

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT on

Object Oriented Java Programming (23CS3PCOOJ)

Submitted by

Kashvi Agarwal (**1BM23CS141**)

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 **Kashvi Agarwal (1BM23CS141)**, 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 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	1-3
2	07/10/24	Student SGPA	4-7
3	14/10/24	Book Details	8-10
4	21/10/24	Area of the Shape	11-14
5	28/10/24	Bank	15-21
6	11/11/24	Package	22-27
7	28/11/24	Exception Handling Inheritance	28-30
8	28/11/24	Threads	31-32
9	28/11/24	Swing Demo	33-36
10	28/11/24	A. Deadlock B. PCFixed	37-45

Program 1

Quadratic Equation

Algorithm:

The image contains three panels of handwritten notes. The left panel shows the Java code for a quadratic equation solver. The middle panel shows the execution output of the program. The right panel shows the calculated roots.

Java Code (Left Panel):

```

Date 30/3/2014
Page - 1 / 1
Labs PROGRAM - 1
Description: A Java program that takes all real
inputs to the quadratic eqn and then
prints out the roots and uses the
quadratic formula
If the discriminant is less than
zero, it displays a message saying that
there are no real roots
import java.util.Scanner;
class Coeff {
    double a;
    double b;
    double c;
}
class PrintInfo {
    static void print() {
        System.out.println("Name: Kashvi Agarwal");
        System.out.println("VAN 1BM23C141");
    }
}
public class QuadraticEquation {
    public static void main (String [] args) {
        PrintInfo.print();
        Scanner scanner = new Scanner (System.in);
        Coeff coeff = new Coeff();
        System.out.println ("Enter coefficient a");
        coeff.a = scanner.nextDouble();
        System.out.println ("Enter coefficient b");
        coeff.b = scanner.nextDouble();
        while (coeff.a == 0) {
            System.out.println ("Not a quadratic
            equation. Please enter
            non-zero value for a");
            coeff.a = scanner.nextDouble();
        }
        coeff.c = scanner.nextDouble();
    }
}

```

Execution Output (Middle Panel):

```

Date 30/3/2014
Page - 1 / 1
System.out.print("Enter coefficient of 'a' ");
coeff.a = scanner.nextDouble();
System.out.print("Enter coefficient 'b' ");
coeff.b = scanner.nextDouble();
coeff.c = scanner.nextDouble();
double d = coeff.b * coeff.b - 4 * coeff.a * coeff.c;
if (d == 0) {
    double r1 = -coeff.b / (2 * coeff.a);
    System.out.println ("Roots are real and equal");
    System.out.println ("Root 1 = " + r1 + " and Root 2 = " + r1);
} else if (d > 0) {
    double r1 = (-coeff.b + Math.sqrt(d)) / (2 * coeff.a);
    double r2 = (-coeff.b - Math.sqrt(d)) / (2 * coeff.a);
    System.out.println ("Roots are real and unequal");
    System.out.println ("Root 1 = " + r1);
    System.out.println ("Root 2 = " + r2);
} else {
    double realPart = -coeff.b / (2 * coeff.a);
    double imaginaryPart = Math.sqrt (-d) / (2 * coeff.a);
    System.out.println ("Roots are imaginary");
    System.out.println ("Root 1 = " + realPart + " + " + imaginaryPart + "i");
    System.out.println ("Root 2 = " + realPart + " - " + imaginaryPart + "i");
}
scanner.close();

```

Calculated Roots (Right Panel):

Name: Kashvi Agarwal
VAN 1BM23C141
Enter coefficient of 'a', 'b', 'c'
Enter coefficient 'a': 1
Enter coefficient 'b': 3
Enter coefficient 'c': 2
Roots are real and unequal.
Root 1 = -1.0
Root 2 = -2.0

Enter coefficient 'a' : 1
Enter coefficient 'b' : 5
Enter coefficient 'c' : 8
Roots are imaginary
Root 1 : -2.5+1.322
Root 2 : -2.5-1.322

Enter coefficient 'a' : 1
Enter coefficient 'b' : -8
Enter coefficient 'c' : 16
Roots are real and equal
Root 1: 4.0
Root 2: 4.0

80.00%

Code:

```

import java.util.Scanner;

class Coeff {
    double a;
    double b;
    double c;
}

class PrintInfo {
    static void print() {
        System.out.println("Name: Kashvi Agarwal");
    }
}

```

```

        System.out.println("USN: 1BM23CS141");
    }
}

public class QuadraticEquation {
    public static void main(String[] args) {
        PrintInfo.print();

        Scanner scanner = new Scanner(System.in);
        Coeff coeff = new Coeff();

        System.out.println("Enter the coefficients of a, b, c:");
        System.out.print("Enter coefficient a: ");
        coeff.a = scanner.nextDouble();
        while (coeff.a == 0) {
            System.out.println("Not a quadratic equation. Please enter a non-zero value for a:");
            coeff.a = scanner.nextDouble();
        }

        System.out.print("Enter coefficient b: ");
        coeff.b = scanner.nextDouble();
        System.out.print("Enter coefficient c: ");
        coeff.c = scanner.nextDouble();

        double d = coeff.b * coeff.b - 4 * coeff.a * coeff.c;

        if (d == 0) {
            double r1 = -coeff.b / (2 * coeff.a);
            System.out.println("Roots are real and equal.");
            System.out.println("Root 1 and Root 2: " + r1);
        } else if (d > 0) {
            double r1 = (-coeff.b + Math.sqrt(d)) / (2 * coeff.a);
            double r2 = (-coeff.b - Math.sqrt(d)) / (2 * coeff.a);
            System.out.println("Roots are real and unique.");
            System.out.println("Root 1: " + r1);
            System.out.println("Root 2: " + r2);
        } else {
            double realPart = -coeff.b / (2 * coeff.a);
            double imaginaryPart = Math.sqrt(-d) / (2 * coeff.a);
            System.out.println("Roots are imaginary.");
            System.out.println("Root 1: " + realPart + " + " + imaginaryPart + "i");
            System.out.println("Root 2: " + realPart + " - " + imaginaryPart + "i");
        }
    }
}

```

```
        }  
  
        scanner.close();  
    }  
}
```

Output:

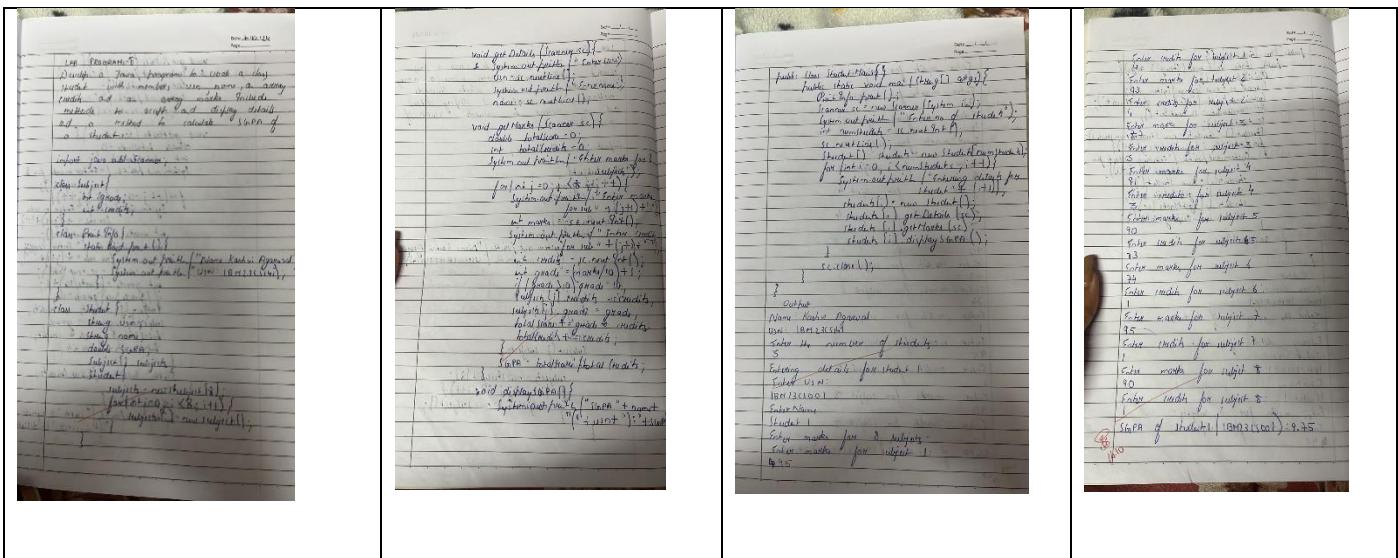
```
D:\1bm23cs141>java QuadraticEquation  
Name: Kashvi Agarwal  
USN: 1BM23CS141  
Enter the coefficients of a, b, c:  
Enter coefficient a: 1  
Enter coefficient b: 3  
Enter coefficient c: 2  
Roots are real and unique.  
Root 1: -1.0  
Root 2: -2.0
```

```
D:\1bm23cs141>java QuadraticEquation  
Name: Kashvi Agarwal  
USN: 1BM23CS141  
Enter the coefficients of a, b, c:  
Enter coefficient a: 1  
Enter coefficient b: 5  
Enter coefficient c: 6  
Roots are real and unique.  
Root 1: -2.0  
Root 2: -3.0
```

