

System.out.println("Current Account
does not support interest computation");

exit = true; // exit loop

case 6:

System.out.println("Current Account
does not support withdrawal");

exit = true; // exit loop

case 7:

System.out.println("Current Account
does not support deposit");

exit = true; // exit loop

case 8:

System.out.println("Current Account
does not support display details");

exit = true; // exit loop

case 9:

System.out.println("Current Account
does not support exit");

exit = true; // exit loop

O/P

=> Enter Customer name: Nitish Mehta
Likith

Enter Account number:

10000

Select Account type

1. Savings

2. Current

2.

---- MENU ----

1. Deposit

2. Withdraw

3. Compute Interest

4. Display Details

5. EXIT

1

Enter Deposit Amount

3000

classmate
Date _____
Page _____

classmate
Date _____
Page _____

----- MENU -----

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

2.

Enter the withdrawal amount

200

----- MENU -----

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

8.

Cannot compute Interest for current Account

----- MENU -----

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

4.

Current Balance 10300.0

----- MENU -----

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

5.

classmate
Date _____
Page _____

Enter the customer name:

Likhith

Enter the Account number:

84

Enter the account balance:

1

Select the account type

1. Savings
2. Current

1.

----- MENU -----

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

1.

Enter the amount

1000

----- MENU -----

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

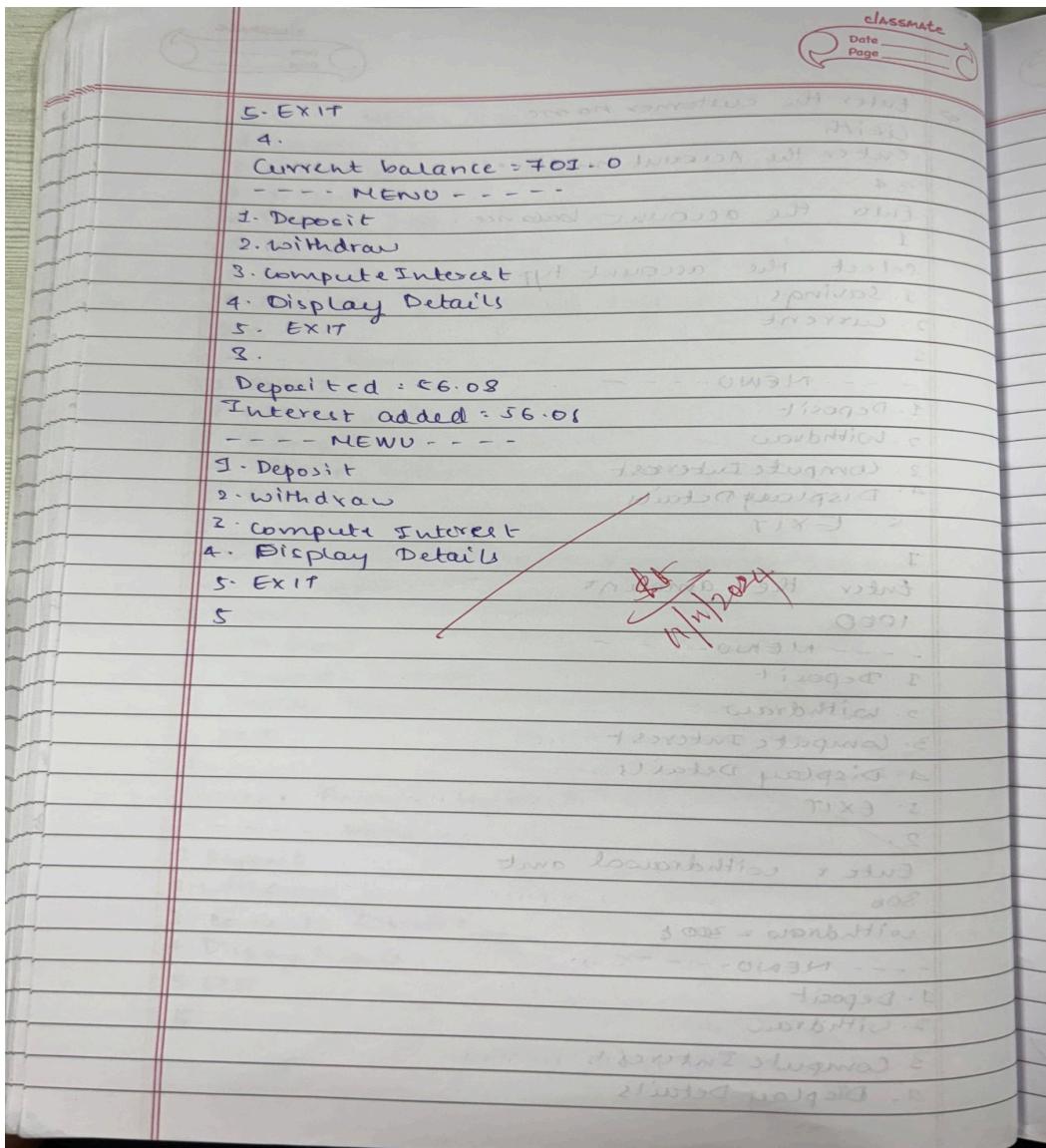
2.

