

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



**Object Oriented Java Programming
(23CS3PCOOJ)**

Submitted by

Kishley Sharma (1BM23CS153)

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 Krithika H Kotian(1BM23CS159), who is a 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

S_no.	Date	Experiment title
1.	30/09/24	Quadratic
2.	07/10/24	SGPA
3.	14/10/24	BookDemo
4.	21/10/24	ShapeArea Calculator
5.	21/10/24	Bank Account
6.	04/11/24	Packages
7.	28/11/24	Exceptions
8.	28/11/24	Threads
9.	28/11/24	SwingDemo
10(A)	28/11/24	PCfixed
10(B)	28/11/24	Deadlock

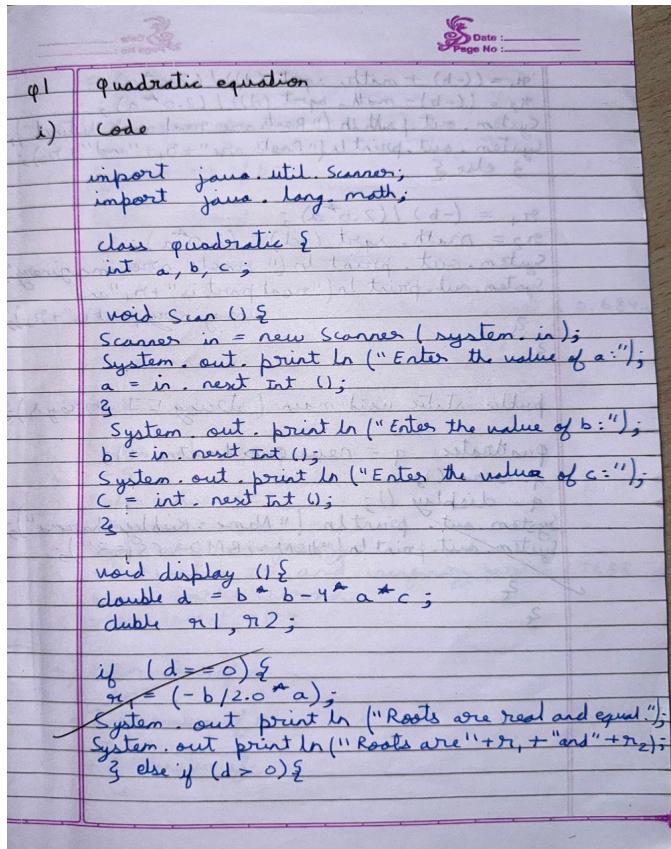
Github Link:

<https://github.com/KishleySharma/JaVALAB>

Program 1

Implement Quadratic Equation

Algorithm:



q1 quadratic equation

i) Code

```
import java.util.Scanner;
import java.lang.Math;

class quadratic {
    int a, b, c;
    void Scan () {
        Scanner in = new Scanner (System.in);
        System.out.print ("Enter the value of a:");
        a = in.nextInt ();
        System.out.print ("Enter the value of b:");
        b = in.nextInt ();
        System.out.print ("Enter the value of c:");
        c = in.nextInt ();
    }
    void display () {
        double d = b * b - 4 * a * c;
        double r1, r2;
        if (d == 0) {
            r1 = (-b / 2.0 * a);
            System.out.println ("Roots are real and equal.");
            System.out.println ("Roots are " + r1 + " and " + r2);
        } else if (d > 0) {
            r1 = (-b + Math.sqrt (d)) / (2.0 * a);
            r2 = (-b - Math.sqrt (d)) / (2.0 * a);
            System.out.println ("Roots are real and distinct.");
            System.out.println ("Roots are " + r1 + " and " + r2);
        } else {
            r1 = r2 = (-b) / (2.0 * a);
            System.out.println ("Roots are complex and conjugate.");
            System.out.println ("Roots are " + r1 + " and " + r2);
        }
    }
}
```

Date : _____
Page No. : _____

```

 $r_1 = ((-b) + \sqrt{d}) / (2.0 * a);$ 
 $r_2 = ((-b) - \sqrt{d}) / (2.0 * a);$ 
System.out.println("Roots are real and distinct.");
System.out.print(" Roots are " + r1 + " and " + r2);
3 else {
     $r_1 = (-b) / (2.0 * a);$ 
 $r_2 = \sqrt{-d} / (2.0 * a);$ 
System.out.println(" Roots are imaginary.");
System.out.print("real part is " + r1 + " and "
    "imaginary part is " + r2);
}
3
public static void main (String [] args) {
    Quadratic q = new Quadratic ();
    q.read ();
    q.display ();
    System.out.println ("Name : Kishley Sharma");
    System.out.println ("USN : IBM23CS153");
}

```

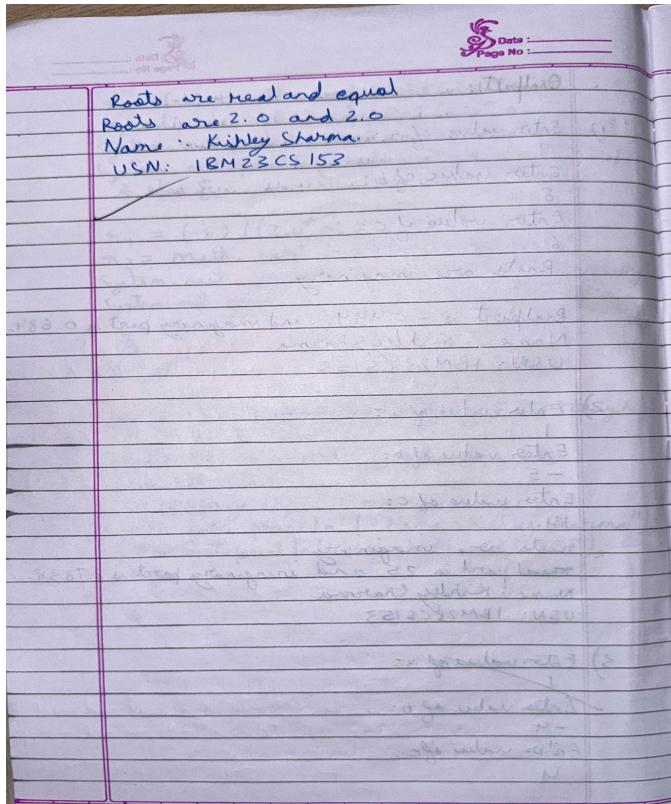
Date : _____
Page No. : _____

Output:

1) Enter value of a:
9
Enter value of b:
8
Enter value of c:
6
Roots are imaginary
Real part is -0.444... and imaginary part is 0.684...
Name : Kishley Sharma
USN : IBM23CS153

2) Enter value of a:
1
Enter value of b:
-5
Enter value of c:
14
Roots are imaginary
Real part is 2.5 and imaginary part is 2.7838...
Name : Kishley Sharma
USN : IBM23CS153

3) Enter value of a:
1
Enter value of b:
-4
Enter value of c:
4



Code:

```
import java.util.Scanner;  
  
import java.lang.Math;  
  
class Quadratic {  
    int a, b, c;  
  
    void scan() {  
        Scanner in = new Scanner(System.in);  
        System.out.println("Enter the value of a:");  
        a = in.nextInt();  
        if (a == 0) {  
            System.out.println("Value of a is invalid. Enter a non-zero value:");  
            a = in.nextInt();  
        }  
        b = in.nextInt();  
        c = in.nextInt();  
        if (a == 0) {  
            System.out.println("Value of a is invalid. Enter a non-zero value:");  
            a = in.nextInt();  
        }  
        if (b == 0) {  
            System.out.println("Value of b is invalid. Enter a non-zero value:");  
            b = in.nextInt();  
        }  
        if (c == 0) {  
            System.out.println("Value of c is invalid. Enter a non-zero value:");  
            c = in.nextInt();  
        }  
    }  
  
    void solve() {  
        double D = b * b - 4 * a * c;  
        if (D < 0) {  
            System.out.println("No real roots");  
        } else if (D == 0) {  
            System.out.println("One real root");  
            double x1 = -b / (2 * a);  
            System.out.println("Root is " + x1);  
        } else {  
            System.out.println("Two real roots");  
            double x1 = (-b + Math.sqrt(D)) / (2 * a);  
            double x2 = (-b - Math.sqrt(D)) / (2 * a);  
            System.out.println("Roots are " + x1 + " and " + x2);  
        }  
    }  
}
```

```

    }

    System.out.println("Enter the value of b:");
    b = in.nextInt();

    System.out.println("Enter the value of c:");
    c = in.nextInt();

}

void display() {

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

    double r1, r2;

    if (d == 0) {
        r1 = (-b) / (2.0 * a);
        System.out.println("Roots are real and equal.");
        System.out.println("Roots are " + r1 + " and " + r1);
    } else if (d > 0) {
        r1 = ((-b) + Math.sqrt(d)) / (2.0 * a);
        r2 = ((-b) - Math.sqrt(d)) / (2.0 * a);
        System.out.println("Roots are real and distinct.");
        System.out.println("Roots are " + r1 + " and " + r2);
    } else {
        r1 = (-b) / (2.0 * a);
        r2 = Math.sqrt(-d) / (2.0 * a);
        System.out.println("Roots are imaginary.");
        System.out.println("Real part is " + r1 + " and imaginary part is " + r2);
    }
}
}

public static void main(String[] args) {
    Quadratic q = new Quadratic();
}