Program 2

Student SGPA

Algorithm:



Code:

```

import java.util.Scanner;

class Subject {
    int grade;
    int credits;
}

class PrintInfo {
    static void print() {
        System.out.println("Name: Kashvi Agarwal");
        System.out.println("USN: 1BM23CS141");
    }
}

class Student {
    String usn;
}
  
```

```

String name;
double SGPA;
Subject[] subjects;

Student() {
    subjects = new Subject[8];
    for (int i = 0; i < 8; i++) {
        subjects[i] = new Subject();
    }
}

void getDetails(Scanner sc) {
    System.out.println("Enter USN:");
    usn = sc.nextLine();
    System.out.println("Enter name:");
    name = sc.nextLine();
}

void getMarks(Scanner sc) {
    double totalScore = 0;
    int totalCredits = 0;

    System.out.println("Enter marks for 8 subjects:");
    for (int j = 0; j < 8; j++) {
        System.out.println("Enter marks for subject " + (j + 1) + ":");
        int marks = sc.nextInt();
        System.out.println("Enter the credits for subject " + (j + 1) + ":");
        int credits = sc.nextInt();

        int grade = (marks / 10) + 1;
        if (grade > 10) grade = 10;

        // Store the information in the subjects array
        subjects[j].credits = credits;
        subjects[j].grade = grade;
    }

    // Calculate score based on grade and credits
    totalScore += grade * credits;
    totalCredits += credits;
}

```

```

// Compute SGPA
SGPA = totalScore / totalCredits;
}

void displaySGPA() {
    System.out.println("SGPA of student " + name + " (" + usn + "): " + SGPA);
}
}

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

        Student[] students = new Student[numStudents];

        for (int i = 0; i < numStudents; i++) {
            System.out.println("Entering details for student " + (i + 1));
            students[i] = new Student();
            students[i].getDetails(sc);
            students[i].getMarks(sc);
            students[i].displaySGPA();
        }

        sc.close();
    }
}

```

Output:

The image shows two side-by-side Windows Command Prompt windows. Both windows have a title bar 'C:\Windows\system2\cmd.exe' and a close button 'X'. The left window shows the execution of the Java program and the input for student 1. The right window shows the execution of the Java program and the input for student 2.

```
C:\Windows\system2\cmd.exe > + x
Microsoft Windows [Version 10.0.22631.4249]
(c) Microsoft Corporation. All rights reserved.

D:\IBMR23CS141>javac StudentMains.java
D:\IBMR23CS141>java StudentMains
Name: Kashvi Agarwal
USN: IBMR23CS141
Enter the number of students:
Entering details for student 1
Enter USN:
IBMR23CS141
Enter name:
STUDENT1
Enter marks for 8 subjects:
Enter marks for subject 1:
99
Enter the credits for subject 1:
4
Enter marks for subject 2:
97
Enter the credits for subject 2:
4
Enter marks for subject 3:
98
Enter the credits for subject 3:
4
Enter marks for subject 4:
96
Enter the credits for subject 4:
4
Enter marks for subject 5:
98
Enter the credits for subject 5:
4
Enter marks for subject 6:
98
Enter the credits for subject 6:
4
Enter marks for subject 7:
98
Enter the credits for subject 7:
4
Enter marks for subject 8:
98
Enter the credits for subject 8:
4
SGPA of student STUDENT1 () : 9.15
Entering details for student 2
Enter USN:
D:\IBMR23CS141>
```

The taskbar at the bottom of the screen shows several pinned icons, including File Explorer, Edge, and File Manager, along with the system clock showing 10:10 08/10/2024.

This screenshot shows a single Windows Command Prompt window with the title 'C:\Windows\system2\cmd.exe' and a close button 'X'. It displays the execution of the Java program and the input for student 3.

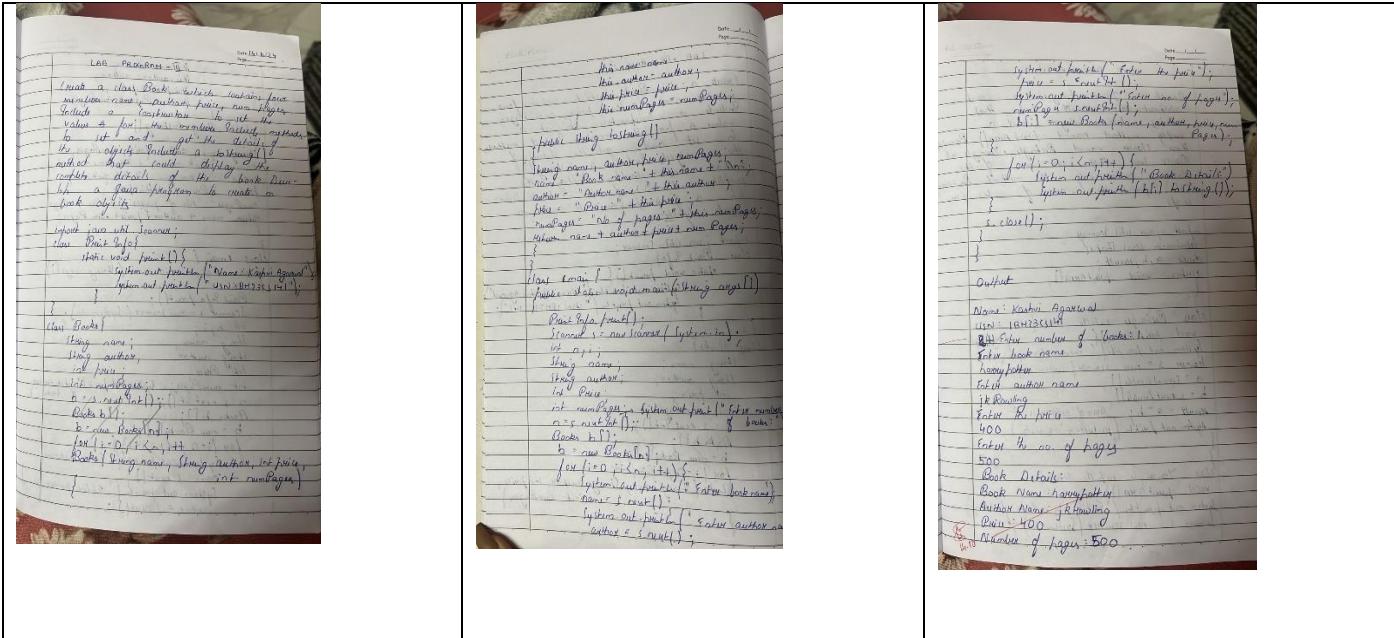
```
C:\Windows\system2\cmd.exe > + ~
SGPA of student STUDENT2 () : 9.15
Entering details for student 3
Enter USN:
STUDENT3
Enter the number of students:
Entering details for student 3
Enter name:
STUDENT3
Enter marks for 8 subjects:
Enter marks for subject 1:
98
Enter the credits for subject 1:
4
Enter marks for subject 2:
96
Enter the credits for subject 2:
4
Enter marks for subject 3:
94
Enter the credits for subject 3:
4
Enter marks for subject 4:
98
Enter the credits for subject 4:
4
Enter marks for subject 5:
74
Enter the credits for subject 5:
4
Enter marks for subject 6:
88
Enter the credits for subject 6:
4
Enter marks for subject 7:
93
Enter the credits for subject 7:
4
Enter marks for subject 8:
92
Enter the credits for subject 8:
4
SGPA of student STUDENT3 () : 9.3
D:\IBMR23CS141>
```

The taskbar at the bottom of the screen shows several pinned icons, including File Explorer, Edge, and File Manager, along with the system clock showing 10:49 08/10/2024.