Enter withdrawal amt

300

----- MENU -----

1. Deposit
2. Withdraw
3. Compute Interest
4. Display Details



CODE-

```
import java.util.*;  
class Account  
{  
    String customerName;  
    String accountNumber;  
    double balance;  
    public Account(String customerName, String accountNumber, double initialBalance)  
    {  
        this.customerName = customerName;  
        this.accountNumber = accountNumber;
```

```

this.balance = initialBalance;
}
public void deposit(double amount)
{
    balance += amount;
    System.out.println("Deposited: " + amount);
}
public void displayBalance()
{
    System.out.println("Current Balance: " + balance);
}
public void withdraw(double amount)
{
    if (amount <= balance)
    {
        balance -= amount;
        System.out.println("Withdrawn: " + amount);
    }
    else
    {
        System.out.println("Insufficient balance for withdrawal.");
    }
}
public double getBalance()
{
    return balance;
}
}
class SavingsAccount extends Account {
    double interestRate=0.08;
    public SavingsAccount(String customerName, String accountNumber, double initialBalance)
    {
        super(customerName, accountNumber, initialBalance);
    }
    public void addInterest() {
        double interest = balance * interestRate;
        deposit(interest);
        System.out.println("Interest added: " + interest);
    }
}
class CurrentAccount extends Account {
    double MIN_BALANCE = 500.0;
    double SERVICE_CHARGE = 50.0;
    public CurrentAccount(String customerName, String accountNumber, double initialBalance) {
        super(customerName, accountNumber, initialBalance);
    }
}

```

```

public void withdraw(double amount)
{
    if (amount > balance)
    {
        System.out.println("Insufficient balance for withdrawal.");
    }
    if (balance - amount < MIN_BALANCE)
    {
        System.out.println("Minimum balance violation. Service charge applied.");
        balance -= SERVICE_CHARGE; // Apply service charge
        System.out.println("Service charge: " + SERVICE_CHARGE);
    }
}
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 :");
        String accNum=s.next();
        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)
        {
            SavingsAccount sav=new SavingsAccount(name,accNum, balance);
            while(!exit)
            {
                System.out.println("-----MENU-----");
                System.out.println("1.Deposit \n2.Withdraw \n3.Compute Interest
\n4.Display Details \n5.Exit");
                System.out.println("Likith.Tarakaram,1BM23CS171");
                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.addInterest();
        break;
    case 4:
        sav.displayBalance();
        break;
    case 5:
        exit=true;
    }
}
else if (opt==2)
{
    CurrentAccount cur=new CurrentAccount(name,accNum, balance);
    while(!exit)
    {
        System.out.println("-----MENU-----");
        System.out.println("1.Deposit \n2.Withdraw \n3.Compute
Interest \n4.Display Details \n5.Exit");
        System.out.println("Likith.Tarakaram,1BM23CS171");
        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.displayBalance();
                break;
            case 5:
                exit=true;
        }
    }
}

```

```

        }
    }
}
}
```

Enter the customer name :

likith

Enter the Account Number :

1

Enter the Account Balance :

10000

Select the Account type

1.Savings

2.Current

2

-----MENU-----

1.Deposit

2.Withdraw

3.Compute Interest

4.Display Details

5.Exit

Likith.Tarakaram,1BM23CS171

1

Enter the deposit amount

300

Deposited: 300.0

-----MENU-----

-----MENU-----

1.Deposit

2.Withdraw

3.Compute Interest

4.Display Details

5.Exit

Likith.Tarakaram,1BM23CS171

2

Enter the withdrawal amount

200

-----MENU-----

1.Deposit

2.Withdraw

3.Compute Interest

4.Display Details

5.Exit

Likith.Tarakaram,1BM23CS171

3

Cannot Compute Interest for Current Account

-----MENU-----

1.Deposit

2.Withdraw

3.Compute Interest

4.Display Details

5.Exit

Likith.Tarakaram,1BM23CS171

4

Current Balance: 10300.0

-----MENU-----

1.Deposit

2.Withdraw

3.Compute Interest

4.Display Details

5.Exit

Likith.Tarakaram,1BM23CS171

5

Enter the customer name :

likith

Enter the Account Number :

34

Enter the Account Balance :

1

Select the Account type

1.Savings

2.Current

1

-----MENU-----

1.Deposit

2.Withdraw

3.Compute Interest

4.Display Details

5.Exit

Likith.Tarakaram,1BM23CS171

1

Enter the deposit amount

1000

Deposited: 1000.0

-----MENU-----

```

-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
Likith.Tarakaram,1BM23CS171
2
Enter the withdrawal amount
300
Withdrawn: 300.0
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
Likith.Tarakaram,1BM23CS171
4
Current Balance: 701.0
-----MENU-----
1.Deposit
2.Withdraw
3.Compute Interest
4.Display Details
5.Exit
Likith.Tarakaram,1BM23CS171
3
Deposited: 56.08

