VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Kavya Singh (1BM23CS146)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

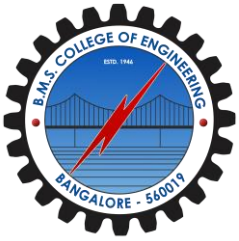
B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Sep-2024 to Jan-2025

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 **Kavya Singh (1BM23CS146),** 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.

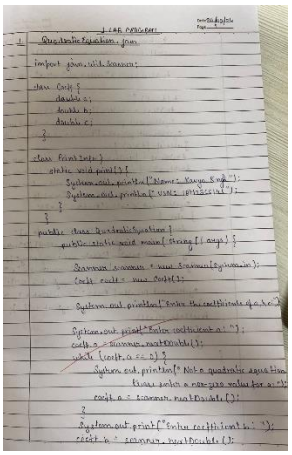
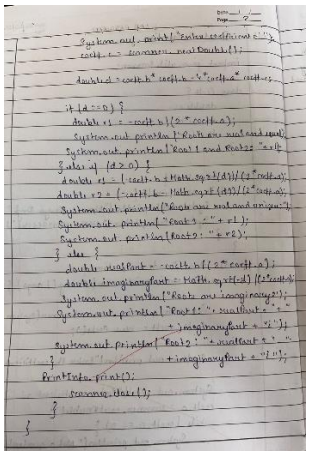
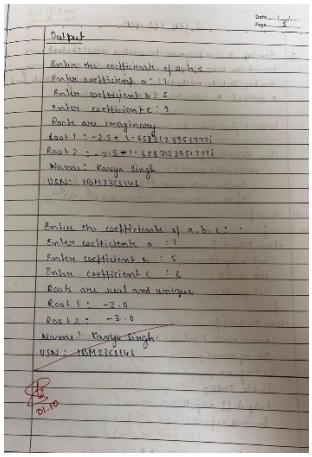
Lab faculty Incharge Name Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link: https://github.com/kavyasingh03/ooj-lab-programs

Program 1
Quadratic Equation

Algorithm:

Code:

```
import java.util.Scanner;

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

class PrintInfo {
  static void print() {
      System.out.println("Name: Kavya Singh");
      System.out.println("USN: 1BM23CS146");
   }
}

public class QuadraticEquation {
   public static void main(String[] args) {

      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");
        }
     PrintInfo.print();
        scanner.close();
    }
}
```

Output:

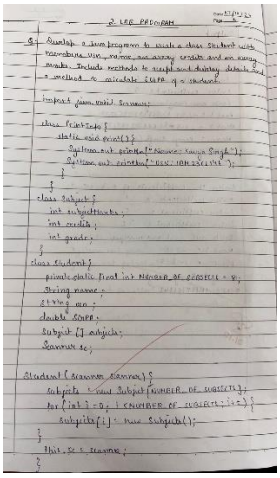
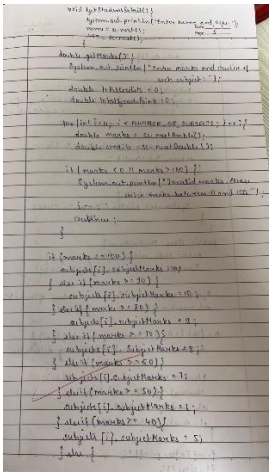
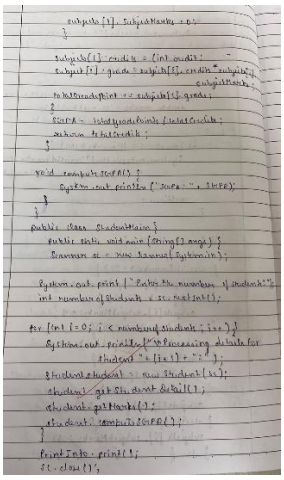
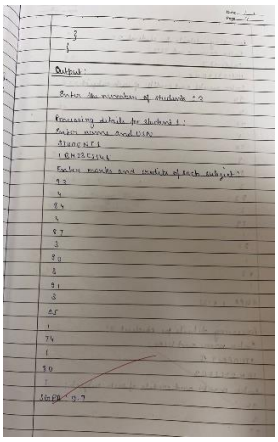
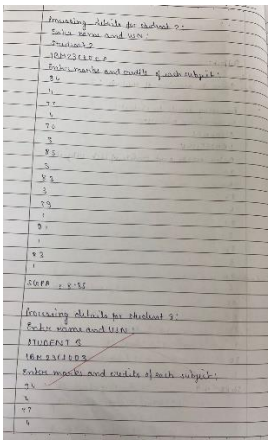
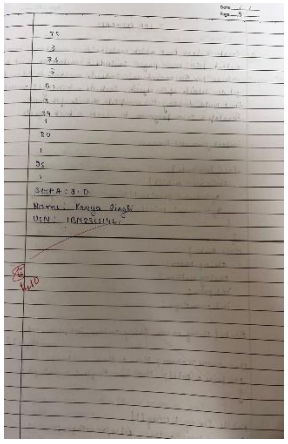
```
D:\1BM23CS146>java QuadraticEquation
Enter the coefficients of a, b, c:
Enter coefficient a: 1
Enter coefficient b: 5
Enter coefficient c: 9
Roots are imaginary.
Root 1: -2.5 + 1.65831239517777i
Root 2: -2.5 - 1.65831239517777i
Name: Kavya Singh
USN: 1BM23CS146
```

```
D:\1BM23CS146>java QuadraticEquation
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
Name: Kavya Singh
USN: 1BM23CS146

D:\1BM23CS146>
```

Program 2

Student SGPA

Algorithm:

Code:

```
import java.util.Scanner;

class PrintInfo {
   static void print() {
      System.out.println("Name: Kavya Singh");
      System.out.println("USN: 1BM23CS146");
   }
}
```

```
class Subject {
   int subjectMarks;
   int credits;
   int grade;
}

class Student {
   private static final int NUMBER_OF_SUBJECTS = 8;
   String name;
   String usn;
   double SGPA;
   Subject[] subjects;
   Scanner sc;

   Student(Scanner scanner) {
      subjects = new Subject[NUMBER_OF_SUBJECTS];
      for (int i = 0; i < NUMBER_OF_SUBJECTS; i++) {
         subjects[i] = new Subject();
      }
      this.sc = scanner;
   }

   void getStudentDetail() {
      System.out.println("Enter name and USN:");
      name = sc.next();
      usn = sc.next();
   }

   double getMarks() {
      System.out.println("Enter marks and credits of each subject:");
      double totalCredits = 0;
      double totalGradePoints = 0;

      for (int i = 0; i < NUMBER_OF_SUBJECTS; i++) {
         double marks = sc.nextDouble();
         double credits = sc.nextDouble();

         if (marks < 0 || marks > 100) {
            System.out.println("Invalid marks. Please enter marks between 0 and 100.");
            i--;
            continue;
         }

         if (marks == 100) {
```

```
            subjects[i].subjectMarks = 10;
        } else if (marks >= 90) {
            subjects[i].subjectMarks = 10;
        } else if (marks >= 80) {
            subjects[i].subjectMarks = 9;
        } else if (marks >= 70) {
            subjects[i].subjectMarks = 8;
        } else if (marks >= 60) {
            subjects[i].subjectMarks = 7;
        } else if (marks >= 50) {
            subjects[i].subjectMarks = 6;
        } else if (marks >= 40) {
            subjects[i].subjectMarks = 5;
        } else {
            subjects[i].subjectMarks = 0;
        }

        subjects[i].credits = (int) credits;
        subjects[i].grade = subjects[i].credits * subjects[i].subjectMarks;
        totalCredits += subjects[i].credits;
        totalGradePoints += subjects[i].grade;
      }

      SGPA = totalGradePoints / totalCredits;
      return totalCredits;
    }

    void computeSGPA() {
      System.out.println("SGPA: " + SGPA);
    }
}

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

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

      for (int i = 0; i < numberOfStudents; i++) {
        System.out.println("\nProcessing details for student " + (i + 1) + ":");
        Student student = new Student(sc);
        student.getStudentDetail();
        student.getMarks();
```

```
            student.computeSGPA();
        }
        PrintInfo.print();

        sc.close();
    }
}
```

Output:

```
D:\1BM23CS146>javac StudentMain.java

D:\1BM23CS146>java StudentMain
Enter the number of students: 3

Processing details for student 1:
Enter name and USN:
STUDENT1
1BM23CS145
Enter marks and credits of each subject:
93
4
94
4
87
3
90
3
91
3
95
1
74
1
80
1
SGPA: 9.7
```

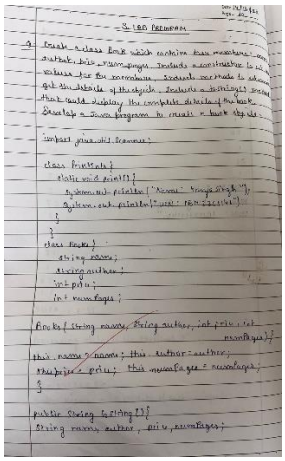
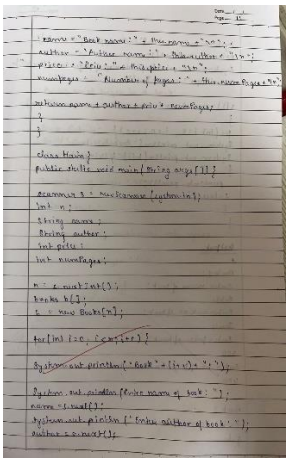
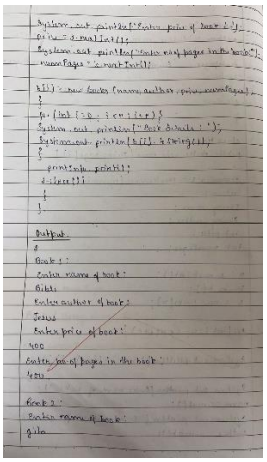
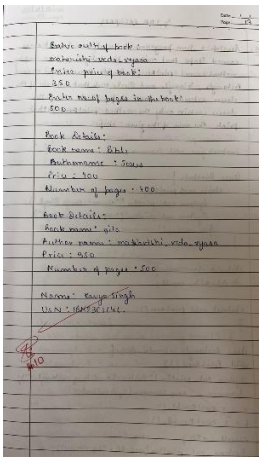
```
Processing details for student 2:
Enter name and USN:
STUDENT2
1BM23CS002
Enter marks and credits of each subject:
94
4
77
4
70
3
85
3
83
3
79
1
```

```
91
1
83
1
SGPA: 8.85

Processing details for student 3:
Enter name and USN:
STUDENT3
1BM23CS003
Enter marks and credits of each subject:
94

4
77
4
70
3
85
3
91
3
74
1
80
1
95
1
SGPA: 9.0
Name: Kavya Singh
USN: 1BM23CS146
```

Program 3

Book Details

Algorithm:

Code:

```
import java.util.Scanner;

class PrintInfo {
    static void print() {
        System.out.println("Name: Kavya Singh");
        System.out.println("USN: 1BM23CS146");
    }
}

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[]){

Scanner s = new Scanner(System.in);
int n;
String name;
String author;
int price;
int numPages;

n = s.nextInt(); //read no. of books
Books b[];
b = new Books[n];

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

System.out.println("Book "+(i+1)+":");

System.out.println("Enter name of book: ");
name = s.next();
System.out.println("Enter author of book: ");
author = s.next();
System.out.println("Enter price of book: ");
price = s.nextInt();
System.out.println("Enter no of pages in the book: ");
numPages = s.nextInt();

b[i] = new Books(name,author,price,numPages);
}

for(int i=0;i<n;i++){
System.out.println("Book Details: ");
System.out.println(b[i].toString());
```

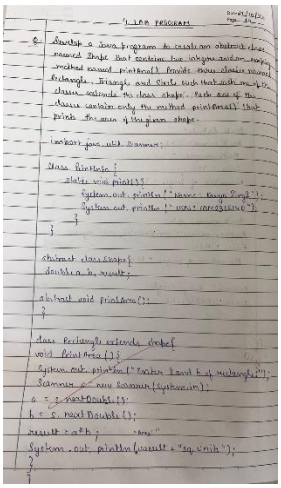
```
}
 PrintInfo.print();
s.close();
}
}
```