Program 3

Book Details

Algorithm:



Code:

```

import java.util.Scanner;
class PrintInfo {
    static void print() {
        System.out.println("Name: Kashvi Agarwal");
        System.out.println("USN: 1BM23CS141");
    }
}
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 main{
        public static void main(String args[])
        {
            PrintInfo.print();
            Scanner s = new Scanner(System.in);

            int n;
            int i;
            String name;
            String author;
            int price;
            int numPages;
            n=s.nextInt();
            Books b[];
            b=new Books[n];
        }
    }
}

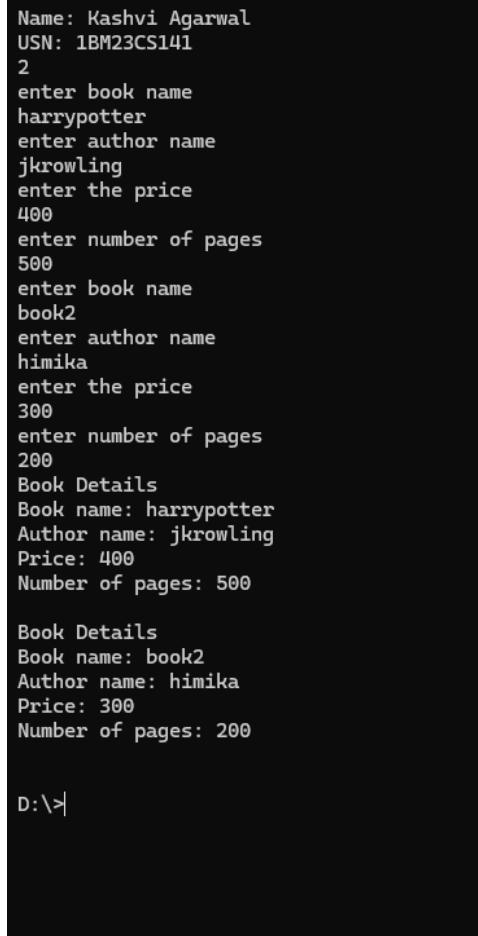
```

```

for(i=0;i<n;i++){
    System.out.println("enter book name");
    name = s.next();
    System.out.println("enter author name");
    author = s.next();
    System.out.println("enter the price");
    price=s.nextInt();
    System.out.println("enter number of pages");
    numPages = s.nextInt();
    b[i] = new Books(name,author,price,numPages);
}
for(i=0;i<n;i++){
    System.out.println("Book Details");
    System.out.println(b[i].toString());
}
s.close();
}
}

```

Output:



```

Name: Kashvi Agarwal
USN: 1BM23CS141
2
enter book name
harrypotter
enter author name
jkrowling
enter the price
400
enter number of pages
500
enter book name
book2
enter author name
himika
enter the price
300
enter number of pages
200
Book Details
Book name: harrypotter
Author name: jkrowling
Price: 400
Number of pages: 500

Book Details
Book name: book2
Author name: himika
Price: 300
Number of pages: 200

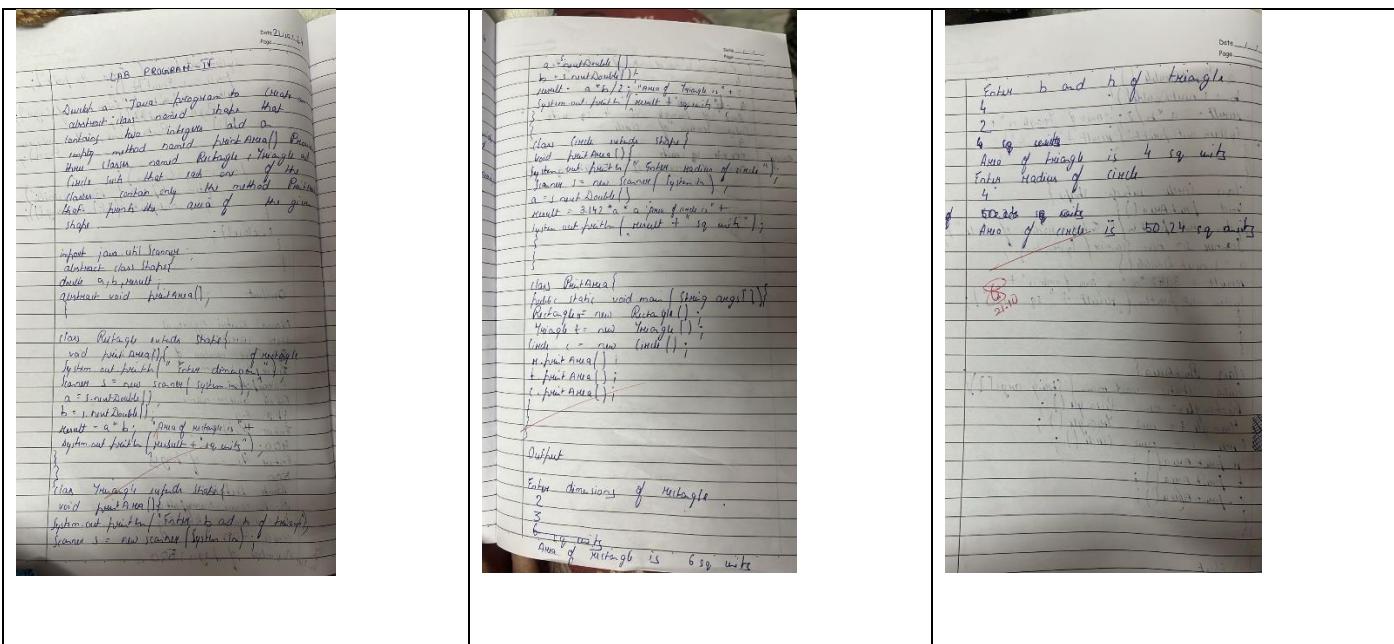
D:\>

```

Program 4

Area of the Shape

Algorithm:



Code:

```

import java.util.Scanner;
class PrintInfo {
    static void print() {
        System.out.println("Name: Kashvi Agarwal");
        System.out.println("USN: 1BM23CS141");
    }
}

abstract class shape{
    int dim1;
    int dim2;
}

```

```

abstract void printarea();
}

class rectangle extends shape{
    public rectangle(){
        this.dim1=dim1;
        this.dim2=dim2;
    }
    public void printarea(){
        Scanner s = new Scanner(System.in);
        System.out.println("enter the l and b");
        dim1=s.nextInt();
        dim2=s.nextInt();

        int area=dim1*dim2;
        System.out.println("area of rectangle: "+area);
    }
}
class triangle extends shape{
    public triangle(){
        this.dim1=dim1;
        this.dim2=dim2;
    }
    public void printarea(){
        Scanner s = new Scanner(System.in);

        System.out.println("enter the l and b");
        dim1=s.nextInt();
        dim2=s.nextInt();
        double area=(dim1*dim2)/2;
        System.out.println("area of triangle: "+area);
    }
}
class circle extends shape{
    final double Pi=3.14;
    public circle(){
        this.dim1=dim1;
    }
    public void printarea(){
        Scanner s = new Scanner(System.in);

```

```
System.out.println("enter the radius");
dim1=s.nextInt();

double area=Pi*dim1*dim1;
System.out.println("area of circle: "+area);
}

}

public class main{
    public static void main (String [] args){
        PrintInfo.print();

rectangle R =new rectangle();

R.printarea();

triangle T = new triangle();
T.printarea();

circle C = new circle();
C.printarea();
}

}
```

Output:

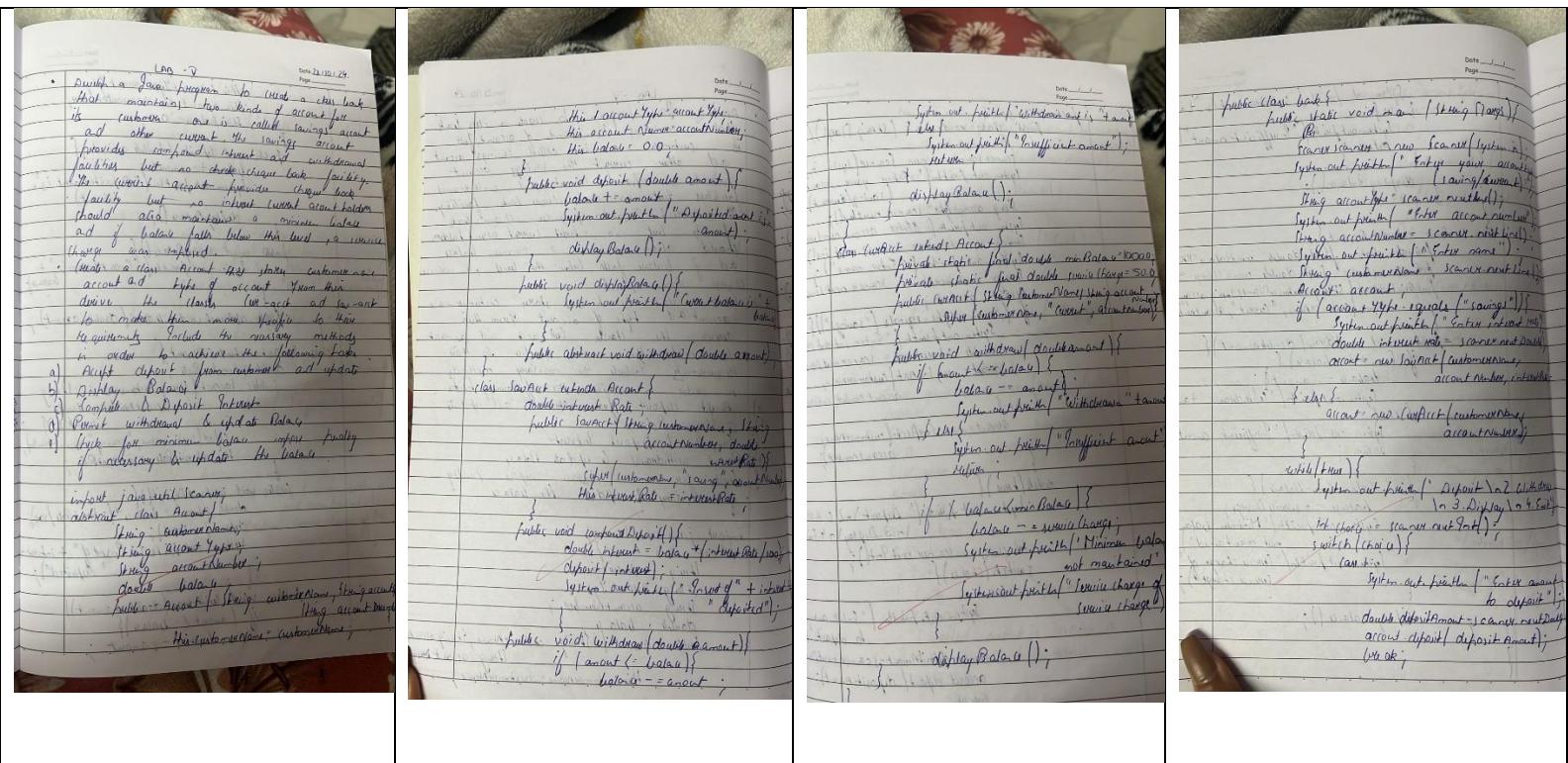
```
D:\1bm23cs141>javac main.java
D:\1bm23cs141>java main
Name: Kashvi Agarwal
USN: 1BM23CS141
enter the l and b
2
3
area of rectangle: 6
enter the l and b
4
2
area of triangle: 4.0
enter the radius
4
area of circle: 50.24

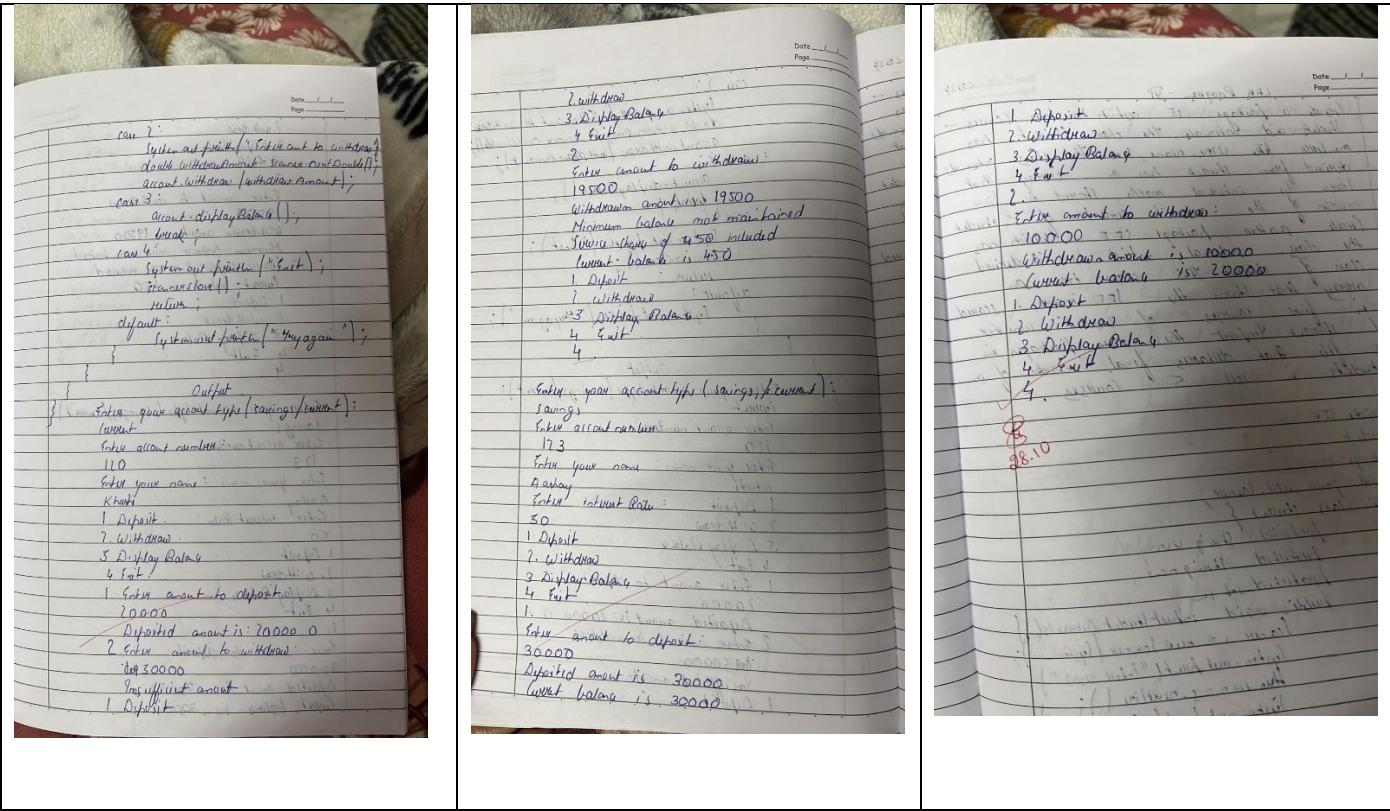
D:\1bm23cs141>
```

Program 5

Bank

Algorithm:





Code:

```

import java.util.Scanner;
class PrintInfo {
    static void print() {
        System.out.println("Name: Kashvi Agarwal");
        System.out.println("USN: 1BM23CS141");
    }
}

```

```

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

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

```

```

        this.accountNumber = accountNumber;
        this.balance = 0.0;
    }

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

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

    public abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate;

    public SavAcct(String customerName, String accountNumber, double interestRate) {
        super(customerName, "savings", accountNumber);
        this.interestRate = interestRate;
    }

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

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

```

```

class CurAcct extends Account {
    private static final double minBalance = 1000.0;
    private static final double serviceCharge = 50.0;

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

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

        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Minimum balance not maintained");
            System.out.println("Service charge of: " + serviceCharge + " included");
        }
        displayBalance();
    }
}

public class bank {
    public static void main(String[] args) {
        PrintInfo.print();

        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter your account type (savings/current):");
        String accountType = scanner.nextLine();
        System.out.println("Enter account number:");
        String accountNumber = scanner.nextLine();
        System.out.println("Enter your name:");
        String customerName = scanner.nextLine();

        Account account;
        if (accountType.equals("savings")) {
            System.out.println("Enter the interest rate:");
        }
    }
}

```

```

        double interestRate = scanner.nextDouble();
        account = new SavAcct(customerName, accountNumber, interestRate);
    } else {
        account = new CurAcct(customerName, accountNumber);
    }

    while (true) {
        System.out.println("1. Deposit\n2. Withdraw\n3. Display Balance\n4. Exit");
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                System.out.println("Enter amount to deposit:");
                double depositAmount = scanner.nextDouble();
                account.deposit(depositAmount);
                break;
            case 2:
                System.out.println("Enter amount to withdraw:");
                double withdrawAmount = scanner.nextDouble();
                account.withdraw(withdrawAmount);
                break;
            case 3:
                account.displayBalance();
                break;
            case 4:
                System.out.println("Exit");
                scanner.close();
                return;
            default:
                System.out.println("Try again");
        }
    }
}

```

Output:

```
D:\lbtm23cs141>javac bank.java  
D:\lbtm23cs141>java bank  
Name: Kashvi Agarwal  
USN: 1BM23CS141  
Enter your account type (savings/current):  
savings  
Enter account number:  
123  
Enter your name:  
Himika  
Enter the interest rate:  
30  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Exit  
1  
Enter amount to deposit:  
30000  
Deposited amount is: 30000.0  
Current balance is: 30000.0  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Exit  
2  
Enter amount to withdraw:  
10000  
Withdrawn amount is: 10000.0  
Current balance is: 20000.0  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Exit  
|
```