```

```

        q.scan();
        q.display();
        System.out.println("Name: Kishley Sharma");
        System.out.println("USN: 1BM23CS153");
    }
}

```

```

C:\Windows\System32\cmd.e  X  +  ▾

D:\Kishley Sharma>
D:\Kishley Sharma>javac Quadratic.java
D:\Kishley Sharma>java Quadratic
Enter the value of a:
9
Enter the value of b:
8
Enter the value of c:
6
Roots are imaginary.
Real part is -0.4444444444444444 and imaginary part is 0.6849348892187751
Name: Kishley Sharma
USN: 1BM23CS153

D:\Kishley Sharma>java Quadratic
Enter the value of a:
1
Enter the value of b:
-5
Enter the value of c:
14
Roots are imaginary.
Real part is 2.5 and imaginary part is 2.7838821814150108
Name: Kishley Sharma
USN: 1BM23CS153

D:\Kishley Sharma>java Quadratic
Enter the value of a:
1
Enter the value of b:
-4
Enter the value of c:
4
Roots are real and equal.
Roots are 2.0 and 2.0
Name: Kishley Sharma
USN: 1BM23CS153

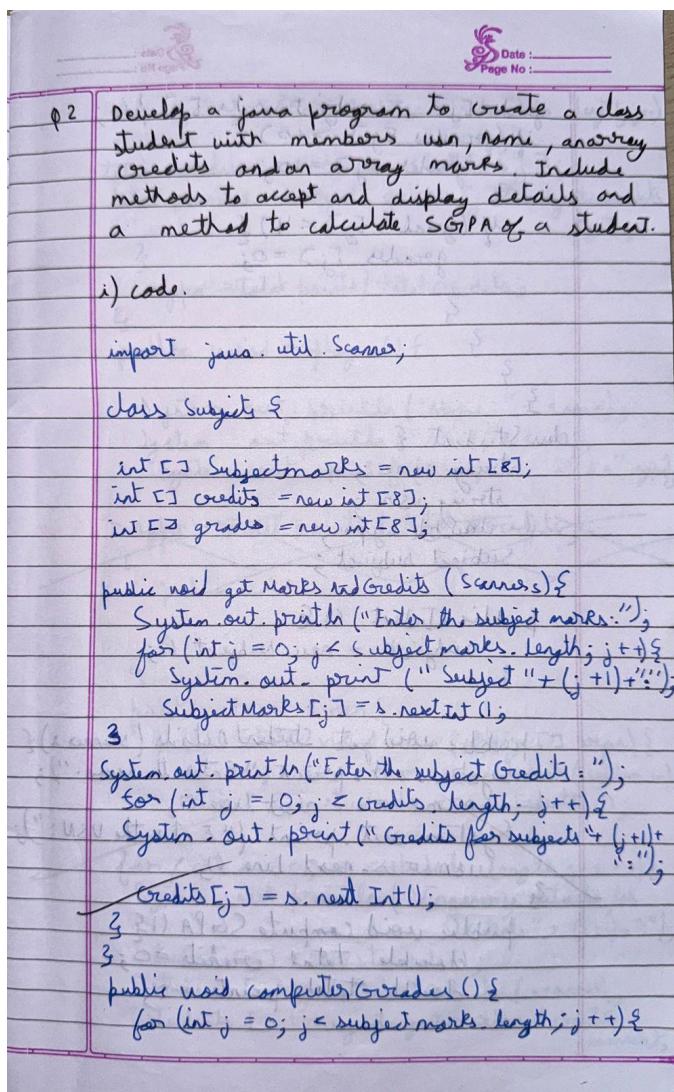
D:\Kishley Sharma>

```

Program 2

Implement SGPA Calculator

Algorithm:



Date : _____
 Page No. : _____

```

grades[j] = subjectmarks[j]/10;
if (grades[j] >= 10) {
  grades[j] = 10;
} else if (grades[j] < 4) {
  grades[j] = 0;
}

class Student {
  String name;
  String usn;
  double sgpa;
  Subject subject;

  public Student() {
    subject = new Subject();
  }

  public void getStudentDetails(Scanner s) {
    System.out.print("Enter the name : ");
    name = s.nextLine();
    System.out.print("Enter the USN : ");
    usn = s.nextLine();
  }

  public void computeSGPA() {
    double totalCredits = 0;
    double totalPoints = 0;
  }
}
  
```

Date : _____
 Page No. : _____

```

for (int j = 0; j < Subject.Credits.length; j++) {
  totalCredits += Subject.Credits[j];
  totalPoints += Subject.grades[j] * Subject.Credits[j];
}

sgpa = totalPoints / totalCredits;

public void display() {
  System.out.println("Name : " + name);
  System.out.println("USN : " + usn);
  System.out.printf("SGPA : %.2f\n", sgpa);
}

public void main() {
  Scanner scanner = new Scanner(System.in);
  Student[] students = new Student[10];

  for (int i = 0; i < students.length; i++) {
    System.out.println("Engineering details for student " + (i + 1));
    students[i] = new Student();
    students[i].getStudentDetails(scanner);
    students[i].subject.getMarksAndCredits(scanner);
  }
}
  
```

Students [x]. subject . Compute Grades () ;
 students [i] . Compute SGPA () ;
 Students [i] . display () ;
 System.out.println ();

 3
 Scanner.close ();

 3
 3
 "Output"

 Entering details for student 1 :
 Enter the name : Kishley Sharma
 Enter the USN : 1BM23CS153
 Enter subject marks :
 Subject 1 : 78
 Subject 2 : 85
 Subject 3 : 76
 Subject 4 : 68
 Subject 5 : 98
 Subject 6 : 39
 Subject 7 : 55
 Subject 8 : 59

 Enter subject credits :
 Credits for Subject 1 : 3
 Credits for Subject 2 : 3
 Credits for Subject 3 : 3
 Credits for Subject 4 : 2

Credits for Subject 5 : 4
 Credits for Subject 6 : 4
 Credits for Subject 7 : 1
 Credits for Subject 8 : 1

 Name : Kishley Sharma
 USN : 1BM23CS153
 SGPA : 5.25

 Output :

 Student Information
 Name : Kishley Sharma
 USN : 1BM23CS153
 SGPA : 5.25

 Student Details
 Name : Kishley Sharma
 USN : 1BM23CS153
 SGPA : 5.25

 Student Marks
 Subject 1 : 78
 Subject 2 : 85
 Subject 3 : 76
 Subject 4 : 68
 Subject 5 : 98
 Subject 6 : 39
 Subject 7 : 55
 Subject 8 : 59

 Total Marks : 617
 SGPA : 5.25

Code:

```
import java.util.Scanner;

class Subject {
    int[] subjectMarks = new int[8];
    int[] credits = new int[8];
    int[] grades = new int[8];

    public void getMarksAndCredits(Scanner s) {
        System.out.println("Enter the subject Marks:");
        for (int j = 0; j < subjectMarks.length; j++) {
            System.out.print("Subject " + (j + 1) + ": ");
            subjectMarks[j] = s.nextInt();
        }

        System.out.println("Enter the subject Credits:");
        for (int j = 0; j < credits.length; j++) {
            System.out.print("Credits for Subject " + (j + 1) + ": ");
            credits[j] = s.nextInt();
        }
    }

    public void computeGrades() {
        for (int j = 0; j < subjectMarks.length; j++) {
            grades[j] = subjectMarks[j] / 10;
            if (grades[j] > 10) {
                grades[j] = 10;
            }
            if (grades[j] < 4) {