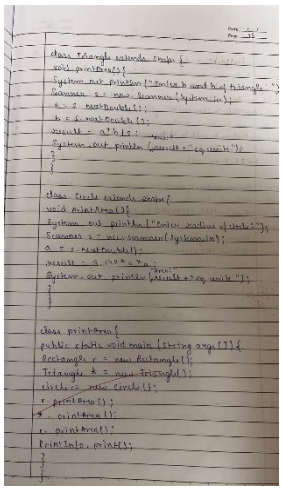
Output:

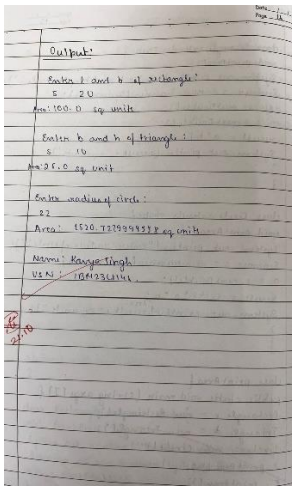
Program 4

Area of the Shape

Algorithm:







Code:

```
import java.util.Scanner;

class PrintInfo {
  static void print() {
      System.out.println("Name: Kavya Singh");
      System.out.println("USN: 1BM23CS146");
   }
}

abstract class Shape{
double a,b,result;

abstract void printArea();
}

class Rectangle extends Shape{
void printArea(){
System.out.println("Enter l and b of rectangle:");
Scanner s=new Scanner(System.in);
a=s.nextDouble();
b=s.nextDouble();
```

```
result=a*b;
System.out.println(result+" sq units");
}
}
class Triangle extends Shape{
void printArea(){
System.out.println("Enter b and h of triangle:");
Scanner s=new Scanner(System.in);
a=s.nextDouble();
b=s.nextDouble();
result=a*b/2;
System.out.println(result+" sq units");
}
}
class Circle extends Shape{
void printArea(){
System.out.println("Enter radius of circle:");
Scanner s=new Scanner(System.in);
a=s.nextDouble();
result=3.142*a*a;
System.out.println(result+" sq units");
}
}

class printArea{
public static void main(String args[]){
Rectangle r=new Rectangle();
Triangle t=new Triangle();
Circle c=new Circle();
r.printArea();
t.printArea();
c.printArea();
 PrintInfo.print();
}
}
```

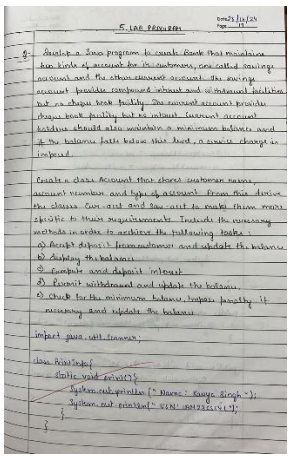
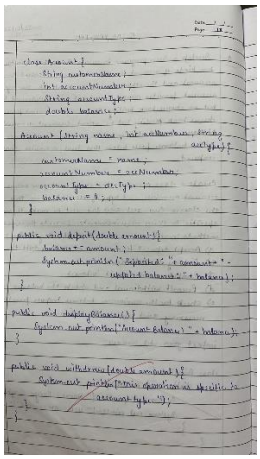
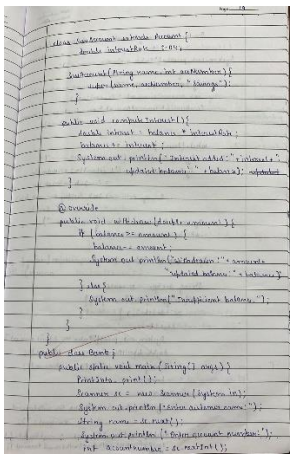
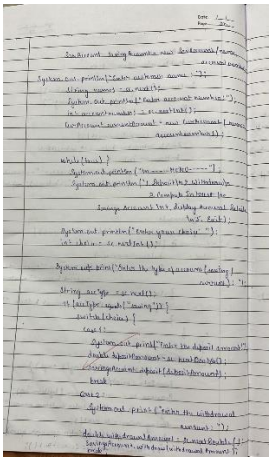
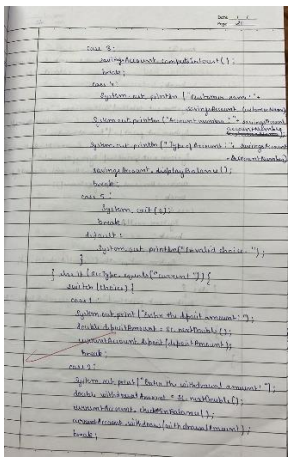
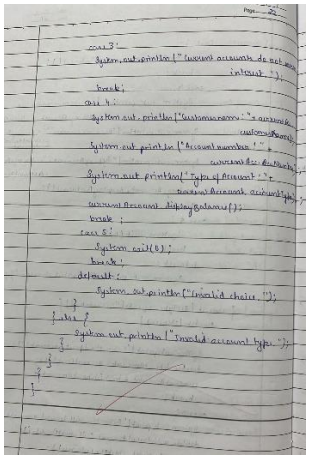
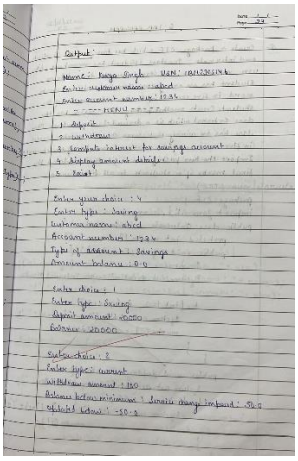
Output:

```
D:\1BM23CS146>javac printArea.java

D:\1BM23CS146>java printArea
Enter l and b of rectangle:
5 20
100.0 sq units
Enter b and h of triangle:
5 10
25.0 sq units
Enter radius of circle:
22
1520.7279999999998 sq units
Name: Kavya Singh
USN: 1BM23CS146

D:\1BM23CS146>
```