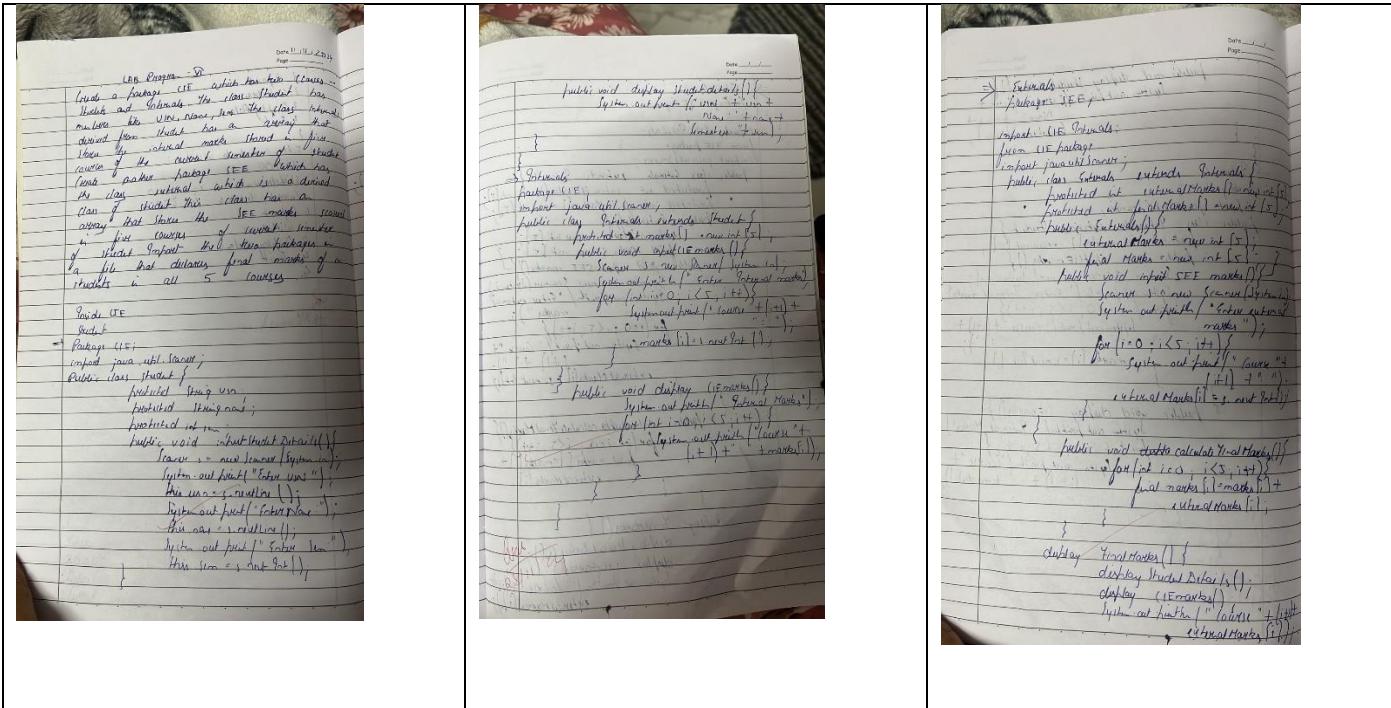
```
Enter your name:  
Himika  
Enter the interest rate:  
30  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Exit  
1  
Enter amount to deposit:  
30000  
Deposited amount is: 30000.0  
Current balance is: 30000.0  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Exit  
2  
Enter amount to withdraw:  
10000  
Withdrawn amount is: 10000.0  
Current balance is: 20000.0  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Exit  
3  
Current balance is: 20000.0  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Exit  
1  
Enter amount to deposit:  
30000  
Deposited amount is: 30000.0  
Current balance is: 30000.0  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Exit  
2  
Enter amount to withdraw:  
40000  
Insufficient amount for withdrawal.  
1. Deposit  
2. Withdraw  
3. Display Balance  
4. Exit  
|
```

```
D:\lbtm23cs141>javac bank.java
D:\lbtm23cs141>java bank
Name: Kashvi Agarwal
USN: 1BM23CS141
Enter your account type (savings/current):
current
Enter account number:
120
Enter your name:
Khushi
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
1
Enter amount to deposit:
20000
Deposited amount is: 20000.0
Current balance is: 20000.0
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
2
Enter amount to withdraw:
30000
Insufficient amount for withdrawal.
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
2
Enter amount to withdraw:
19500
Withdrawn amount is: 19500.0
Minimum balance not maintained
Service charge of: 50.0 included
Current balance is: 450.0
1. Deposit
2. Withdraw
3. Display Balance
4. Exit
3
Current balance is: 450.0
```

Program 6

Packages

Algorithm:



<pre> super out.println("Enter Marks"); for(int i=0; i<5; i++) { System.out.print("Course " + (i+1) + ": "); marks[i] = scanner.nextInt(); } System.out.println("Internal Marks: " + marks); </pre>	<p>Output</p> <table border="1"> <thead> <tr> <th colspan="2">Index number of student 2</th> </tr> </thead> <tbody> <tr> <td>Enter USN : 10M23CS112</td> <td>Course 1 : 30</td> </tr> <tr> <td>Enter Name : Anil</td> <td>Course 2 : 40</td> </tr> <tr> <td>Enter IEE Internals : 10</td> <td>Course 3 : 20</td> </tr> <tr> <td>Enter IEE Project : 10</td> <td>Course 4 : 60</td> </tr> <tr> <td>Enter IEE Final : 100</td> <td>Course 5 : 30</td> </tr> <tr> <td>Enter External Marks : 100</td> <td></td> </tr> <tr> <td>Course 1 : 30</td> <td>Enter Internal Marks : 100</td> </tr> <tr> <td>Course 2 : 40</td> <td>Course 1 : 30</td> </tr> <tr> <td>Course 3 : 20</td> <td>Course 2 : 40</td> </tr> <tr> <td>Course 4 : 60</td> <td>Course 3 : 20</td> </tr> <tr> <td>Course 5 : 30</td> <td>Course 4 : 60</td> </tr> <tr> <td></td> <td>Course 5 : 30</td> </tr> <tr> <td></td> <td>Enter External Marks : 100</td> </tr> <tr> <td></td> <td>Course 1 : 30</td> </tr> <tr> <td></td> <td>Course 2 : 40</td> </tr> <tr> <td></td> <td>Course 3 : 20</td> </tr> <tr> <td></td> <td>Course 4 : 60</td> </tr> <tr> <td></td> <td>Course 5 : 30</td> </tr> </tbody> </table>	Index number of student 2		Enter USN : 10M23CS112	Course 1 : 30	Enter Name : Anil	Course 2 : 40	Enter IEE Internals : 10	Course 3 : 20	Enter IEE Project : 10	Course 4 : 60	Enter IEE Final : 100	Course 5 : 30	Enter External Marks : 100		Course 1 : 30	Enter Internal Marks : 100	Course 2 : 40	Course 1 : 30	Course 3 : 20	Course 2 : 40	Course 4 : 60	Course 3 : 20	Course 5 : 30	Course 4 : 60		Course 5 : 30		Enter External Marks : 100		Course 1 : 30		Course 2 : 40		Course 3 : 20		Course 4 : 60		Course 5 : 30	<p>Final Marks</p> <table border="1"> <thead> <tr> <th>Course</th> <th>Marks</th> </tr> </thead> <tbody> <tr> <td>Course 1</td> <td>60</td> </tr> <tr> <td>Course 2</td> <td>80</td> </tr> <tr> <td>Course 3</td> <td>50</td> </tr> <tr> <td>Course 4</td> <td>100</td> </tr> <tr> <td>Course 5</td> <td>500</td> </tr> </tbody> </table> <p>UIN : 10M23CS112 Name : Anil Index number : 2</p>	Course	Marks	Course 1	60	Course 2	80	Course 3	50	Course 4	100	Course 5	500
Index number of student 2																																																				
Enter USN : 10M23CS112	Course 1 : 30																																																			
Enter Name : Anil	Course 2 : 40																																																			
Enter IEE Internals : 10	Course 3 : 20																																																			
Enter IEE Project : 10	Course 4 : 60																																																			
Enter IEE Final : 100	Course 5 : 30																																																			
Enter External Marks : 100																																																				
Course 1 : 30	Enter Internal Marks : 100																																																			
Course 2 : 40	Course 1 : 30																																																			
Course 3 : 20	Course 2 : 40																																																			
Course 4 : 60	Course 3 : 20																																																			
Course 5 : 30	Course 4 : 60																																																			
	Course 5 : 30																																																			
	Enter External Marks : 100																																																			
	Course 1 : 30																																																			
	Course 2 : 40																																																			
	Course 3 : 20																																																			
	Course 4 : 60																																																			
	Course 5 : 30																																																			
Course	Marks																																																			
Course 1	60																																																			
Course 2	80																																																			
Course 3	50																																																			
Course 4	100																																																			
Course 5	500																																																			

Code: CIE

Internals.java

```

package CIE;

import java.util.Scanner;

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

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

    public void displayCIEmarks() {
        System.out.println("Internal Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + marks[i]);
        }
    }
}

```

```
        }
    }
}
```

Student.java

```
package CIE;

import java.util.Scanner;

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

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

SEE:

Student.java

```
package SEE;
```

```
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals {
protected int externalMarks[] = new int[5];
protected int finalMarks[] = new int[5];
```

```

public Externals() {      externalMarks =
new int[5];
    finalMarks = new int[5];
}
public void inputSEEmarks() {
Scanner s = new Scanner(System.in);
    System.out.println("Enter External Marks for 5 courses: ");
    for (int i = 0; i < 5; i++) {
        System.out.print("Course " + (i + 1) + ": ");
        externalMarks[i] = s.nextInt();
    }
}
public void calculateFinalMarks() {
for (int i = 0; i < 5; i++) {
    finalMarks[i] = marks[i] + externalMarks[i];
}
}
public void displayFinalMarks() {
displayStudentDetails();
    displayCIEmarks();

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

```

Main.java

```

import SEE.Externals;

import java.util.Scanner;