```

Interest added: 56.08

-----MENU-----

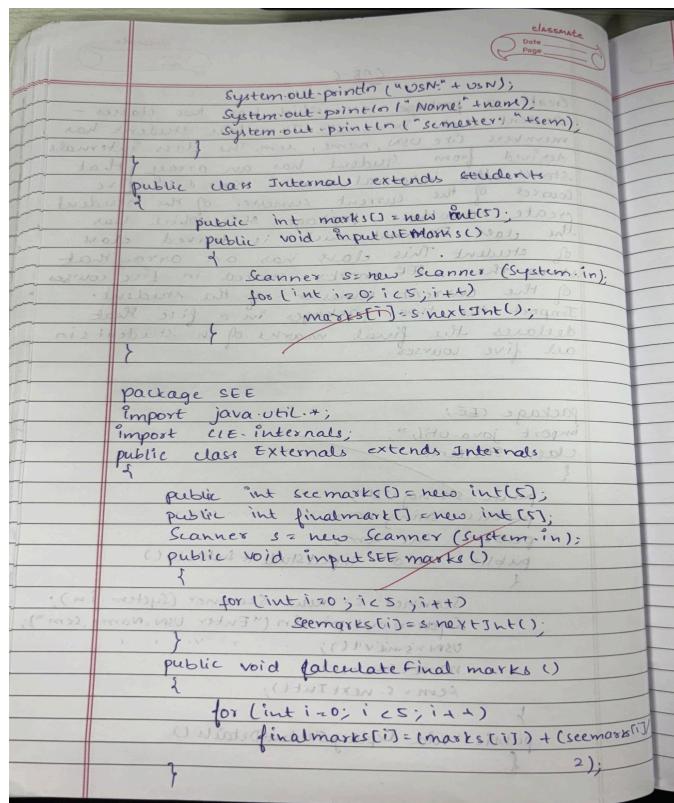
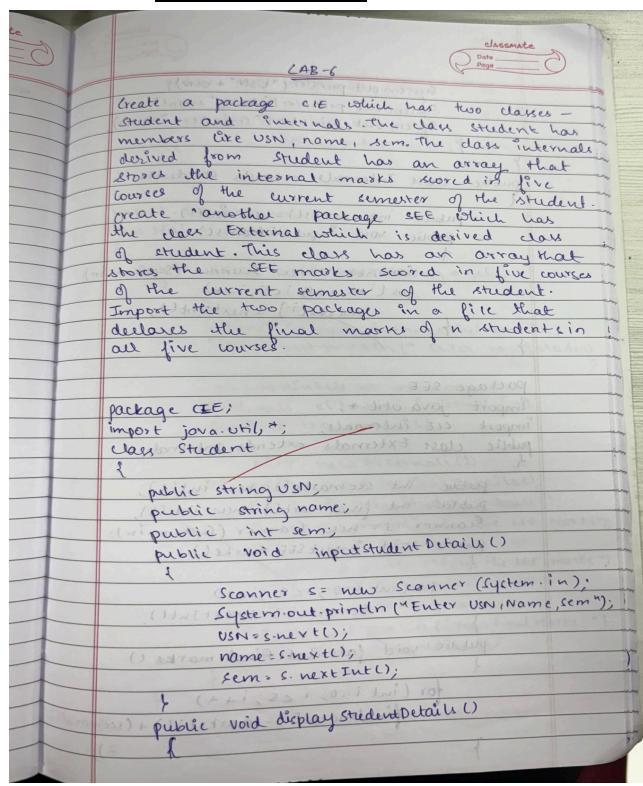
- 1.Deposit
- 2.Withdraw
- 3.Compute Interest
- 4.Display Details
- 5.Exit

Likith.Tarakaram,1BM23CS171

Program 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.

ALGORITHM-



```

classmate
Date _____
Page _____
public void displayFinalMark()
{
    for (int i=0; i<5; i++)
        System.out.println("Finalmarks(" + i + ")");
}

import SEE.Externals
class MainMarks
{
    public static void main(String args[])
    {
        int n;
        Scanner s = new Scanner(System.in)
        System.out.println("Enter no. of students")
        n = s.nextInt();
        Externals e[] = new Externals[n];
        for (int i=0; i<n; i++)
        {
            e[i] = new Externals();
            e[i].inputStudentDetails();
            e[i].displayStudentDetails();
            System.out.println("Enter the CIE Marks");
            e[i].inputCIEmarks();
            System.out.println("Enter the SEE marks");
            e[i].inputSEEmarks();
            e[i].calculateFinalMarks();
            System.out.println("Your Final Marks:");
            e[i].displayFinalMarks();
        }
    }
}

```

classmate
Date _____
Page _____

OOP

Enter the number of students

2

(L1) Enter name, USN, semester

Lipith

USN : IBN123CS171

Semester : 3

Enter the CIE marks

48

49

50

50

48

Enter the SEE marks

99

100

100

98

96

Your Final marks are

97

99

100

99

93

Enter Name, USN, semester

Shailash

IBN123CS172

3

Name : Shailash

USN : IBN123CS172

CODE-

```
package CIE;  
import java.util.*;  
class Student  
{  
    public String usn;  
    public String name;
```

```

public int sem;
public void inputStudentDetails()
{
    Scanner s=new Scanner(System.in);
    System.out.println("Enter Name,USN,Semester");
    usn=s.next();
    name=s.next();
    sem=s.nextInt();
}
public void displayStudentDetails()
{
    System.out.println("USN:"+usn);
    System.out.println("Name"+name);
    System.out.println("Semester:"+sem);
}
public class Internals extends Student
{
    public int marks[]={};
    public void inputCIEMarks()
    {
        Scanner s=new Scanner(System.in);
        for(int i=0;i<5;i++)
            marks[i]=s.nextInt();
    }
}
package SEE;
import java.util.*;
import CIE.Internals;
public class Externals extends Internals
{
    public int seemarks[]={};
    public int finalmarks[]={};
    Scanner s=new Scanner(System.in);
    public void inputSEEmarks()
    {
        for(int i=0;i<5;i++)
            seemarks[i]=s.nextInt();
    }
    public void calculateFinalMarks()
    {
        for(int i=0;i<5;i++)
            finalmarks[i]=(marks[i])+(seemarks[i]/2);
    }
    public void displayFinalMarks()
    {
        for(int i=0;i<5;i++)

```

```

        System.out.println(finalmarks[i]);
    }
}
import SEE.Externals;
import java.util.*;
class MainMarks
{
    public static void main(String args[])
    {
        int n;
        Scanner s=new Scanner(System.in);
        System.out.println("enter number of students:");
        n=s.nextInt();
        Externals e[]={new Externals[n];
        for(int i=0;i<n;i++)
        {
            e[i]=new Externals();
            e[i].inputStudentDetails();
            e[i].displayStudentDetails();
            System.out.println("Enter the CIE marks");
            e[i].inputCIEMarks();
            System.out.println("Enter the SEE marks");
            e[i].inputSEEmarks();
            e[i].calculateFinalMarks();
            System.out.println("Your Final marks are");
            e[i].displayFinalMarks();
        }
    }
}