Program 5

Bank

Algorithm:

Code:

```
import java.util.Scanner;

class PrintInfo {
  static void print() {
      System.out.println("Name: Kavya Singh");
      System.out.println("USN: 1BM23CS146");
   }
}
```

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


    Account(String name, int accNumber, String accType) {
        customerName = name;
        accountNumber = accNumber;
        accountType = accType;
        balance = 0;
    }


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

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


    public void withdraw(double amount) {
        System.out.println("This operation is specific to account type.");
    }
}


class SavAccount extends Account {
    double interestRate = 0.04;  // 4% annual interest rate

    SavAccount(String name, int accNumber) {
        super(name, accNumber, "Savings");
    }


    public void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest added: " + interest + ". Updated balance: " + balance);
```

```
    }



    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount + ". Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}
class CurAccount extends Account {
    double minBalance = 500.0;
    double serviceCharge = 50.0;

    CurAccount(String name, int accNumber) {
        super(name, accNumber, "Current");
    }

    public void checkMinBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Balance below minimum. Service charge imposed: " + serviceCharge
+ ". Updated balance: " + balance);
        }
    }



    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount + ". Updated balance: " + balance);
            checkMinBalance();
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

public class Bank {
```

```
    public static void main(String[] args) {
        PrintInfo.print();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter customer name:");
        String name=sc.next();
        System.out.println("Enter account number:");
        int accountnumber=sc.nextInt();
        SavAccount savingsAccount = new SavAccount(name, accountnumber);
System.out.println("Enter customer name:");
        String name1=sc.next();
        System.out.println("Enter account number:");
        int accountnumber1=sc.nextInt();
        CurAccount currentAccount = new CurAccount(name1, accountnumber1);

        while (true) {
            System.out.println("\n-----MENU-----");
            System.out.println("1. Deposit\n2. Withdraw\n3. Compute Interest for Savings
Account\n4. Display Account Details\n5. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();

            System.out.print("Enter the type of account (saving/current): ");
            String accType = sc.next();

            if (accType.equals("saving")) {
                switch (choice) {
                    case 1:
                        System.out.print("Enter the deposit amount: ");
                        double depositAmount = sc.nextDouble();
                        savingsAccount.deposit(depositAmount);
                        break;
                    case 2:
                        System.out.print("Enter the withdrawal amount: ");
                        double withdrawalAmount = sc.nextDouble();
                        savingsAccount.withdraw(withdrawalAmount);
                        break;
                    case 3:
                        savingsAccount.computeInterest();
                        break;
                    case 4:
                        System.out.println("Customer name: " + savingsAccount.customerName);
                        System.out.println("Account number: " + savingsAccount.accountNumber);
                        System.out.println("Type of Account: " + savingsAccount.accountType);
```

```
                savingsAccount.displayBalance();
                break;
            case 5:
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice.");
        }
    } else if (accType.equals("current")) {
        switch (choice) {
            case 1:
                System.out.print("Enter the deposit amount: ");
                double depositAmount = sc.nextDouble();
                currentAccount.deposit(depositAmount);
                break;
            case 2:
                System.out.print("Enter the withdrawal amount: ");
                double withdrawalAmount = sc.nextDouble();
                currentAccount.checkMinBalance();
                currentAccount.withdraw(withdrawalAmount);
                break;
            case 3:
                System.out.println("Current accounts do not earn interest.");
                break;
            case 4:
                System.out.println("Customer name: " + currentAccount.customerName);
                System.out.println("Account number: " + currentAccount.accountNumber);
                System.out.println("Type of Account: " + currentAccount.accountType);
                currentAccount.displayBalance();
                break;
            case 5:
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice.");
        }
    } else {
        System.out.println("Invalid account type.");
    }
  }
}
```

Output:

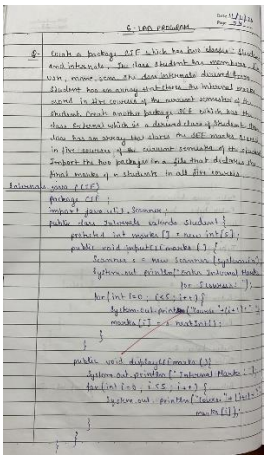
```
D:\1BM23CS146>javac Bank.java

D:\1BM23CS146>java Bank
Name: Kavya Singh
USN: 1BM23CS146
Enter customer name:
abcd
Enter account number:
1234
Enter customer name:
ghfd
Enter account number:
1234

-----MENU-----
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 1
Enter the type of account (saving/current): saving
Enter the deposit amount: 2300
Deposited: 2300.0. Updated balance: 2300.0

-----MENU-----
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 2
Enter the type of account (saving/current): saving
Enter the withdrawal amount: 300
Withdrawn: 300.0. Updated balance: 2000.0

-----MENU-----
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 4
Enter the type of account (saving/current): saving
Customer name: abcd
Account number: 1234
Type of Account: Savings
Account Balance: 2000.0
```

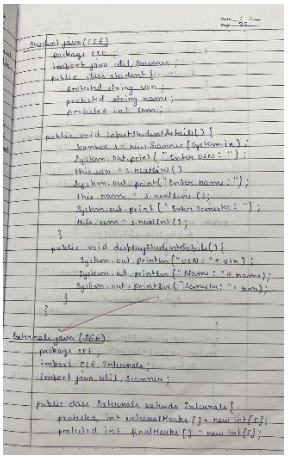
```
-----MENU-----
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 5
Enter the type of account (saving/current): savings
Invalid account type.