```

```

        grades[j] = 0;
    }
}
}
}

class Student {
    String name;
    String usn;
    double sgpa;
    Subject subject;

    public Student() {
        subject = new Subject();
    }

    public void getStudentDetails(Scanner s) {
        System.out.print("Enter the Name: ");
        name = s.nextLine();
        System.out.print("Enter the USN: ");
        usn = s.nextLine();
    }

    public void computeSGPA() {
        double totalCredits = 0;
        double totalPoints = 0;

        for (int j = 0; j < subject.credits.length; j++) {
            totalCredits += subject.credits[j];
            totalPoints += subject.grades[j] * subject.credits[j];
        }
    }
}

```

```

    }

    sgpa = totalPoints / totalCredits;

}

public void display() {

    System.out.println("Name: " + name);
    System.out.println("USN: " + usn);
    System.out.printf("SGPA: %.2f\n", sgpa);

}

}

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        Student[] students = new Student[1];

        for (int i = 0; i < students.length; i++) {
            System.out.println("Entering details for Student " + (i + 1) + ":");
            students[i] = new Student();
            students[i].getStudentDetails(scanner);
            students[i].subject.getMarksAndCredits(scanner);
            students[i].subject.computeGrades();
            students[i].computeSGPA();
            students[i].display();
            System.out.println();
        }

        System.out.println("Name: Kishley Sharma");
        System.out.println("USN: 1BM23CS153");
    }
}

```

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

```
D:\Kishley Sharma>javac Main.java  
  
D:\Kishley Sharma>java Main  
Entering details for Student 1:  
Enter the Name: Kishley Sharma  
Enter the USN: 1BM23CS153  
Enter the subject Marks:  
Subject 1: 78  
Subject 2: 45  
Subject 3: 76  
Subject 4: 68  
Subject 5: 98  
Subject 6: 39  
Subject 7: 55  
Subject 8: 59  
Enter the subject Credits:  
Credits for Subject 1: 3  
Credits for Subject 2: 3  
Credits for Subject 3: 2  
Credits for Subject 4: 2  
Credits for Subject 5: 4  
Credits for Subject 6: 4  
Credits for Subject 7: 1  
Credits for Subject 8: 1  
Name: Kishley Sharma  
USN: 1BM23CS153  
SGPA: 5.25  
  
D:\Kishley Sharma>
```

Program 3

BookDemo

Algorithm:

Q3 Create a class Book which contains four members: name, author, price, numPages. Include a constructor to set the values for the members. Include methods to set and get the details of the object. Include a toString() method that could display the complete details of the book. Develop a Java program to create a book object.

2 (code)

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

```
public static void main (String [] args) {  
    Scanner s = new Scanner (System.in);  
    System.out.print ("Enter the number of books: ");  
    int n = s.nextInt();  
    Book [] books = new Book [n];  
  
    for (int i = 0; i < n; i++) {  
        s.nextLine();  
        System.out.print ("Enter details for Book " +  
                         (i + 1) + ": ");  
  
        System.out.print ("Enter Book Name: ");  
        String name = s.nextLine();  
        System.out.print ("Enter Author Name: ");  
        String author = s.nextLine();  
        System.out.print ("Enter price: ");  
        double price = s.nextDouble();  
  
        System.out.print ("Enter number of pages: ");  
        int numPages = s.nextInt();  
        books [i] = new Book (name, author, price, numPages);  
    }  
  
    System.out.println ("Kishley Sharma");
```

Date : _____
Page No. : _____

```

System.out.println("USN-1BM23CS153");
System.out.print("In Book Details:");
for(int i=0; i<n; i++) {
    System.out.print(" " + (i+1) + ":");
    System.out.println(books[i].toString());
}

```

Output

Enter the number of Books = 2

Enter details for book 1:

Enter book's name: Kirby's empire
 Enter author's name: Kirby
 Enter price = 10,000
 Enter Number of pages = 20

Enter details for book 2:

Enter book's name = Pinky
 Enter author's name = Pinku
 Enter price = 20
 Enter Number of pages = 20

Book Details:

Date : _____
Page No. : _____

Book 1:
 Book name = Kirby's empire
 Author = Kirby
 Price = \$ 10,000
 Number of pages = 20

Book 2:
 Book name = Pinky
 Author = Pinku
 Price = 20
 Number of pages = 20

By 14/10/20

1) Using single inheritance by adding
 (Author) class to books
 2) Author derived class
 3) Encapsulating structure up to each function
 4) Overriding the behaviour
 5) Overloading the behaviour

Code:

```
import java.util.Scanner;

class Book {

    private String name;
    private String author;
    private double price;
    private int numPages;

    Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public String toString() {
        return "Book Name: " + name + "\nAuthor: " + author + "\nPrice: $" + price + "\nNumber of Pages: "
               + numPages;
    }
}

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

    System.out.print("Enter the number of books: ");
    int n = s.nextInt();

    Book[] books = new Book[n];

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

```
s.nextLine();

System.out.println("\nEnter details for Book " + (i + 1) + ":");

System.out.print("Enter Book Name: ");
String name = s.nextLine();

System.out.print("Enter Author Name: ");
String author = s.nextLine();

System.out.print("Enter Price: ");
double price = s.nextDouble();

System.out.print("Enter Number of Pages: ");
int numPages = s.nextInt();

books[i] = new Book(name, author, price, numPages);

}

System.out.println("\nBook Details:");
for (int i = 0; i < n; i++) {
    System.out.println("\nBook " + (i + 1) + ":");

    System.out.println(books[i].toString());
}

// Display the name and USN of the author after the output
System.out.println("\nName: Kishley Sharma");
System.out.println("USN: 1BM23CS153");

s.close();
```

```
}
```

```
D:\Kishley Sharma>javac Book.java

D:\Kishley Sharma>java Book
Enter the number of books: 2

Enter details for Book 1:
Enter Book Name: Kishley's empiore
Enter Author Name: Kishley
Enter Price: 10000
Enter Number of Pages: 20

Enter details for Book 2:
Enter Book Name: pinky
Enter Author Name: pinku
```

```
arma
```

USN-1BM23CS153

Book Details:

Book 1:

Book Name: Kishley's empiore
Author: Kishley
Price: \$10000.0
Number of Pages: 20

Book 2:

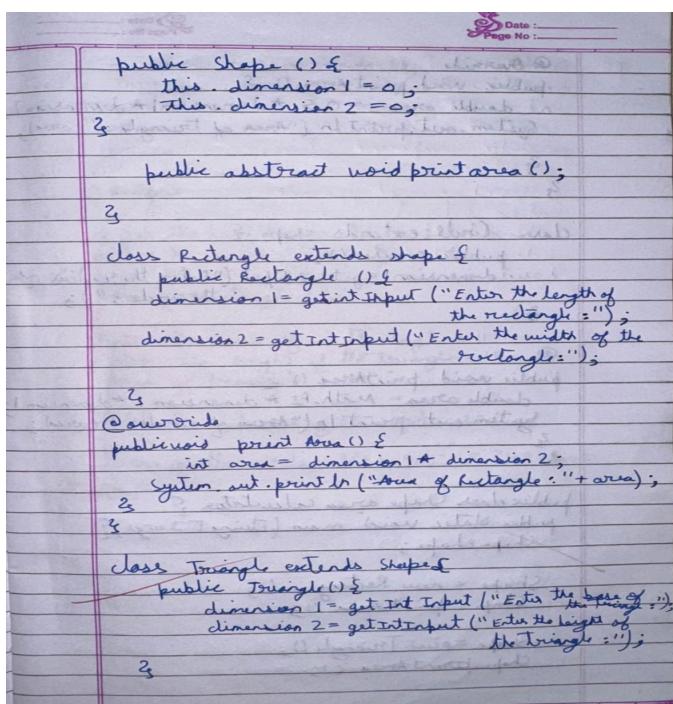
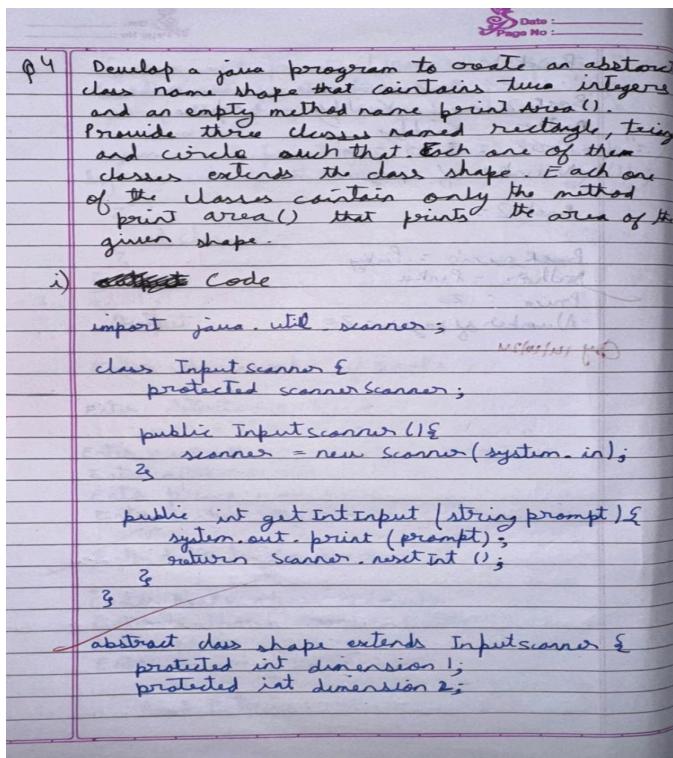
Book Name: pinky
Author: pinku
Price: \$20.0
Number of Pages: 20