```

OUTPUT-

```

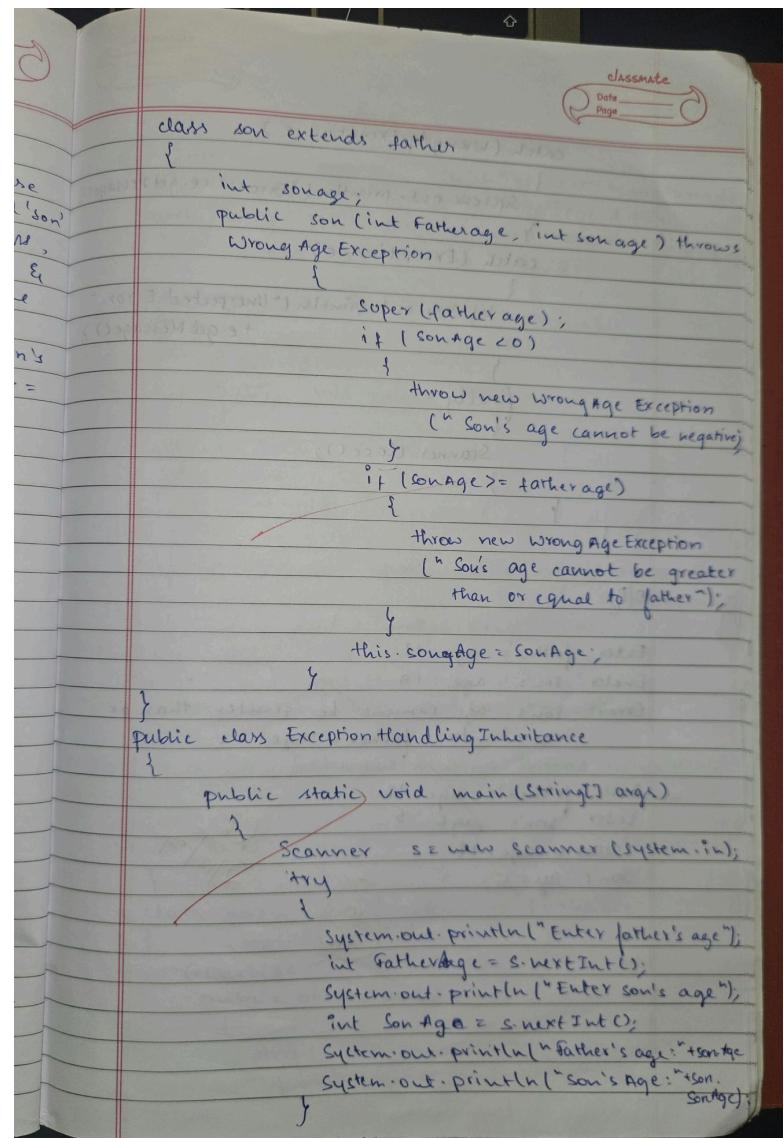
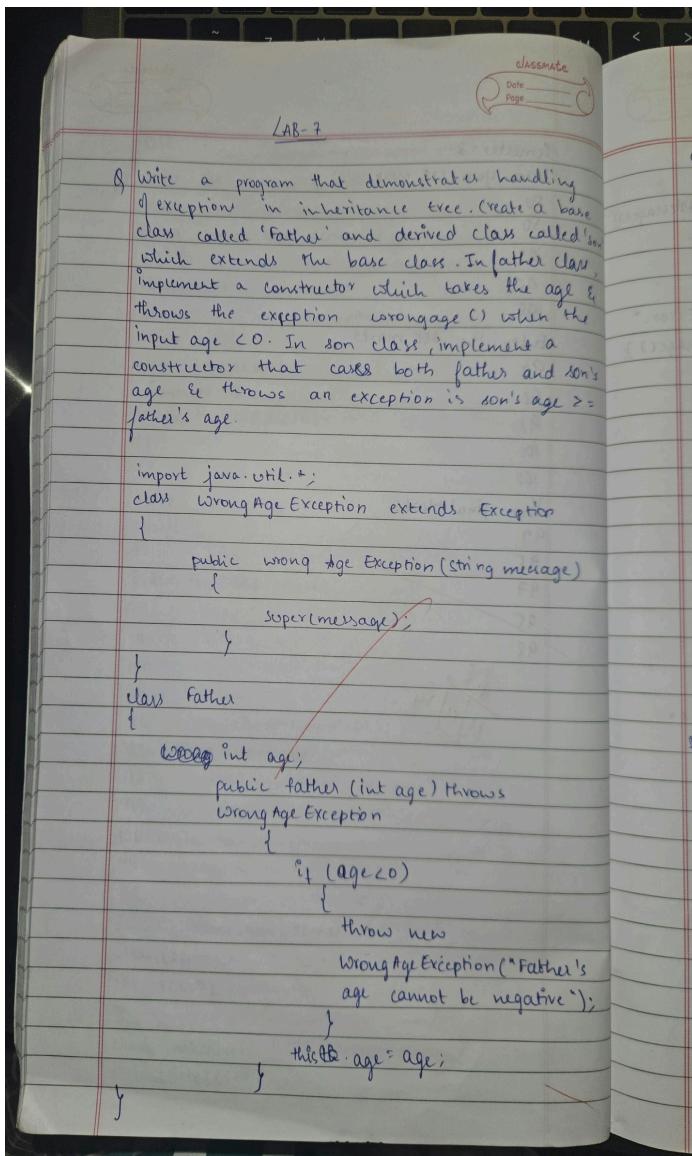
D:\IBM23CS171\JAVA>javac CIEInternals.java
D:\IBM23CS171\JAVA>javac SEEExternals.java
D:\IBM23CS171\JAVA>javac MainMarks.java
D:\IBM23CS171\JAVA>java MainMarks
Enter number of students:
2
Enter Name,USN,Semester
lkhith
1BM23CS171
1
USN:lkhith
Name1BM23CS171
Semester:1
Enter the CIE marks
48
49
58
59
48
Enter the SEE marks
99
100
100
98
90
Your Final marks are
97
99
100
99
93
Enter Name,USN,Semester
Shailesh
1BM23CS172
1
USN:Shailesh
Name1BM23CS172
Semester:3
Enter the CIE marks
50
50
48
48
48
Enter the SEE marks
99
98
99
100
100
Your Final marks are
99
95
97
95
98
D:\IBM23CS171\JAVA>

```

Program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age < 0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is >= father's age.