-----MENU-----
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 2
Enter the type of account (saving/current): current
Enter the withdrawal amount: 500
Balance below minimum. Service charge imposed: 50.0. Updated balance: -50.0
Insufficient balance.
```

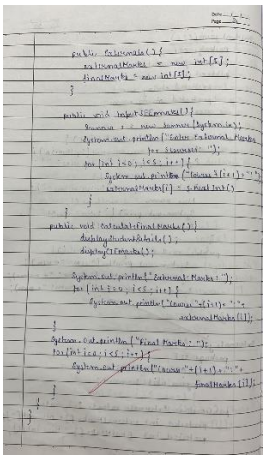
Program 6

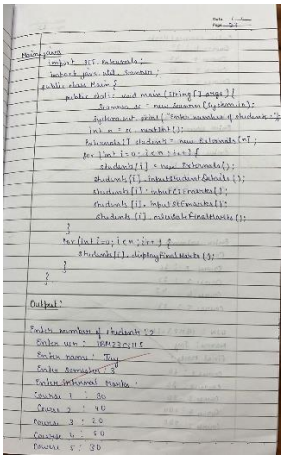
Package

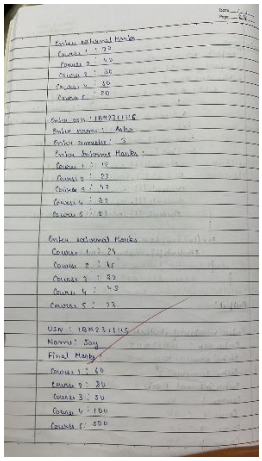
Algorithm:

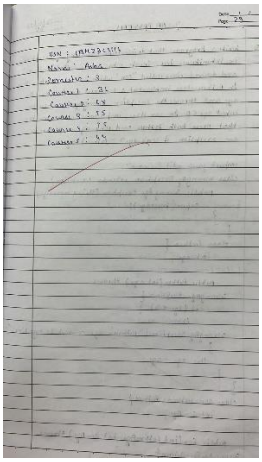












Code: CIE

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:

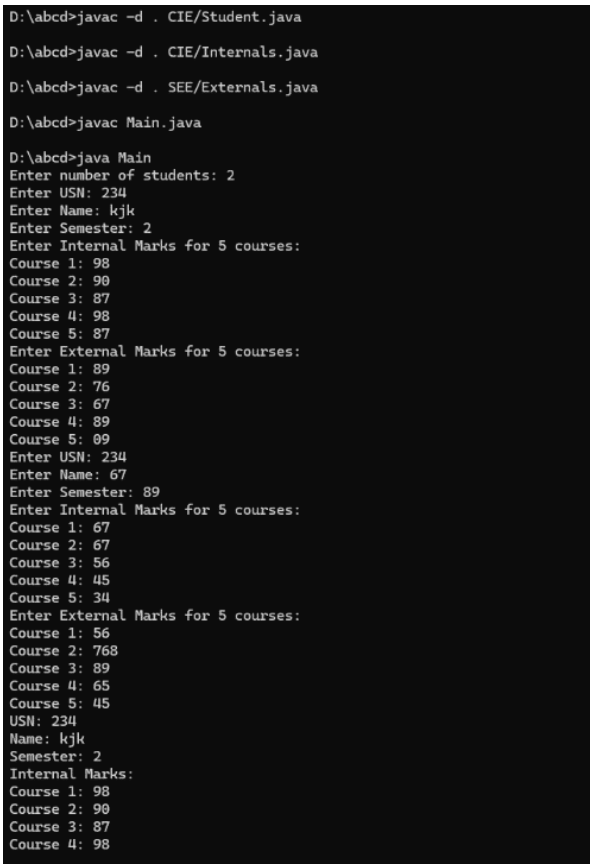
Student.java

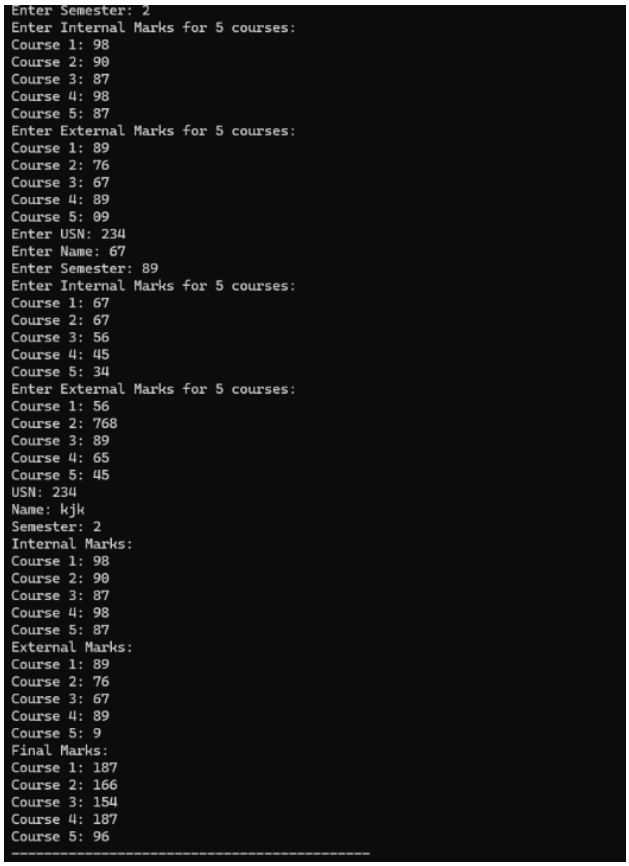
```
package SEE;

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:

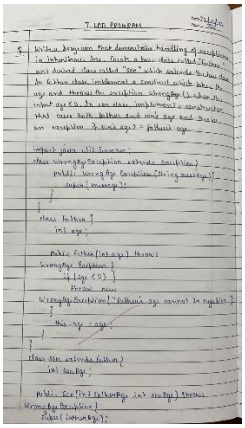

D:\abcd>javac -d . CIE/Student.java
D:\abcd>javac -d . CIE/Internals.java
D:\abcd>javac -d . SEE/Externals.java
D:\abcd>javac Main.java
D:\abcd>java Main
Enter number of students: 2
Enter USN: 234
Enter Name: kjk
Enter Semester: 2
Enter Internal Marks for 5 courses:
Course 1: 98
Course 2: 90
Course 3: 87
Course 4: 98
Course 5: 87
Enter External Marks for 5 courses:
Course 1: 89
Course 2: 76
Course 3: 67
Course 4: 89
Course 5: 09
Enter USN: 234
Enter Name: 67
Enter Semester: 89
Enter Internal Marks for 5 courses:
Course 1: 67
Course 2: 67
Course 3: 56
Course 4: 45
Course 5: 34
Enter External Marks for 5 courses:
Course 1: 56
Course 2: 768
Course 3: 89
Course 4: 65
Course 5: 45
USN: 234
Name: kjk
Semester: 2
Internal Marks:
Course 1: 98
Course 2: 90
Course 3: 87
Course 4: 98