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

```

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

Externals[] students = new Externals[n];

for (int i = 0; i < n; i++) {
    students[i] = new Externals();
    students[i].inputStudentDetails();
    students[i].inputCIEmarks();
    students[i].inputSEEmarks();
    students[i].calculateFinalMarks();
}

for (int i = 0; i < n; i++) {
    students[i].displayFinalMarks();
    System.out.println("-----");
}
}
```

Output:

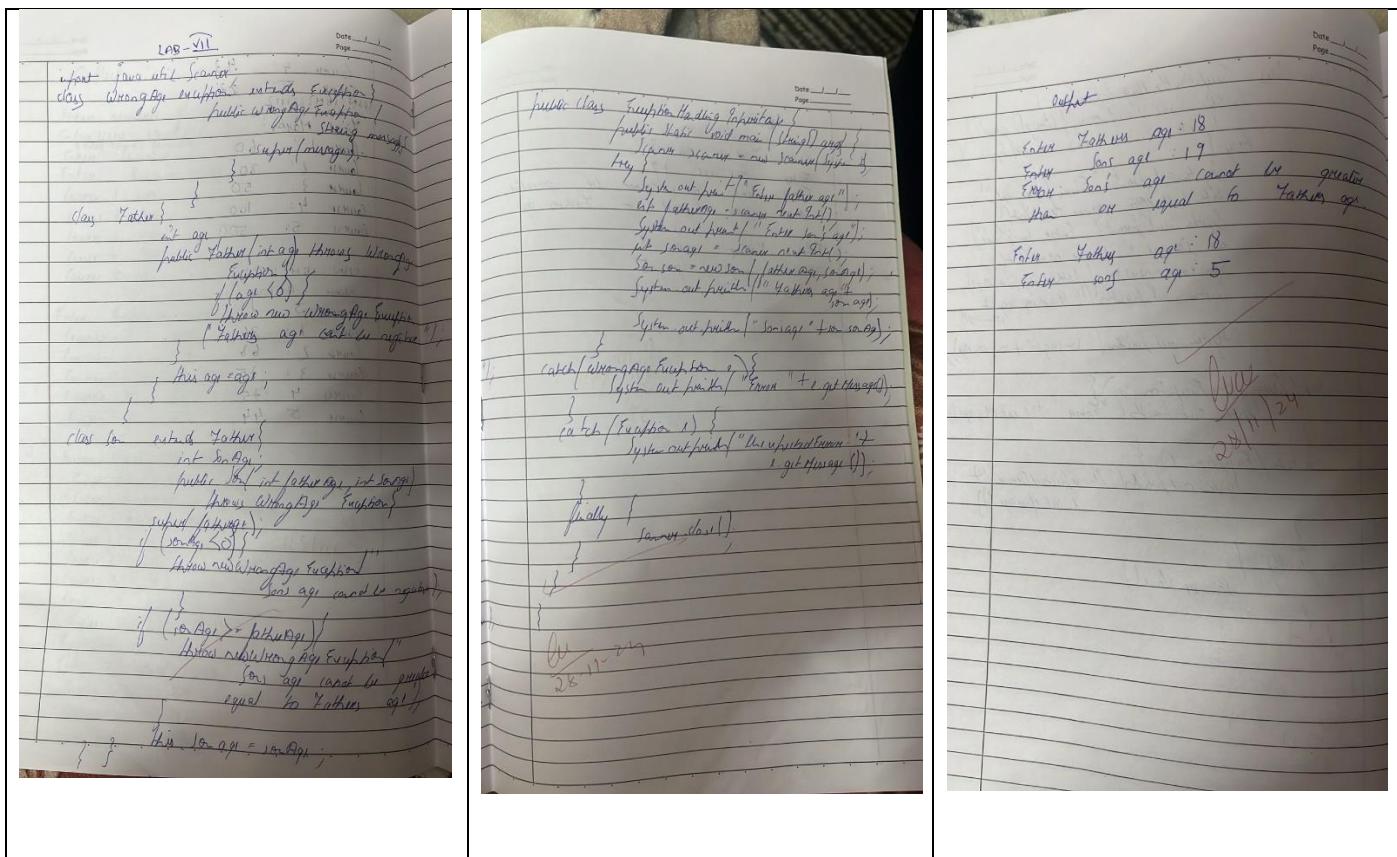
```
C:\Users\STUDENT\Desktop\1BMK>javac -d . CIE/Student.java  
C:\Users\STUDENT\Desktop\1BMK>javac -d . CIE/Internals.java  
C:\Users\STUDENT\Desktop\1BMK>javac -d . SEE/Externals.java  
C:\Users\STUDENT\Desktop\1BMK>javac Main.java  
  
C:\Users\STUDENT\Desktop\1BMK>java Main  
Enter number of students: 2  
Enter USN: 1bm23cs121  
Enter Name: himika  
Enter Semester: 3  
Enter Internal Marks for 5 courses:  
Course 1: 30  
Course 2: 40  
Course 3: 20  
Course 4: 50  
Course 5: 30  
Enter External Marks for 5 courses:  
Course 1: 30  
Course 2: 40  
Course 3: 30  
Course 4: 50  
Course 5: 20  
Enter USN: 1bm23cs122  
Enter Name: chiraiya  
Enter Semester: 2  
Enter Internal Marks for 5 courses:  
Course 1: 12  
Course 2: 23  
Course 3: 43  
Course 4: 32  
Course 5: 21  
Enter External Marks for 5 courses:  
Course 1: 24  
Course 2: 45  
Course 3: 32  
Course 4: 43  
Course 5: 23  
USN: 1bm23cs121  
Name: himika  
Semester: 3  
Internal Marks:  
Course 1: 30  
Course 2: 40  
Course 3: 20  
Course 4: 50  
Course 5: 30
```

```
Course 5: 21  
Enter External Marks for 5 courses:  
Course 1: 24  
Course 2: 45  
Course 3: 32  
Course 4: 43  
Course 5: 23  
USN: 1bm23cs121  
Name: himika  
Semester: 3  
Internal Marks:  
Course 1: 30  
Course 2: 40  
Course 3: 20  
Course 4: 50  
Course 5: 30  
External Marks:  
Course 1: 30  
Course 2: 40  
Course 3: 30  
Course 4: 50  
Course 5: 20  
Final Marks:  
Course 1: 60  
Course 2: 88  
Course 3: 50  
Course 4: 100  
Course 5: 50  
-----  
USN: 1bm23cs122  
Name: chiraiya  
Semester: 2  
Internal Marks:  
Course 1: 12  
Course 2: 23  
Course 3: 43  
Course 4: 32  
Course 5: 21  
External Marks:  
Course 1: 24  
Course 2: 45  
Course 3: 32  
Course 4: 43  
Course 5: 23  
Final Marks:  
Course 1: 36  
Course 2: 68  
Course 3: 75  
Course 4: 75  
Course 5: 44  
-----
```

Program 7

Exception Handling Inheritance

Algorithm:



Code:

```

import java.util.Scanner;

// Custom exception for invalid ages
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

// Father class
class Father {


```

```

int age;

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

// Son class extending Father
class Son extends Father {
    int sonAge;

    // Constructor
    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;
    }
}

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

        try {
            // Input for Father's age
            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();

            // Input for Son's age
            System.out.print("Enter Son's age: ");
            int sonAge = scanner.nextInt();
        }
    }
}