ALGORITHM-



classmate
Date _____
Page _____

```

        catch (WrongAgeException e)
        {
            System.out.println("Error:" + e.getLocalizedMessage());
        }
        catch (Exception e)
        {
            System.out.println("Unexpected Error:" + e.getMessage());
        }
    }
    finally
    {
        Scanner.close();
    }
}

```

Output:

```

Enter father's age: 18
Enter son's age: 19
Error: son's age cannot be greater than or
equal to father's age.

```

Enter father's age: 18
 Enter son's age: 5
 Father's age: 18
 Son's age: 5

~~10%~~

CODE-

```

import java.util.Scanner;

class PrintInfo
{
    static void print()
    {
        System.out.println("Likith.Tarakaram");
        System.out.println("1BM23CS171");
    }
}

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

class Father {
    int age;
}

```

```

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

class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be negative.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to Father's age.");
        }
        this.sonAge = sonAge;
    }
}

public class ExceptionHandlingInheritance
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        PrintInfo.print();
        try
        {
            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();
            System.out.print("Enter Son's age: ");
            int sonAge = scanner.nextInt();
            Son son = new Son(fatherAge, sonAge);
            System.out.println("Father's age: " + son.age);
            System.out.println("Son's age: " + son.sonAge);
        }
        catch (WrongAgeException e)
        {
            System.out.println("Error: " + e.getMessage());
        }
    }
}

```

```
        catch (Exception e)
        {
            System.out.println("Unexpected error: " + e.getMessage());
        }
    finally
    {
        scanner.close();
    }
}
```

OUTPUT-

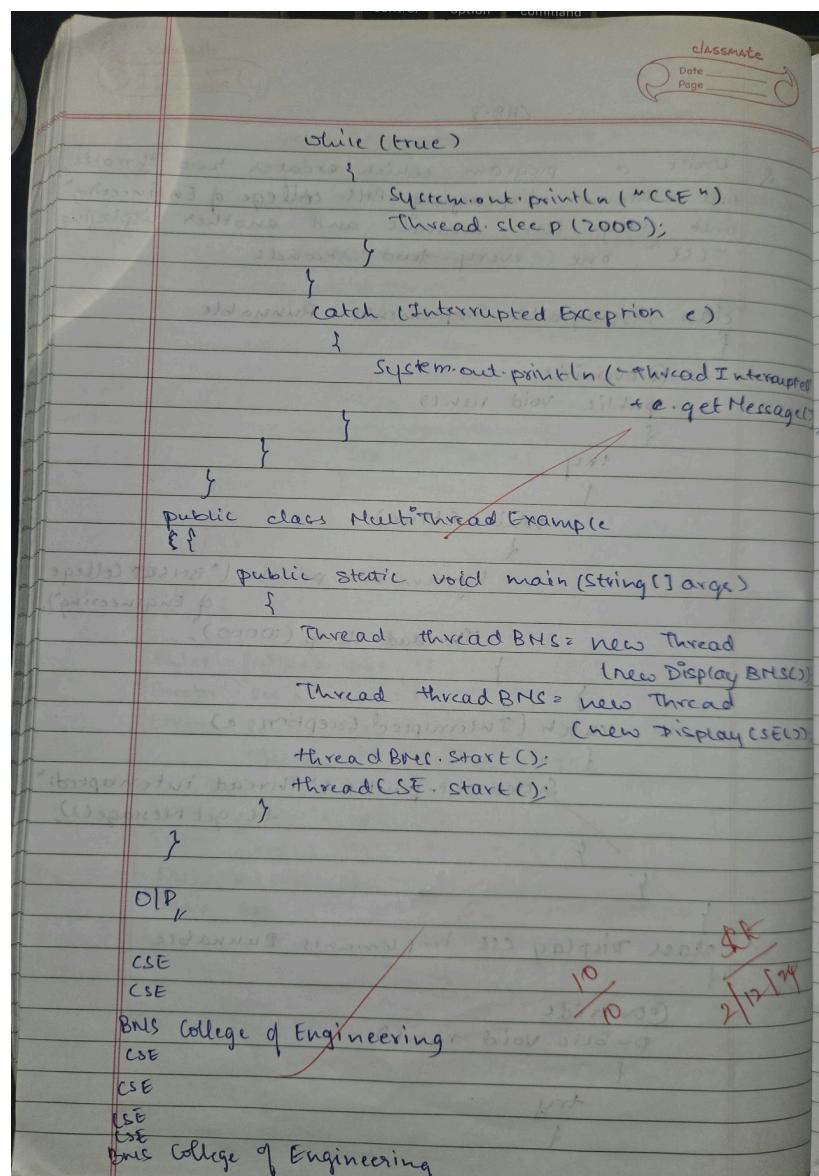
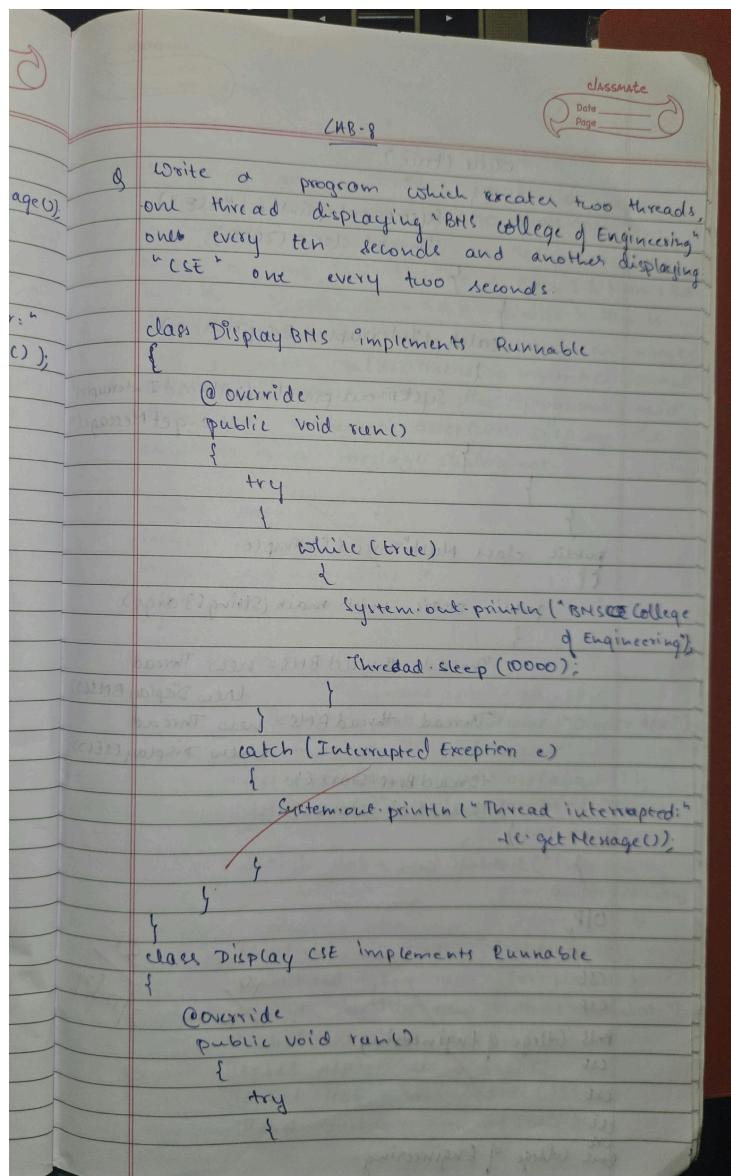
```
Name: Likith T
USN: 1BM23CS171
Enter Father's age: 23
Enter Son's age: 34
Error: Son's age cannot be greater than or equal to Father's age.
```

```
Name: Likith T
USN: 1BM23CS171
Enter Father's age: 45
Enter Son's age: 23
Father's age: 45
Son's age: 23
```