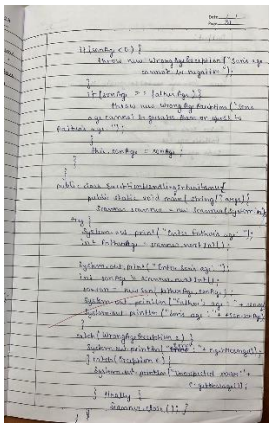

Enter Semester: 2
Enter Internal Marks for 5 courses:
Course 1: 98
Course 2: 90
Course 3: 87
Course 4: 98
Course 5: 87
Enter External Marks for 5 courses:
Course 1: 89
Course 2: 76
Course 3: 67
Course 4: 89
Course 5: 09
Enter USN: 234
Enter Name: 67
Enter Semester: 89
Enter Internal Marks for 5 courses:
Course 1: 67
Course 2: 67
Course 3: 56
Course 4: 45
Course 5: 34
Enter External Marks for 5 courses:
Course 1: 56
Course 2: 768
Course 3: 89
Course 4: 65
Course 5: 45
USN: 234
Name: kjk
Semester: 2
Internal Marks:
Course 1: 98
Course 2: 90
Course 3: 87
Course 4: 98
Course 5: 87
External Marks:
Course 1: 89
Course 2: 76
Course 3: 67
Course 4: 89
Course 5: 9
Final Marks:
Course 1: 187
Course 2: 166
Course 3: 154
Course 4: 187
Course 5: 96

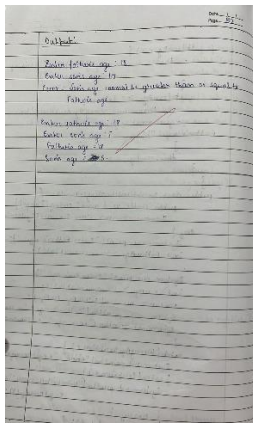
Program 7

Exception Handling Inheritance

Algorithm:







Code:

```
import java.util.Scanner;

class PrintInfo {
  static void print() {
      System.out.println("Name: Kavya Singh");
      System.out.println("USN: 1BM23CS146");
   }
}


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

class Father {
   int age;

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

```
        this.age = age;
    }
}

class Son extends Father {
    int sonAge;

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

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

        try {

            System.out.print("Enter Father's age: ");
            int fatherAge = scanner.nextInt();

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

            Son son = new Son(fatherAge, sonAge);
            System.out.println("Father's age: " + son.age);
            System.out.println("Son's age: " + son.sonAge);

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

```
    }
  }
}
```

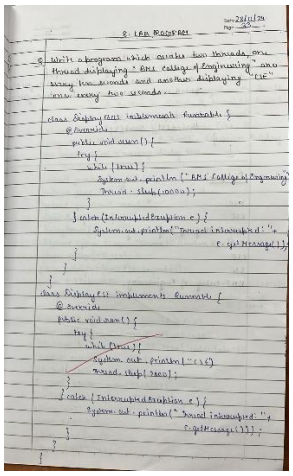
Output:

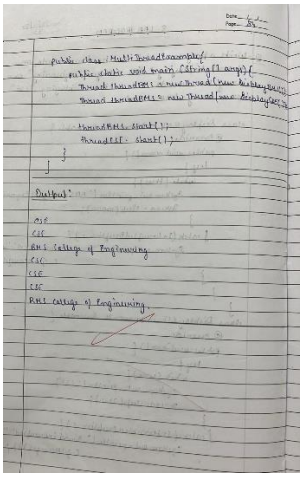
```
D:\1BM23CS146>java ExceptionHandlingInheritance
Name: Kavya Singh
USN: 1BM23CS146
Enter Father's age: 40
Enter Son's age: 13
Father's age: 40
Son's age: 13

D:\1BM23CS146>javac ExceptionHandlingInheritance.java

D:\1BM23CS146>java ExceptionHandlingInheritance
Name: Kavya Singh
USN: 1BM23CS146
Enter Father's age: 23
Enter Son's age: 28
Error: Son's age cannot be greater than or equal to Father's age.
```

Program 8

Threads

Algorithm:



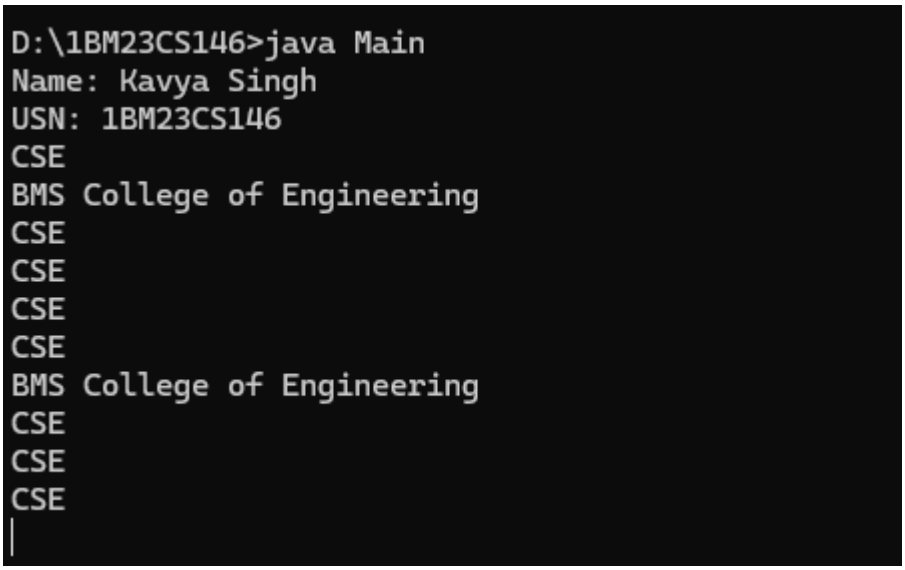


Code:

```
class PrintInfo {
  static void print() {
      System.out.println("Name: Kavya Singh");
      System.out.println("USN: 1BM23CS146");
   }
}
public class Main {

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

   static class CSEDisplayThread extends Thread {
```