```
D:\Kishley Sharma>
```

Program 4

ShapeArea Calculator

Algorithm:



Date : _____
 Page No. : _____

```

  @ Override
  public void printArea () {
    double area = 0.5 * dimension1 * dimension2;
    System.out.println ("Area of Triangle : " + area);
  }

  class Circle extends Shape {
    public Circle () {
      dimension1 = getIntInput ("Enter the radius of the circle : ");
    }
  }

  @
  public void printArea () {
    double area = Math.PI * dimension1 * dimension1;
    System.out.println ("Area of circle : " + area);
  }

  public class ShapeAreaCalculator {
    public static void main (String args) {
      Shape shape;
      shape = new Rectangle ();
      shape.printArea ();
      shape = new Triangle ();
      shape.printArea ();
    }
  }
  
```

Date : _____
 Page No. : _____

```

  Shape = new Circle ();
  Shape.printArea ();

  i) output
  Enter the length of rectangle : 18
  Enter the width of rectangle : 12
  Area of rectangle : 360

  Enter the base of a triangle : 11
  Enter the height of the triangle : 15
  Area of triangle : 82.5

  Enter the radius of the circle : 25
  Area of the circle : 1661.9025137490005
  
```

Code:

```
import java.util.Scanner;

class InputScanner {
    protected Scanner scanner;

    public InputScanner() {
        scanner = new Scanner(System.in);
    }

    public int getIntInput(String prompt) {
        System.out.print(prompt);
        return scanner.nextInt();
    }
}

abstract class Shape extends InputScanner {
    protected int dimension1;
    protected int dimension2;

    // Constructor to initialize dimensions
    public Shape() {
        this.dimension1 = 0;
        this.dimension2 = 0;
    }

    // Abstract method to print area
    public abstract void printArea();
}
```

```

class Rectangle extends Shape {
    public Rectangle() {
        dimension1 = getIntInput("Enter the length of the rectangle: ");
        dimension2 = getIntInput("Enter the width of the rectangle: ");
    }

    @Override
    public void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    public Triangle() {
        dimension1 = getIntInput("Enter the base of the triangle: ");
        dimension2 = getIntInput("Enter the height of the triangle: ");
    }

    @Override
    public void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}

class Circle extends Shape {
    public Circle() {
        dimension1 = getIntInput("Enter the radius of the circle: ");
    }

    @Override

```

```

public void printArea() {
    double area = Math.PI * dimension1 * dimension1;
    System.out.println("Area of Circle: " + area);
}

}

public class ShapeAreaCalculator {
    public static void main(String[] args) {
        Shape shape;
        shape = new Rectangle();
        shape.printArea();
        shape = new Triangle();
        shape.printArea();

        shape = new Circle();
        shape.printArea();

        // Display the name and USN of the author after the output
        System.out.println("Name: Kishley Sharma");
        System.out.println("USN: 1BM23CS153");
    }
}

```

```

D:\Kishley Sharma>javac ShapeAreaCalculator.java

D:\Kishley Sharma>java ShapeAreaCalculator
Enter the length of the rectangle: 18
Enter the width of the rectangle: 20
Area of Rectangle: 360
Enter the base of the triangle: 11
Enter the height of the triangle: 15
Area of Triangle: 82.5
Enter the radius of the circle: 23
Area of Circle: 1661.9025137490005

D:\Kishley Sharma>

```

Program 5

Bank Account

Algorithm:

Q5 Develop a Java program to create a 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.

i) Code

```
import java.util.Scanner;
class account {
    String name;
    int acc_no;
    String acc_type;
    float balance;
    Scanner sc = new Scanner (System.in);
    void read () {
        System.out.println ("Enter the name : ");
        name = sc.nextLine();
        System.out.println ("Enter the initial balance : ");
        balance = sc.nextFloat();
    }
}
```

```
void acceptDeposit (float amount) {
    if (amount > 0)
        balance += amount;
    System.out.println ("Deposited : " + amount);
}
else
    System.out.println ("Deposit must be positive");
}

void withdraw () {
    System.out.println ("Enter the amount to withdraw : ");
    float wdr = sc.nextInt();
    if (wdr <= balance)
        balance -= wdr;
    System.out.println ("Insufficient balance");
}
else
    balance -= wdr;
    System.out.println ("withdraw : " + wdr);
}

void display () {
    System.out.println ("Account details");
    System.out.println ("Customer name : " + name);
}
```

Date : _____
Page No. : _____

```

System.out.println("Account Number : " + acc.no);
System.out.println("Account type : " + acc.type);
System.out.println("Balance : " + balance);
}

class Current extends Account {
    void withdraw() {
        System.out.println("Enter the amount to withdraw");
        float uder = sc.nextfloat();
        if (balance - uder < MIN_BALANCE) {
            System.out.println("Minimum balance existed!");
        } else {
            balance -= uder;
            System.out.println("withdraw : " + uder);
        }
    }

    class Bank {
        public static void main(String args[]) {
            Scanner sc = new Scanner(System.in);
            Account acc;
            System.out.println("choice account type (1: Saving  
2: Current) : ");
        }
    }
}

```

Date : _____
Page No. : _____

```

int acc_type = sc.nextInt();
if (acc_type == 1)
    account = new Savings();
else if (acc_type == 2)
    account = new Current();
else {
    System.out.println("Invalid account type");
    account = new Current();
}
System.out.println("Initial acc type");
account.read();
while (true) {
    System.out.println("-1. Name = -");
    System.out.println("1. Deposit");
    System.out.println("2. withdraw");
    System.out.println("3. Compute Interest for Savings");
    System.out.println("4. Display account details");
    System.out.println("Enter your choice : ");
    int choice = sc.nextInt();
    switch (choice) {
        case 1:
            System.out.print("Enter deposit amount : ");
            float deposit = sc.nextFloat();
            account.Deposit(deposit);
            break;
    }
}

```



Date : _____
Page No. : _____

Case 2 :

```
acc. withdraw ()  
break;
```

Case 3 :

```
if (acc , all type == "Salary");
```

```
System.out.print ("Enter rate, then compound,  
and time.");
```

```
int rate = Sc. nextInt();
```

```
int num = Sc. nextInt();
```

```
int time = Sc. nextInt();
```

```
(acc. component 3rd int (rate, num, time));
```

else

```
System.out.println ("Not salary account");
```

else

```
break;
```

~~Case 4 :~~

```
acc display ();
```

```
break;
```

~~Case 5 :~~

```
System.out.println ("Exits");
```

```
return;
```

```
default:
```

 Anna University
Date : _____
Page No. : _____

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

2
3
9

ii) Output

choose account type

Enter the name:

Kishley

Enter account number:

153

Enter account type (Savings / A/cnt)

Savings

Enter the initial balance

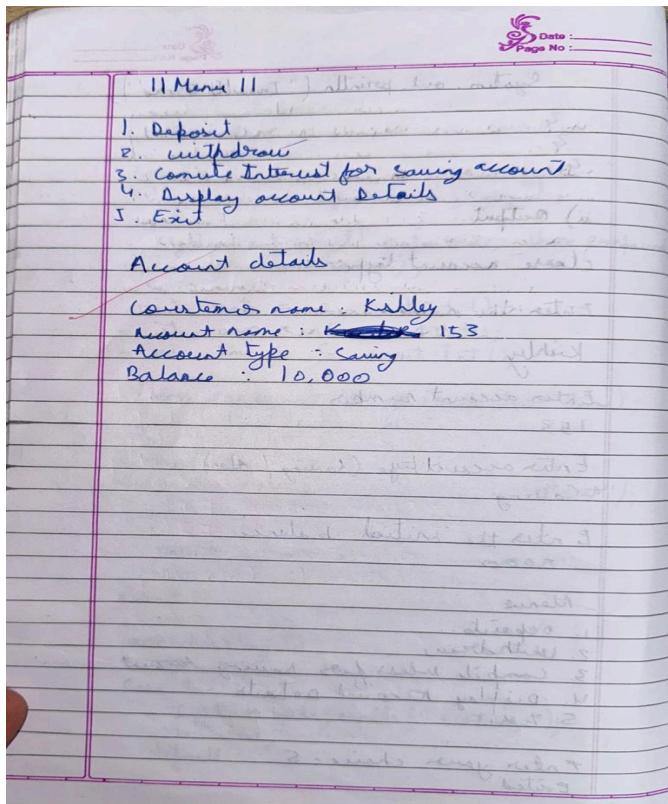
5000

Menu

1. Deposits
2. Withdrawal
3. Compute interest for savings account
4. Display account details
5. Exit

Enter your choice: 5

Exited



Code:

```
import java.util.Scanner;

class Account {

    String customerName;
    String accountNumber;
    double balance;
    String accountType;

    public Account(String customerName, String accountNumber, String accountType) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0.0;
    }
}
```

```

}

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

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

public double getBalance() {
    return balance;
}

class SavingsAccount extends Account {
    double interestRate;

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

    public void computeInterest() {
        double interest = balance * interestRate / 100;
        balance += interest;
        System.out.println("Interest computed and added: " + interest);
        displayBalance();
    }
}

```

```
}

public void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrew: " + amount);
        displayBalance();
    } else {
        System.out.println("Insufficient balance!");
    }
}
```

```
class CurrentAccount extends Account {
    double minimumBalance;
    double serviceCharge;

    public CurrentAccount(String customerName, String accountNumber, double minimumBalance,
        double serviceCharge) {
        super(customerName, accountNumber, "Current");
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }
}
```

```
public void withdraw(double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println("Withdrew: " + amount);
        displayBalance();
    } else {
```

```

        System.out.println("Insufficient balance!");
    }

    checkMinimumBalance();

}

public void checkMinimumBalance() {
    if (balance < minimumBalance) {
        balance -= serviceCharge;
        System.out.println("Service charge applied: " + serviceCharge);
        displayBalance();
    }
}

}

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

        System.out.println("Enter customer name:");
        String name = scanner.nextLine();

        System.out.println("Enter account number:");
        String accountNumber = scanner.nextLine();

        System.out.println("Choose account type (1 for Savings, 2 for Current):");
        int choice = scanner.nextInt();

        // Clear the newline character left by nextInt()
        scanner.nextLine();
    }
}

```

```

Account account = null;

if (choice == 1) {

    System.out.println("Enter interest rate:");
    double interestRate = scanner.nextDouble();
    scanner.nextLine(); // Clear the newline
    account = new SavingsAccount(name, accountNumber, interestRate);

} else if (choice == 2) {

    System.out.println("Enter minimum balance:");
    double minBalance = scanner.nextDouble();
    System.out.println("Enter service charge:");
    double serviceCharge = scanner.nextDouble();
    scanner.nextLine(); // Clear the newline
    account = new CurrentAccount(name, accountNumber, minBalance, serviceCharge);

} else {

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

while (true) {

    System.out.println("Choose an action (1 for Deposit, 2 for Withdraw, 3 for Display Balance, 4 for Compute Interest, 5 to Exit):");

    int action = scanner.nextInt();
    scanner.nextLine(); // Clear the newline

    switch (action) {

        case 1:

            System.out.println("Enter amount to deposit:");
            double depositAmount = scanner.nextDouble();
            scanner.nextLine(); // Clear the newline
            account.deposit(depositAmount);
    }
}

```

```

break;

case 2:
    System.out.println("Enter amount to withdraw:");
    double withdrawAmount = scanner.nextDouble();
    scanner.nextLine(); // Clear the newline
    if (account instanceof SavingsAccount) {
        ((SavingsAccount) account).withdraw(withdrawAmount);
    } else if (account instanceof CurrentAccount) {
        ((CurrentAccount) account).withdraw(withdrawAmount);
    }
    break;

case 3:
    account.displayBalance();
    break;

case 4:
    if (account instanceof SavingsAccount) {
        ((SavingsAccount) account).computeInterest();
    } else {
        System.out.println("Interest computation is not applicable for Current Account.");
    }
    break;

case 5:
    System.out.println("Exiting...");
    scanner.close(); // Close the scanner to release resources
    System.out.println("Name: Kishley Sharma"); // Added as per request
    System.out.println("USN: 1BM23CS153"); // Added as per request
    return;

default:
    System.out.println("Invalid action.");
}

```

```
    }  
}  
}
```

```
D:\Kishley Sharma>javac Bank.java  
  
D:\Kishley Sharma>java Bank  
Enter customer name:  
Kishley  
Enter account number:  
153  
Choose account type (1 for Savings, 2 for Current):  
1  
Enter interest rate:  
12  
Choose an action (1 for Deposit, 2 for Withdraw, 3 for Display Balance, 4 for Compute Interest, 5 to Exit):  
1  
Enter amount to deposit:  
5000  
Deposited: 5000.0  
Balance: 5000.0  
Choose an action (1 for Deposit, 2 for Withdraw, 3 for Display Balance, 4 for Compute Interest, 5 to Exit):  
2  
Enter amount to withdraw:  
2000  
Withdrew: 2000.0  
Balance: 3000.0  
Choose an action (1 for Deposit, 2 for Withdraw, 3 for Display Balance, 4 for Compute Interest, 5 to Exit):  
5  
Exiting...  
Name: Kishley Sharma  
USN: 1BM23CS153  
  
D:\Kishley Sharma>
```

Program 6

Packages

Algorithm:

Q6 Create a package CIE which has two classes Student and Internals. The class Student has members like name, vno, 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 a student in all five courses.

```
package CIE;
import java.util.Scanner;
class Student {
    int vno;
    String name;
    int sem;
    Scanner sc = new Scanner(System.in);
    public void getDetails (String n, int v, int s) {
        name = n;
        vno = v;
        sem = s;
    }
    public void display (Student s) {
        System.out.println("Student Name: " + name + " in Vno: "
                           + vno + " in Semester: " + sem);
    }
}
```

```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    public int [] imarks = new int [5];
    Scanner s = new Scanner (System.in);
    public void getimarks () {
        System.out.print ("Enter CIE marks for 5 subjects: ");
        for (int i = 0; i < 5; i++) {
            imarks[i] = s.nextInt ();
        }
    }
}
package SEE;
import CIE.internals;
import java.util.Scanner;
public class External extends Internals {
    int [] smarks = new int [5];
    int [] fmarks = new int [5];
    Scanner s = new Scanner (System.in);
    public void getmarks () {
        System.out.print ("Enter SEE marks for 5 subjects: ");
        for (int i = 0; i < 5; i++) {
            smarks[i] = s.nextInt ();
        }
    }
    public void calculate () {
        for (int i = 0; i < 5; i++) {
            fmarks[i] = (smarks[i] * 1) + imarks[i];
        }
    }
}
```

```

public void displaymarks () {
    System.out.println ("Final marks are : ");
    for (int i = 0; i < 5; i++)
        System.out.print (marks[i] + " ");
}

import java.util.*;
import java.io.*;
import java.util.Scanner;
public class final {
    public static void main (String args []) {
        int n;
        Scanner S1 = new Scanner (System.in);
        System.out.println ("Enter number of students : ");
        n = (Integer) int (1);
        External E1 = new External [n];
        for (int i = 0; i < n; i++) {
            E1[i] = new External ();
            System.out.println ("Enter name, usn, semester of student");
            String name = S1.nextLine();
            int usn = S1.nextInt();
            int sem = S1.nextInt();
            E1[i].getdetails (name, usn, sem);
            E1[i].displaystudent ();
            E1[i].getmarks ();
            E1[i].getpercentage ();
            E1[i].califinal ();
            E1[i].displaymarks ();
        }
    }
}

```

output

20 students info will be stored
Enter number of Students : 2

Enter name, USN, Semester of Student : Ram 101 3
Student name : Alice
USN : 101
Semester : 3

Enter CIE marks for 5 subjects
80 75 90 85 70

Enter SEE marks for 5 subjects
160 150 180 170 100

Final marks are 100 150 180 170 140

Enter name, USN, semester of student
Bob 102 3

Student name : Bob
semester : 3
USN : 102

Enter CIE marks for 5 students
95 50 50 45 50
" . . . SEE . . . 5 students
95 100 100 96 98

Final marks are
92 80 100 93 99.

Code:

```
import java.util.Scanner;

class Subject {

    int subjectMarks[] = new int[8];
    int credits[] = new int[8];
    int grades[] = new int[8];

    public void getMarksAndCredits(Scanner s) {
        System.out.println("Enter the subject Marks:");
        for (int j = 0; j < 8; j++) {
            System.out.print("Subject " + (j + 1) + ": ");
            subjectMarks[j] = s.nextInt();
        }

        System.out.println("Enter the subject Credits:");
        for (int j = 0; j < 8; j++) {
            System.out.print("Credits for Subject " + (j + 1) + ": ");
            credits[j] = s.nextInt();
        }
    }

    public void computeGrades() {
        for (int j = 0; j < 8; j++) {
            grades[j] = subjectMarks[j] / 10 + 1;
            if (grades[j] > 10) {
                grades[j] = 10;
            }
            if (grades[j] < 4) {
                grades[j] = 0;
            }
        }
    }
}
```

```

        }
    }
}

class Student {
    String name;
    String usn;
    double sgpa;
    Subject subject;

    public Student() {
        subject = new Subject();
    }

    public void getStudentDetails(Scanner s) {
        System.out.print("Enter the Name: ");
        name = s.nextLine();
        System.out.print("Enter the USN: ");
        usn = s.nextLine();
    }

    public void computeSGPA() {
        double totalCredits = 0;
        double totalPoints = 0;

        for (int j = 0; j < 8; j++) {
            totalCredits += subject.credits[j];
            totalPoints += subject.grades[j] * subject.credits[j];
        }
    }
}

```

```

sgpa = totalPoints / totalCredits;

}

public void display() {

    System.out.println("Name: " + name);
    System.out.println("USN: " + usn);
    System.out.println("SGPA: " + sgpa);
}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);
    Student[] students = new Student[3];
    System.out.println("Name: Kishley Sharma");
    System.out.println("USN: 1BM23CS153");

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

        System.out.println("Entering details for Student " + (i + 1) + ":");
        students[i] = new Student();
        students[i].getStudentDetails(sc);
        students[i].subject.getMarksAndCredits(sc);
        students[i].subject.computeGrades();
        students[i].computeSGPA();
        students[i].display();
        System.out.println();
    }
}

```

```
D:\Kishley Sharma>javac Student.java

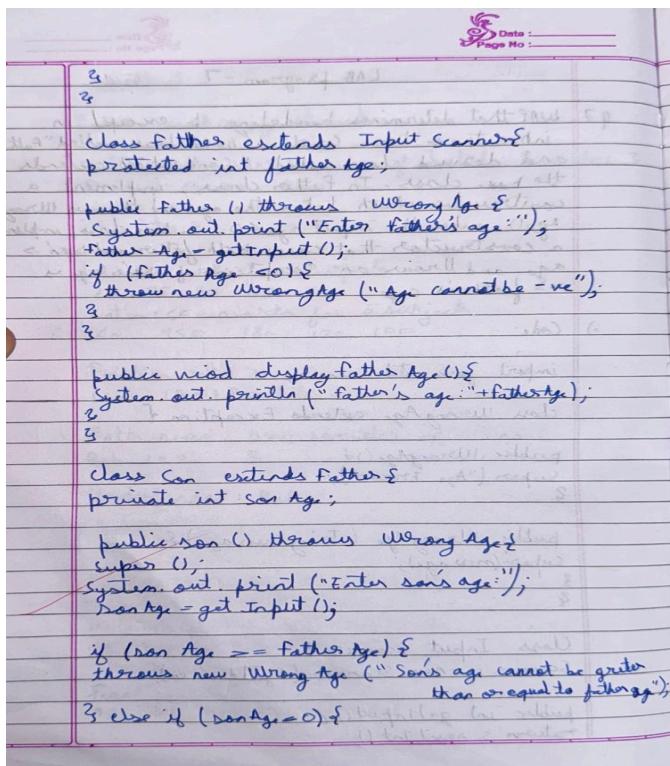
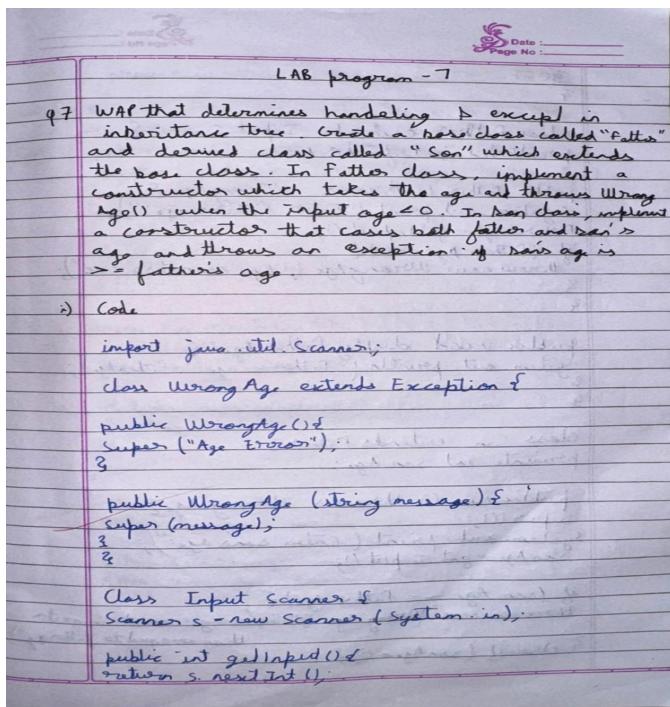
D:\Kishley Sharma>java Student
Name: Kishley Sharma
USN: 1BM23CS153
Entering details for Student 1:
Enter the Name: Kishley
Enter the USN: 153
Enter the subject Marks:
Subject 1: 45
Subject 2: 234
Subject 3: 98
Subject 4: 55
Subject 5: 78
Subject 6: 84
Subject 7: 92
Subject 8: 21
Enter the subject Credits:
Credits for Subject 1: 3
Credits for Subject 2: 4
Credits for Subject 3: 2
Credits for Subject 4: 1
Credits for Subject 5: 0
Credits for Subject 6: 2
Credits for Subject 7: 1
Credits for Subject 8: 1
Name: Kishley
USN: 153
SGPA: 7.785714285714286

Entering details for Student 2:
Enter the Name: Enter the USN: 159
Enter the subject Marks:
Subject 1: 67
Subject 2:
67
Subject 3: 87
Subject 4: 45
Subject 5: 23
Subject 6: 45
Subject 7: 65
Subject 8: 76
Enter the subject Credits:
Credits for Subject 1: 3
Credits for Subject 2: 4
Credits for Subject 3: 3
Credits for Subject 4: 3
Credits for Subject 5: 2
Credits for Subject 6: 2
Credits for Subject 7: 1
Credits for Subject 8: 1
```

Program 7

Exceptions

Algorithm:



```

Date : _____
Page No. : _____
1. Now new wrong Age ("Age cannot be -ve");
2.
3. If son's age is greater than father's age, then throw
   public void display Son Age() {
       System.out.println("Son age : " + sonAge);
   }
2.
3. If son's age is less than or equal to father's age
public class Main {
    public static void main (String [] args) {
        try {
            Son son = new Son ();
            son.display Father ();
            Son.display SonAge ();
        } catch (WrongAge e) {
            System.out.println ("Error : " + e.getMessage ());
        }
    }
}
2.
2.