```

```
// Create Son object
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:

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91889\OneDrive\Desktop\BMS PDF>javac ExceptionHandlingInheritance.java

C:\Users\91889\OneDrive\Desktop\BMS PDF>java ExceptionHandlingInheritance
Enter Father's age: 18
Enter Son's age: 5
Father's age: 18
Son's age: 5

C:\Users\91889\OneDrive\Desktop\BMS PDF>15
'15' is not recognized as an internal or external command,
operable program or batch file.

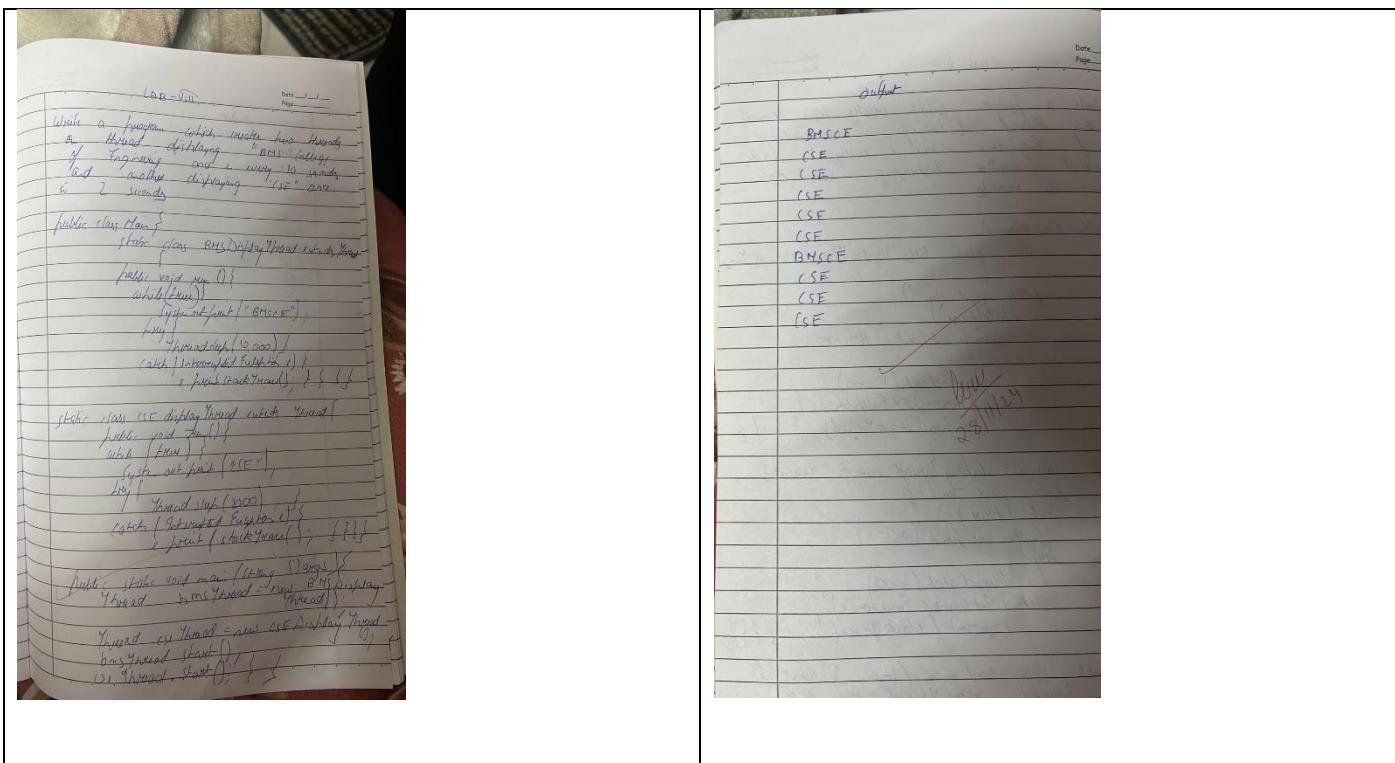
C:\Users\91889\OneDrive\Desktop\BMS PDF>javac ExceptionHandlingInheritance.java

C:\Users\91889\OneDrive\Desktop\BMS PDF>java ExceptionHandlingInheritance
Enter Father's age: 15
Enter Son's age: 16
Error: Son's age cannot be greater than or equal to Father's age.
```

Program 8

Threads

Algorithm:



Code:

```
public class Main {
```

```
// Thread to display "BMS College of Engineering" every 10 seconds
static class BMSDisplayThread extends Thread {
    public void run() {
        while (true) {
            System.out.println("BMS College of Engineering");
            try {
                Thread.sleep(10000); // sleep for 10 seconds
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

}

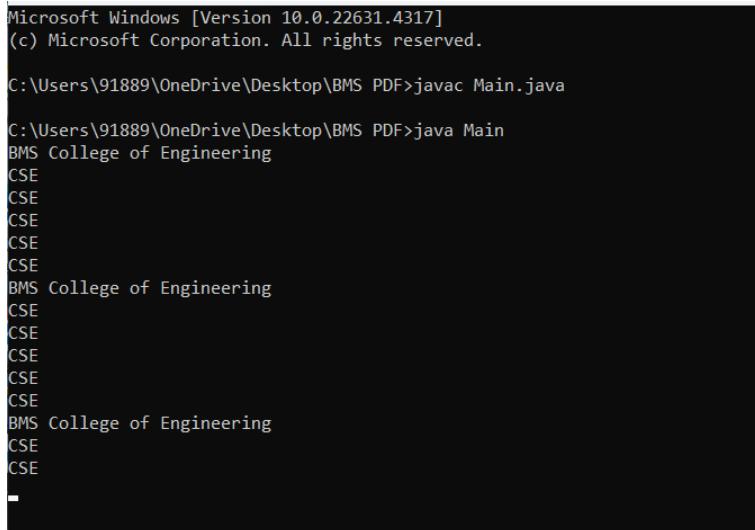
// Thread to display "CSE" every 2 seconds
static class CSEDisplayThread extends Thread {
    public void run() {
        while (true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000); // sleep for 2 seconds
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public static void main(String[] args) {
    // Create and start both threads
    Thread bmsThread = new BMSDisplayThread();
    Thread cseThread = new CSEDisplayThread();

    bmsThread.start();
    cseThread.start();
}
}

```

Output:



```

Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

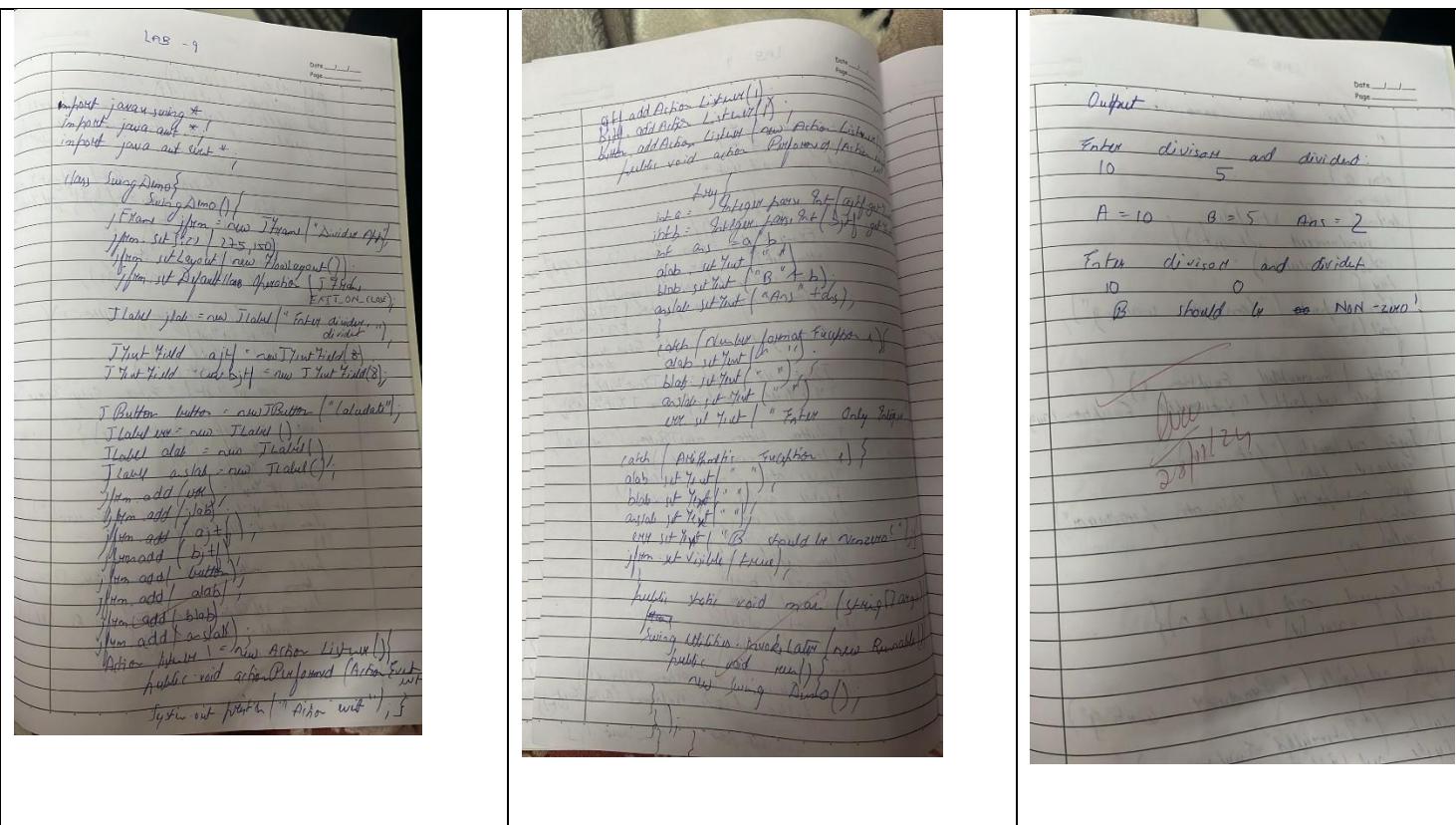
C:\Users\91889\OneDrive\Desktop\BMS PDF>javac Main.java

C:\Users\91889\OneDrive\Desktop\BMS PDF>java Main
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
-
```

Program 9

Swing Demo

Algorithm:



Code:

```

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

class SwingDemo {
    SwingDemo() {
        // Create JFrame container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 200);
        jfrm.setLayout(new FlowLayout());

        // To terminate the program when the window closes
    }
}

```

```

jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Label prompting user
JLabel jlab = new JLabel("Enter the divisor and dividend:");

// Create text fields for input
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);

// Create button to trigger calculation
JButton button = new JButton("Calculate");

// Labels to display output and errors
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();

// Add components to the frame
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
jfrm.add(err);

// Action listener for button click
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            // Parse integers from text fields
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());

            // Perform division
            int ans = a / b;

            // Display results
            alab.setText("A = " + a);
            blab.setText("B = " + b);
        }
    }
});