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-



CODE-

```

class PrintInfo {
    static void print()
    {
    }
}

```

```

        {
            System.out.println("Likith.Tarakaram");
            System.out.println("1BM23CS171");
        }
    }

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

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

    public static void main(String[] args)
}

```

```
{  
    PrintInfo.print();  
    Thread bmsThread = new BMSDisplayThread();  
    Thread cseThread = new CSEDisplayThread();  
    bmsThread.start();  
    cseThread.start();  
}  
}
```

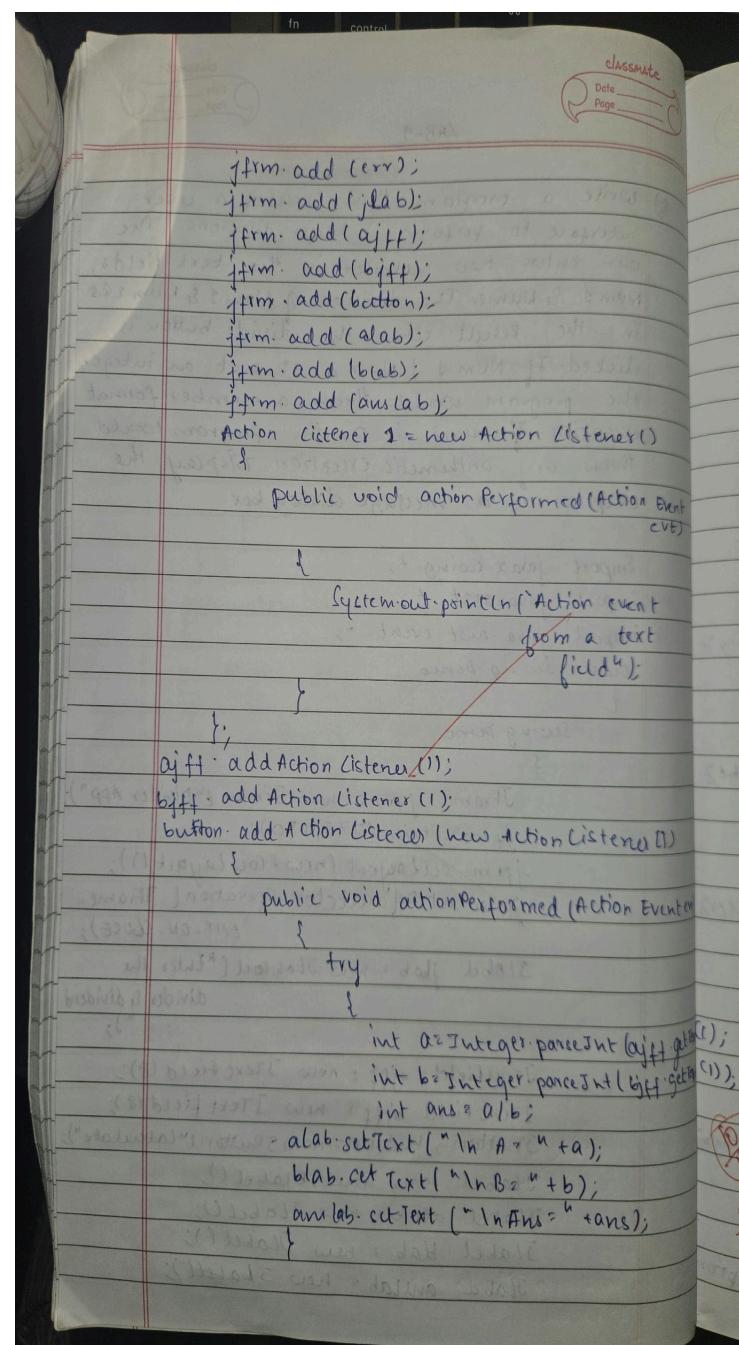
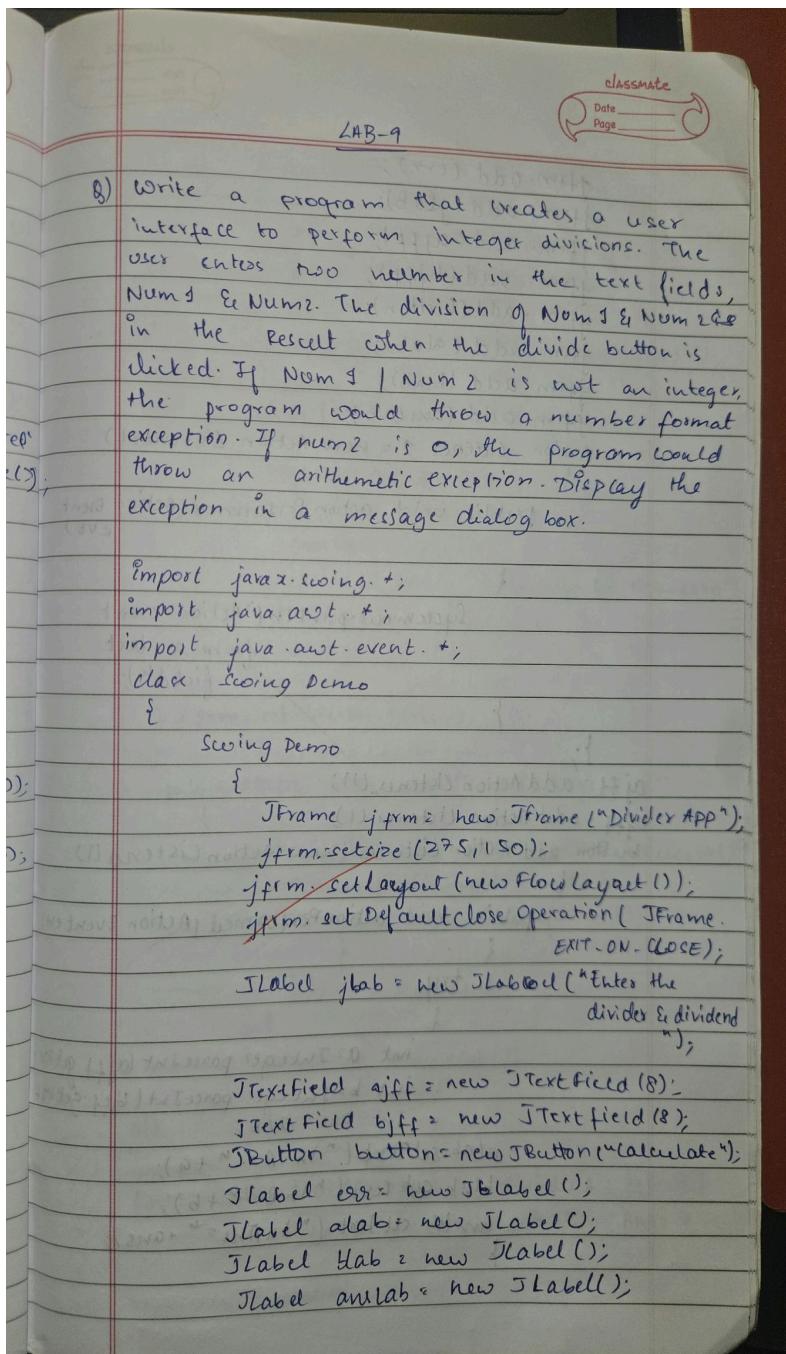
OUTPUT-