Output
Enter father's age: 41
Enter son's age: 60
- Son's age cannot be greater than father's age

```

Code:

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

```
class WrongAge extends Exception {
```

```
    public WrongAge() {
        super("Age Error");
    }
}
```

```
    public WrongAge(String message) {
        super(message);
    }
}
```

```
class InputScanner {  
    Scanner s = new Scanner(System.in);  
  
    public int getInput() {  
        return s.nextInt();  
    }  
}  
  
class Father extends InputScanner {  
    protected int fatherAge;  
  
    public Father() throws WrongAge {  
        System.out.print("Enter father's age: ");  
        fatherAge = getInput();  
        if (fatherAge < 0) {  
            throw new WrongAge("Age cannot be negative");  
        }  
    }  
  
    public void displayFatherAge() {  
        System.out.println("Father's age: " + fatherAge);  
    }  
}  
  
class Son extends Father {  
    private int sonAge;
```

```

public Son() throws WrongAge {
    super();
    System.out.print("Enter son's age: ");
    sonAge = getInput();

    if (sonAge >= fatherAge) {
        throw new WrongAge("Son's age cannot be greater than or equal to father's age");
    } else if (sonAge < 0) {
        throw new WrongAge("Age cannot be negative");
    }
}

public void displaySonAge() {
    System.out.println("Son's age: " + sonAge);
}

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

```

```

System.out.println("Name: Kishley Sharma");
System.out.println("USN: 1BM23CS153");

}

}

```

```

D:\Kishley Sharma>javac Main.java

D:\Kishley Sharma>java Main
Enter father's age: 41
Enter son's age: 60
Error: Son's age cannot be greater than or equal to father's age
Name: Kishley Sharma
USN: 1BM23CS153

```

Program 8

Threads

Algorithm:

LAB 8

Q → WAP which creates two threads, one thread displaying "BMS college of Engg" once every 10 seconds and another "CSE" once every 2 seconds.

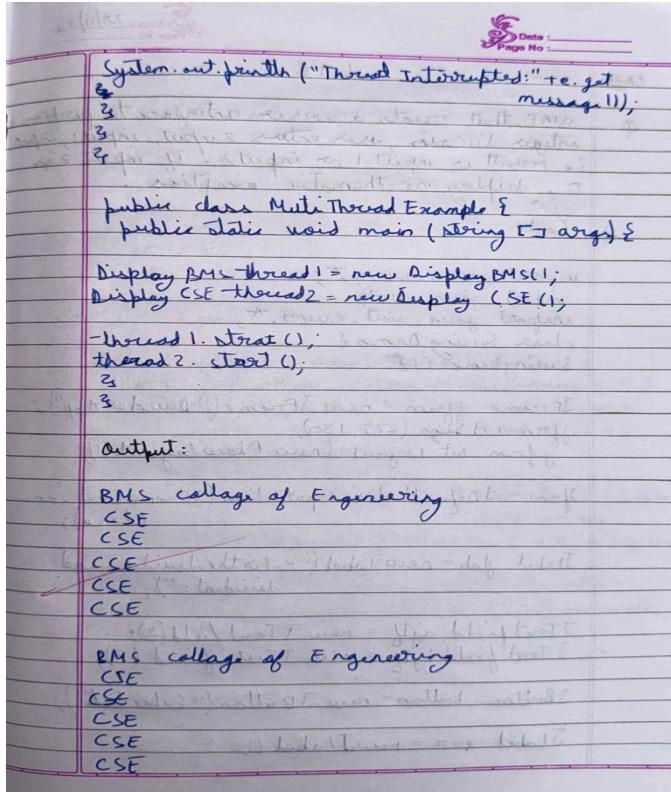
i) "code"

```

class DisplayBMS extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println(" BMS college of Engineering ");
                Thread.sleep(10000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted:" + e.getMessage());
        }
    }
}

class DisplayCSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println(" CSE ");
                Thread.sleep(1000);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted:" + e.getMessage());
        }
    }
}

```



Code:

```
class DisplayBMS extends Thread {  
    public void run() {  
        try {  
            while (true) {  
                System.out.println("BMS College of Engineering");  
                Thread.sleep(10000); // Sleep for 10 seconds  
            }  
        } catch (InterruptedException e) {  
            System.out.println("Thread interrupted: " + e.getMessage());  
        }  
    }  
}
```

```

class DisplayCSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(1000); // Sleep for 1 second
            }
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted: " + e.getMessage());
        }
    }
}

```

```

public class MultiThreadExample {
    public static void main(String[] args) {
        // Creating instances of the threads
        DisplayBMS thread1 = new DisplayBMS();
        DisplayCSE thread2 = new DisplayCSE();

        // Starting the threads
        thread1.start();
        thread2.start();

        // Adding a way to stop the threads after some time (e.g., 20 seconds)
        try {
            Thread.sleep(20000); // Let the threads run for 20 seconds
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}

```

```

// Interrupt both threads to stop them gracefully
thread1.interrupt();
thread2.interrupt();

try {
    // Wait for both threads to finish
    thread1.join();
    thread2.join();
} catch (InterruptedException e) {
    e.printStackTrace();
}

System.out.println("Name: Kishley Sharma");
System.out.println("USN: 1BM23CS153");
}
}

```

```

D:\Kishley Sharma>javac MultiThreadExample.java
D:\Kishley Sharma>java MultiThreadExample
BMS College of Engineering
CSE
BMS College of Engineering
CSE
Thread interrupted: sleep interrupted
Thread interrupted: sleep interrupted
Name: Kishley Sharma
USN: 1BM23CS153
D:\Kishley Sharma>

```

Program 9

Interface

Algorithm:

L 1 B 9

Q WAP that creates a user interface to perform integer division over enter 2 inputs, input1, input2.
1. If input1 or input2 is 0, display arithmetic exception.
2. If input2 is 0, display arithmetic exception.

i) Code

```
import javax.swing.*; import java.awt.*;  
import java.awt.event.*;  
class SwingDemo extends JFrame  
SwingDemo() {  
    JTextField atextField = new JTextField("Enter the dividend");  
    JTextField btextField = new JTextField("Enter the divisor");  
    JButton calculateButton = new JButton("Calculate");  
    JLabel resultLabel = new JLabel("Result");  
    JPanel panel = new JPanel();  
    panel.setLayout(new GridLayout(3, 1));  
    panel.add(atextField);  
    panel.add(btextField);  
    panel.add(calculateButton);  
    panel.add(resultLabel);  
    setLayout(panel);  
    calculateButton.addActionListener(new ActionListener() {  
        public void actionPerformed(ActionEvent e) {  
            int a = Integer.parseInt(atextField.getText());  
            int b = Integer.parseInt(btextField.getText());  
            if (b == 0) {  
                JOptionPane.showMessageDialog(null, "Divisor cannot be zero");  
            } else {  
                resultLabel.setText("Result: " + (a / b));  
            }  
        }  
    });  
    setSize(300, 200);  
    setVisible(true);  
}  
public static void main(String[] args) {  
    SwingDemo demo = new SwingDemo();  
}
```

JLabel alab = new JLabel("Enter the dividend");
JLabel blab = new JLabel("Enter the divisor");
JLabel anslab = new JLabel("Result");

jform.add(alab);
jform.add(blab);
jform.add(atextf);
jform.add(btextf);
jform.add(button);
jform.add(alab);
jform.add(blab);
jform.add(anslab);

actionListener1 = new ActionListener() {
 public void actionPerformed(ActionEvent e) {
 System.out.println("Action event from a text field");
 atextf.addActionListener(e);
 btextf.addActionListener(e);
 button.addActionListener(e);
 }
};
button.addActionListener(e);

void actionPerformed(ActionEvent e) {
 try {
 Integer.parseInt(atextf.getText());
 Integer.parseInt(btextf.getText());
 int a = Integer.parseInt(atextf.getText());
 int b = Integer.parseInt(btextf.getText());
 int ans = a / b;
 alab.setText("Result: " + a);
 blab.setText("Result: " + b);
 anslab.setText("Result: " + ans);
 } catch (NumberFormatException e) {
 alab.setText("Result: " + e);
 }
}

Date : _____
 Page No. : _____

```

        blab.settext("");
        anslab.setText("");
        err.setText("Enter Only Integers!");  

        catch (ArithmaticException e){}
        alab.setText("");
        blab.setText("");
        anslab.setText("");
        err.setText("E should be NON zero!");  

        }
    
```

₂ form.setVisible(true);

```

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

₃ ~~int a = 10; int b = 5; int c = a/b;~~
~~System.out.println("a/b = " + c);~~
~~a = 10; b = 5; c = a/b;~~
~~System.out.println("a/b = " + c);~~
~~return 0;~~

₃ ~~int a = 10; int b = 5; int c = a/b;~~
~~System.out.println("a/b = " + c);~~
~~a = 10; b = 5; c = a/b;~~
~~System.out.println("a/b = " + c);~~
~~return 0;~~

₃ ~~int a = 10; int b = 5; int c = a/b;~~
~~System.out.println("a/b = " + c);~~
~~a = 10; b = 5; c = a/b;~~
~~System.out.println("a/b = " + c);~~
~~return 0;~~

ii) output

Divide 52 by 2	= 26
Enter the divisor and dividend:	
52	2
<u>calculate</u>	
Result: 26	

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(300, 200);
        jfrm.setLayout(new GridLayout(5, 1, 5, 5)); // GridLayout with spacing
        jfrm.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE); // Prevent default close
        action