```

```

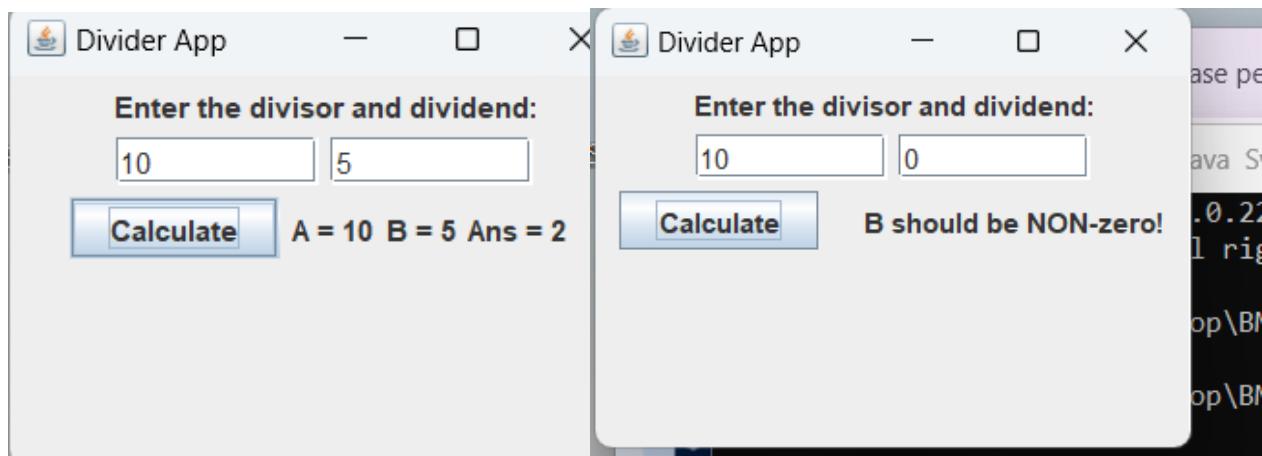
        anslab.setText("Ans = " + ans);
        err.setText("");
    } catch (NumberFormatException e) {
        // Handle non-integer input
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("Enter Only Integers!");
    } catch (ArithmaticException e) {
        // Handle division by zero
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("B should be NON-zero!");
    }
}
});

// Display the frame
jfrm.setVisible(true);
}

public static void main(String[] args) {
    // Create the frame on the Event Dispatch Thread
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SwingDemo();
        }
    });
}
}

```

Output:



Program 10 a

Deadlock

Algorithm:

Date _____
Page _____

Date _____
Page _____

Date _____
Page _____

```

    class Q {
        int n;
        LinkedList<value> q = new LinkedList<value>();
        synchronized void put(value vt) {
            q.add(vt);
        }
        synchronized value get() {
            return q.remove(0);
        }
        synchronized void print() {
            System.out.print("Consumer waiting");
        }
        synchronized void print() {
            System.out.println("Put " + n);
        }
        synchronized void print() {
            System.out.println("Producer working");
        }
        synchronized void print() {
            System.out.println("Producer waiting");
        }
    }

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

    class Consumer implements Runnable {
        Q q;
        Thread t;
        public void run() {
            int n = 0;
            while (n < 15) {
                int m = q.get();
                System.out.println("Consumer " + m);
                n++;
            }
        }
    }

    public class P {
        public static void main(String args[]) {
            Q q = new Q();
            new Producer(q).start();
            new Consumer(q).start();
            System.out.println("Press control C to stop");
        }
    }

```

```

class A {
    // Deadlock
    synchronized void foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.foo");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("A interrupted");
        }
        System.out.println("A returning to main");
        b.lost();
    }
}

class B {
    synchronized void lost(A a) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered B.lost");
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("B interrupted");
        }
        System.out.println("B returning to main");
        a.lost();
    }

    void lost() {
        System.out.println("Inside A.lost");
    }
}

```

Thread current Thread: Main (a main thread)
 Thread b = new Thread(this, "Racing Thread");
 b.start();
 a.lost();
 b.lost(); // Break in main

```

public void run() {
    try {
        b.lost();
        System.out.println("Break in other thread");
    } catch (Exception e) {
        System.out.println("Inside main");
    }
}

Output
Main Thread entered A.foo
Racing Thread entered B.lost
Main thread sleeping to call B.lost()
Racing Thread trying to call A.lost()
Break in other thread
Inside A.lost
Break in main thread
  
```

26/11/2021

Code:

```
class A {
```

```

    synchronized void foo(B b) {

        String name =
        Thread.currentThread().getName();

        System.out.println(name + " entered A.foo");

        try {

            Thread.sleep(1000);

        } catch(Exception e) {

            System.out.println("A Interrupted");

        }
    }
}
```

```
System.out.println(name + " trying to call B.last()");

b.last();

}

void last() {

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

}

}

class B {

synchronized void bar(A a) {

String name =
Thread.currentThread().getName();

System.out.println(name + " entered B.bar");

try {

Thread.sleep(1000);

} catch(Exception e) {

System.out.println("B Interrupted");

}

System.out.println(name + " trying to call A.last()");

a.last();

}

void last() {
```

```

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

}

}

class Deadlock implements Runnable
{

A a = new A();

B b = new B();

Deadlock() {

    Thread.currentThread().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: KASHVI AGARWAL, USN: 1BM23CS141");
    new Deadlock();
}

```

Program 10 b

PCfixed

Code:

```
class Q {  
  
    int n;  
  
    boolean valueSet = false;  
  
    synchronized int get() {  
  
        while(!valueSet)  
            try {  
  
                System.out.println("\nConsumer waiting\n");  
  
                wait();  
  
            } catch(InterruptedException e) {  
  
                System.out.println("InterruptedException caught");  
  
            }  
  
        System.out.println("Got: " + n);  
  
        valueSet = false;  
  
        System.out.println("\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: KASHVI AGARWAL, USN: 1BM23CS141");

    Q q = new Q();

    new Producer(q);

    new Consumer(q);

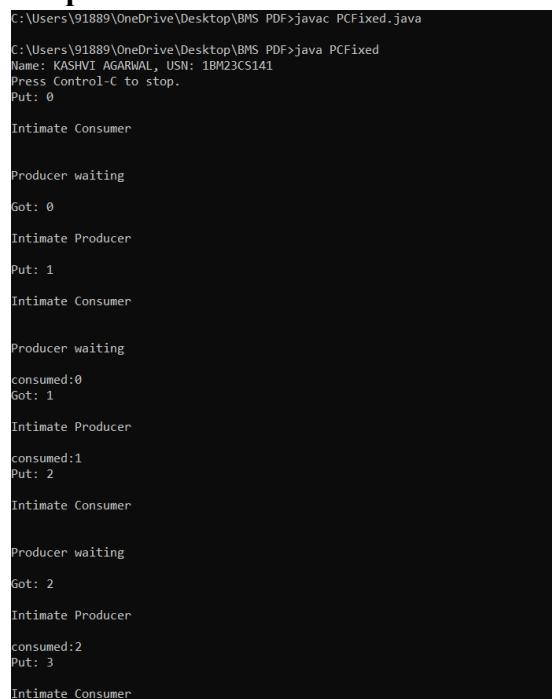
    System.out.println("Press Control-C to stop.");

}

}

```

Output:



C:\Users\91889\OneDrive\Desktop\BMS PDF>javac PCFixed.java
C:\Users\91889\OneDrive\Desktop\BMS PDF>java PCFixed
Name: KASHVI AGARWAL, USN: 1BM23CS141
Press Control-C to stop.
Put: 0
Intimate Consumer
Producer waiting
Got: 0
Intimate Producer
Put: 1
Intimate Consumer
Producer waiting
consumed:0
Got: 1
Intimate Producer
consumed:1
Put: 2
Intimate Consumer
Producer waiting
Got: 2
Intimate Producer
consumed:2
Put: 3
Intimate Consumer

```
 ②: \Windows\System32\cmd.exe
Producer waiting
Got: 8
Intimate Producer
consumed:8
Put: 9
Intimate Consumer
Got: 9
Producer waiting
Got: 9
Intimate Producer
consumed:9
Put: 10
Intimate Consumer
Got: 10
Producer waiting
Got: 10
Intimate Producer
consumed:10
Put: 11
Intimate Consumer
Got: 11
Producer waiting
Got: 11
Intimate Producer
consumed:11
Put: 12
Intimate Consumer
Got: 12
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
C:\Windows\System32\cmd.exe
```

```
(c) Microsoft Corporation. All rights reserved.
C:\Users\91889\OneDrive\Desktop\BMS PDF>javac Deadlock.java
C:\Users\91889\OneDrive\Desktop\BMS PDF>java Deadlock
Name: KASHVI AGARWAL, USN: 1BM23CS141
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
RacingThread trying to call A.last()
Inside A.last
Back in otherthread
Inside A.last
Back in mainthread
C:\Users\91889\OneDrive\Desktop\BMS PDF>
```