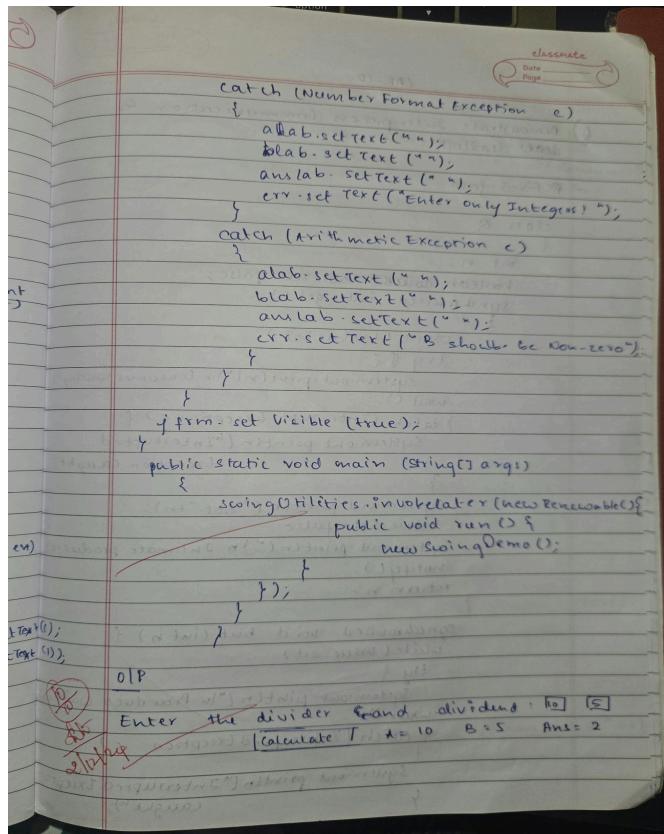
```
Likith.Tarakaram  
1BM23CS171  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
RMS College of Engineering
```

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-





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 divider and divident:");
        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);
ActionListener l = new ActionListener() {
public void actionPerformed(ActionEvent evt) { System.out.println("Action event from a text field");
} };
ajtf.addActionListener(l);
bjtf.addActionListener(l);
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;
alab.setText("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = "+ ans);
}
catch(NumberFormatException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("Enter Only Integers!"); }
catch(ArithmeticException 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();
}});
}
}
}

```

OUTPUT-

Enter the divider and dividend:	<input type="text" value="10"/>	<input type="text" value="5"/>	<input type="button" value="Calculate"/>	A = 10 B = 5 Ans = 2
---------------------------------	---------------------------------	--------------------------------	--	----------------------

PROGRAM 10:

Deadlock

ALGORITHM-

classmate
Date _____
Page _____

LAB-10

Q) Demonstrate Inter process communication by deal deadlock

→ Pcfixed.java