        // Create labels and text fields
        JLabel jlab = new JLabel("Enter the divisor and dividend:");
        JTextField ajtf = new JTextField(10);
        JTextField bjtf = new JTextField(10);

        JButton button = new JButton("Calculate");
        JLabel err = new JLabel("", SwingConstants.CENTER);
        err.setForeground(Color.RED); // Highlight error messages in red

        JLabel anslab = new JLabel("", SwingConstants.CENTER);

        // Add components to the frame
        jfrm.add(jlab);
        JPanel inputPanel = new JPanel(new FlowLayout());
        inputPanel.add(ajtf);
        inputPanel.add(bjtf);
```

```

jfrm.add(inputPanel);
jfrm.add(button);
jfrm.add(err);
jfrm.add(anslab);

// Add ActionListener to the button
button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        try {
            err.setText(""); // Clear previous errors
            anslab.setText(""); // Clear previous results

            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());

            if (b == 0) {
                throw new ArithmeticException("Divisor cannot be zero");
            }

            int ans = a / b;
            anslab.setText("Result: " + ans);
        } catch (NumberFormatException e) {
            err.setText("Enter valid integers!");
        } catch (ArithmeticException e) {
            err.setText("Divisor cannot be zero!");
        }
    }
});

// Add a WindowListener to handle when the window is closing

```

```

jfrm.addWindowListener(new WindowAdapter() {
    @Override
    public void windowClosing(WindowEvent e) {
        System.out.println("Name: Kishley Sharma");
        System.out.println("USN: 1BM23CS153");
        System.exit(0); // Ensure the application terminates after printing the details
    }
});

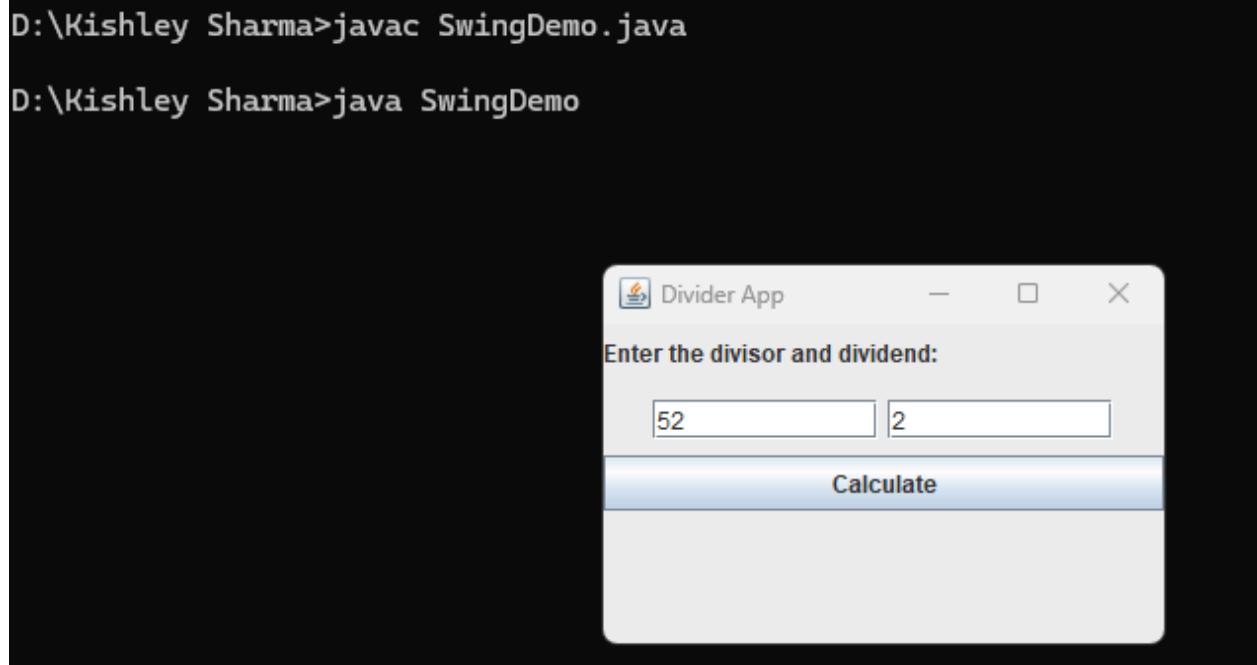
// Display frame
jfrm.setVisible(true);
}

public static void main(String[] args) {
    // Create frame on event dispatching thread
    SwingUtilities.invokeLater(() -> new SwingDemo());
}
}

```

D:\Kishley Sharma>javac SwingDemo.java

D:\Kishley Sharma>java SwingDemo



Program 10 A

PCfixed

Algorithm:

Lab program : 10 (a) [100 marks]

(a) Demonstrate Interprocess Communication & Deadlock

```
class Q {
    int n;
    boolean values = false;
    synchronized int get() {
        while (!valueSet)
            try { } catch (InterruptedException e) {
                System.out.println("Exception Caught");
                System.out.println("Get", + n);
                valueSet = true;
                System.out.println("In Internal Producer");
                notify();
                return n;
            }
        synchronized void put(int n) {
            while (!valueSet)
                try { System.out.println("In Producer Waiting"); } catch (InterruptedException e) {
                    System.out.println("Caught");
                    this.n = n;
                    valueSet = true;
                    System.out.println("In Internal Consumer");
                    notify();
                }
            class P implements Runnable {
                public void run() {
                    ...
                }
            }
        }
    }
}
```

Producer (q) :

```
This q = q;
new Thread(this, "Producer").start(); 3
public void run() {
    int i = 0;
    while (i < 15) {
        q.put(i++); 3 3
    }
}
class Consumer implements Runnable {
    public void run() {
        int i = 0;
        while (i < 15) {
            int n = q.get();
            i++;
        }
    }
}
class P implements Runnable {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
    }
}
```

Output :

Put: 1	Put: 5
get: 1	get: 5
Put: 2	Put: 6
get: 2	get: 6
Put: 3	Put: 7
get: 3	get: 7
Put: 4	Put: 8
get: 4	get: 8

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++;
    }
}

public class PCFixed {
    public static void main(String args[]) {
        Q q = new Q();
        Producer producer = new Producer(q);
        Consumer consumer = new Consumer(q);

        // Wait for the producer and consumer threads to finish
        try {
            Thread.sleep(1000); // Give threads some time to start and produce/consume
            producer.run(); // Directly run producer thread logic
            consumer.run(); // Directly run consumer thread logic
        } catch (InterruptedException e) {
            System.out.println("Thread interrupted");
        }

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

```

```
        System.out.println("Name: Kishley Sharma");
        System.out.println("USN: 1BM23CS153");
    }
}
```

```
D:\Kishley Sharma>javac PCFixed.java
D:\Kishley Sharma>java PCFixed
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

Producer waiting

Got: 3
```

Program 10 B

Deadlock

Algorithm:

```
b) 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.bar()");
        b.bar();
        wait();
        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.foo()");
        a.foo();
    }
}

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

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock () {
}
```

```
Thread = Current Thread();
Thread t = new Thread(this, "Racing Thread");
t.start();
a.foo(b);
System.out.println("Back in main thread");
public void run () {
    b.bar(a);
    System.out.println("Back in other thread");
}
public static void main (String args[]) {
    new Deadlock();
}

Output : Main thread entered A.foo
Racing Thread entered B.bar
Main Thread trying to call b.last()
Inside A.last
Back in main thread
Racing Thread trying to call a.foo()
Inside A.last
Back in other thread.
```

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 B.last");
}

class Deadlock implements Runnable {

    A a = new A();
    B b = new B();

    Deadlock() {
        Thread.currentThread().setName("MainThread");

        // Create a new thread and start it
        Thread t = new Thread(this, "RacingThread");
        t.start();
    }

    try {

```

```

// Main thread locks object A
a.foo(b);

System.out.println("Back in main thread");
t.join(); // Ensure main thread waits for the other thread to finish
} catch (InterruptedException e) {
    System.out.println("MainThread Interrupted");
}

// Print the name and USN after the program is finished
System.out.println("Name: Kishley Sharma");
System.out.println("USN: 1BM23CS153");
}

public void run() {
    // Other thread locks object B
    b.bar(a);
    System.out.println("Back in other thread");
}

public static void main(String args[]) {
    new Deadlock();
}

```

```

D:\Kishley Sharma>javac Deadlock.java
D:\Kishley Sharma>java Deadlock
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 other thread
Inside B.last
Back in main thread
Name: Kishley Sharma
USN: 1BM23CS153

D:\Kishley Sharma>

```