```

class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet)
            try {
                System.out.println("In consumer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
            }
        System.out.println("Got:" + n);
        valueSet = true;
    }
    synchronized void put(int n) {
        while (valueSet)
            try {
                System.out.println("In Producer waiting");
                wait();
            } catch (InterruptedException e) {
                System.out.println("Interrupted Exception caught");
            }
        System.out.println("In Intimate producer");
        notify();
        return n;
    }
}

```

classmate
Date _____
Page _____

```

this.n = n;
valueSet = true;
System.out.println("Put:" + n);
System.out.println("In Intimate consumer");
notify();
}

class Producer implements Runnable {
    Q q;
    Producer(Q q) {
        this.q = q;
    }
    public void run() {
        int i = 0;
        while (i < 15) {
            q.put(i++);
        }
    }
}

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

```

classmate
Date _____
Page _____

```

class PCfixed {
    public static void Main (String [] args) {
        System.out.println ("Name : Likith,
                            USN : IB2123CS171");
        Q q = new Q ();
        new Producer (q);
        new Consumer (q);
        System.out.println ("Press control-C to stop");
    }
}

class 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 ();
        }
    }
}

```

classmate
Date _____
Page _____

```

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 ();
    }
}

class Deadlock implements Runnable {
    A a = new A ();
    B b = new B ();
    Deadlock () {
        Thread.currentThread().setName ("Main
                                    thread");
        Thread t = new Thread (this, "Racing Thread");
        t.start ();
        a.foo (b);
        System.out.println ("Back in mainthread");
    }

    public void run () {
        b.bar (a);
    }
}

```

classmate
Date _____
Page _____

```

System.out.println ("Back in other thread");
public static void main (String args[]) {
    System.out.println ("Name : Likith, USN : IB2123CS171");
    new Deadlock ();
}

class Deadlock {
    Name: Likith (USN: IBH23CS171)
    Main thread entered A.foo
    Racing thread entered B.bar
    Main thread trying to call B.last()
    Racing thread trying to call A.last()
    Inside A.last
    Inside A.last
    Back in other thread
    Back in other thread
}

class PCfixed {
    Name: Likith , USN - IBH23CS171
    Press control C to stop
    Put: 0
    Intimate Consumer
    Producer waiting
    Got: 0
    Intimate producer
    Put: 1
    Intimate consumer
    Producer waiting
    Got: 0
    Intimate producer
    Consumed: 1
}

```

classmate
Date _____
Page _____

```

Put: 2
Intimate consumer
Producer waiting
Consumed: 1
Got: 2
Intimate Producer
Consumed: 2
Put: 3
Intimate Consumer
Producer waiting
Got: 3
Intimate Producer
Consumed: 3
Put: 4
Intimate Consumer
Producer waiting
Got: 4
Intimate Producer
Consumed: 4
Put: 5
Intimate consumer consumer
Producer waiting
Got: 5
Intimate Producer
Consumed: 5
Put: 6
Intimate consumer
Producer waiting
Got: 6
Intimate Producer
Consumed: 6
Put: 7
Intimate consumer
Producer waiting

```

Got: 7
 Intimate Producer
 Consumed: 7
 Put: 8
 Intimate Consumer
 Producer waiting
 Got: 8
 Intimate Producer
 Consumed: 8
 Put: 9
 Intimate Consumer
 Producer waiting
 Got: 9
 Intimate Producer
 Consumed: 9
 Put: 10
 Intimate Consumer
 Producer waiting
 Got: 10
 Intimate Producer
 Consumed: 10
 Put: 11
 Intimate Consumer
 Producer waiting
 Got: 11
 Intimate Producer
 Consumed: 11
 Put: 12
 Intimate Consumer
 Producer waiting
 Got: 12
 Intimate Producer
 Consumed: 12
 Put: 13

Intimate Consumer
 Producer waiting
 Got: 13
 Intimate producer
 Consumed: 13
 Put: 14
 Intimate Consumer
 Producer waiting
 Got: 14
 Intimate producer
 Consumed: 14

~~8~~
~~12~~
~~10~~
~~9~~
~~11~~

CODE-

PROGRAM 10A-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();
    }
}

```

```

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); // get lock on a in this thread.

    System.out.println("Back in mainthread");

}

public void run() {

    b.bar(a); // get lock on b in otherthread.

    System.out.println("Back in otherthread");

}

public static void main(String args[]) {
    System.out.println("Name: Likith T, USN: 1BM23CS171");
    new Deadlock();

}
}

```

PROGRAM 10B-PCFIXED:

```
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("\nIntimate 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("\nIntimate 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[]) {
        System.out.println("Name: Likith T, USN: 1BM23CS171");
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}

```

OUTPUT-

```
Name: Likith T, USN: 1BM23CS171
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
Inside A.last
Back in mainthread
Inside A.last
Back in otherthread
E:\1BM23CS146>java PCFixed
Name: Likith T, USN: 1BM23CS171
Press Control-C to stop.
Put: 0

Intimate Consumer

Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer

consumed:0

Producer waiting

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

Producer waiting

Got: 4

Intimate Producer

consumed:4
Put: 5

Intimate Consumer

Producer waiting

Got: 5

Intimate Producer

consumed:5
Put: 6

Intimate Consumer
```

```
consumed:6
Put: 7

Intimate Consumer

Producer waiting

Got: 7

Intimate Producer

consumed:7
Put: 8

Intimate Consumer

Producer waiting

Got: 8

Intimate Producer

Put: 9

Intimate Consumer

Producer waiting

consumed:8
Got: 9

Intimate Producer

consumed:9
Put: 10

Intimate Consumer

Producer waiting

Got: 10

Intimate Producer
```

```
Got: 11

Intimate Producer

consumed:11
Put: 12

Intimate Consumer

Producer waiting

Got: 12

Intimate Producer

consumed:12
Put: 13

Intimate Consumer

Producer waiting

Got: 13

Intimate Producer

consumed:13
Put: 14

Intimate Consumer

Got: 14

Intimate Producer

consumed:14
```