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

    public static void main(String[] args) {
        PrintInfo.print();
        Thread bmsThread = new BMSDisplayThread();
        Thread cseThread = new CSEDisplayThread();

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

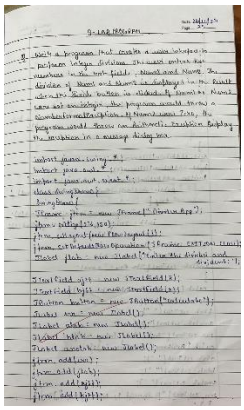
Output:

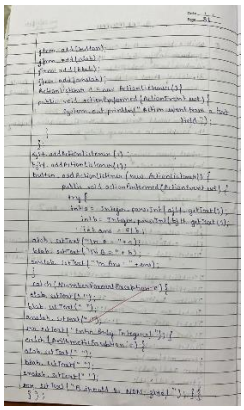


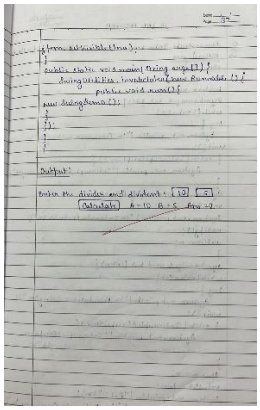
Program 9

Swing Demo

Algorithm:







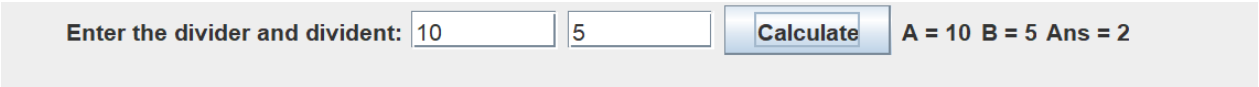
Code:

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

class SwingDemo {

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

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

        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(ajtf);
```

```
    jfrm.add(bjtf);
    jfrm.add(button);
    jfrm.add(alab);
    jfrm.add(blab);
    jfrm.add(anslab);

    ActionListener l = new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
      }
    };
    ajtf.addActionListener(l);
    bjtf.addActionListener(l);

    button.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
        try {
          int a = Integer.parseInt(ajtf.getText());
          int b = Integer.parseInt(bjtf.getText());
          int ans = a / b;
          alab.setText("A = " + a);
          blab.setText("B = " + b);
          anslab.setText("Ans = " + ans);
          err.setText("");  // Clear previous errors

        } catch (NumberFormatException e) {
          alab.setText("");
          blab.setText("");
          anslab.setText("");
          err.setText("Enter Only Integers!");
        } catch (ArithmeticException e) {
          alab.setText("");
          blab.setText("");
          anslab.setText("");
          err.setText("B should be NON zero!");
        }
      }
    });

    jfrm.setVisible(true);
  }

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

```
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new SwingDemo();
            }
        });
    }
}
```

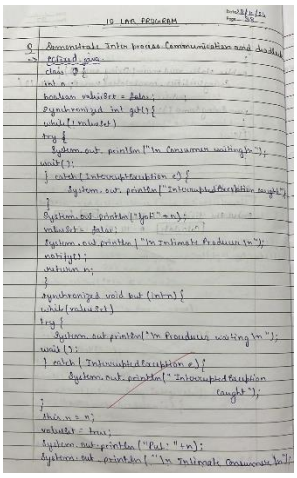
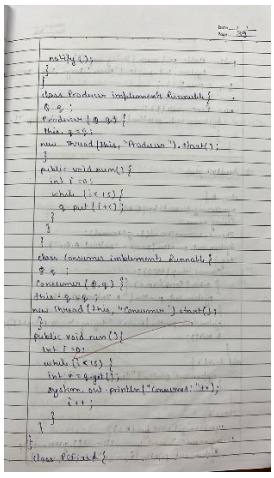
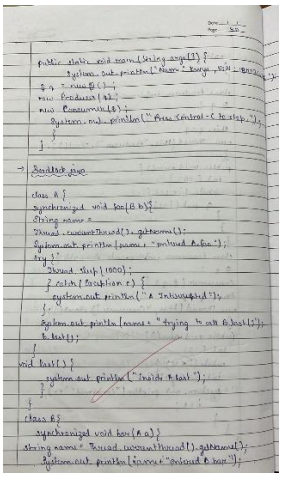
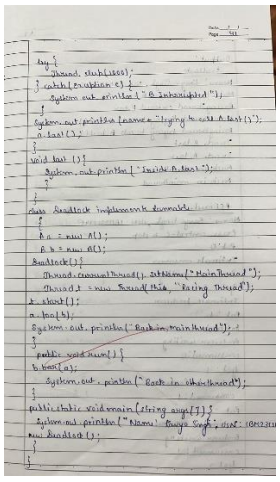
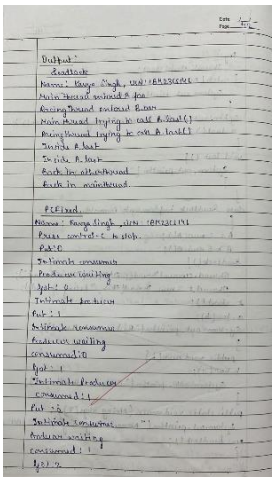
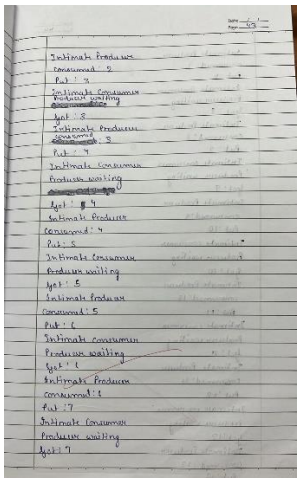
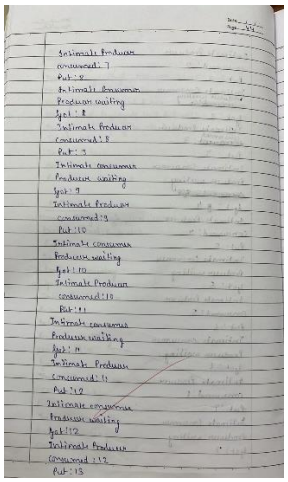
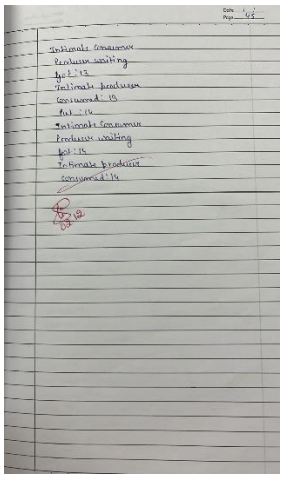
Output:



Program 10 a

Deadlock

Algorithm:

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");
        Thread t = new Thread(this, "RacingThread");
        t.start();

        a.foo(b);
        System.out.println("Back in mainthread");
    }
    public void run() {
```

```
        b.bar(a);
        System.out.println("Back in otherthread");
    }
    public static void main(String args[]) {
        System.out.println("Name: Kavya Singh, USN: 1BM23CS146");
        new Deadlock();
    }
}
```

Program 10 b

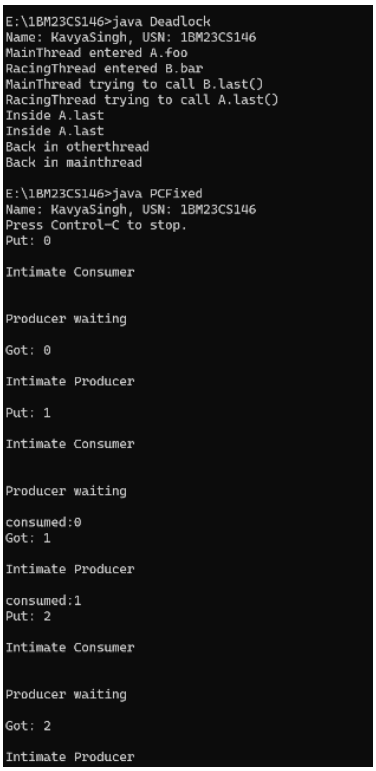
PCfixed

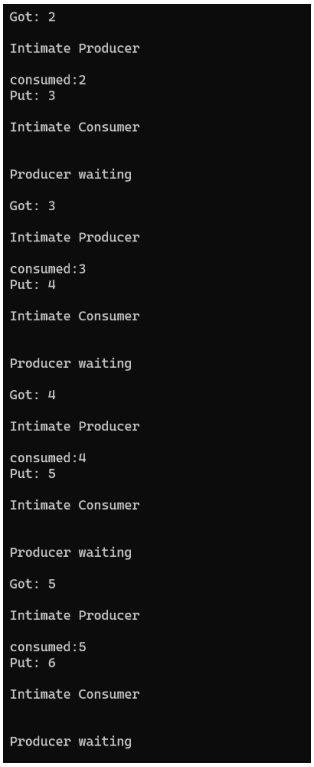
Code:

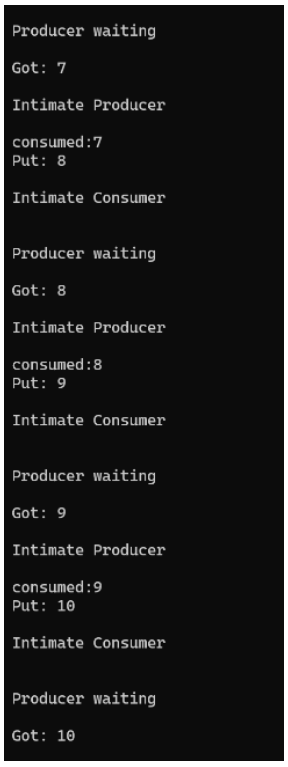
```
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (!valueSet) {
            try {
                System.out.println("\nConsumer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException 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;
```

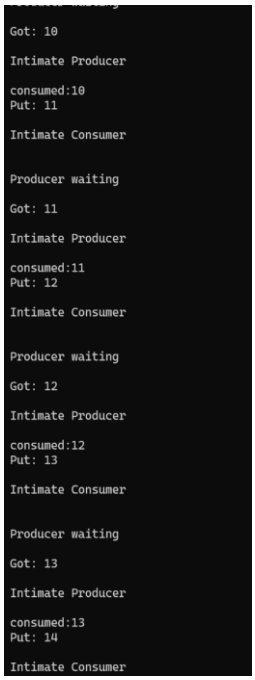
```
        System.out.println("Put: " + n);
        System.out.println("\nIntimate Consumer\n");
        notify();
    }
}
class Producer implements Runnable {
    Q q;
    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }
    public void run() {
        int i = 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: Kavya Singh, USN: 1BM23CS146");
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
}
```

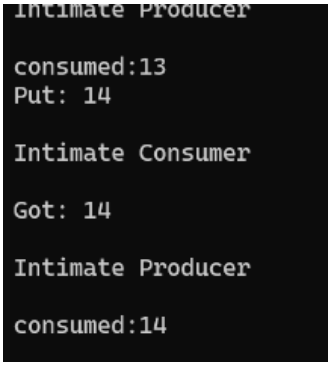
Output:


E:\1BM23CS146>java Deadlock
Name: KavyaSingh, USN: 1BM23CS146
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
RacingThread trying to call A.last()
Inside A.last
Inside A.last
Back in otherthread
Back in mainthread
E:\1BM23CS146>java PCFixed
Name: KavyaSingh, USN: 1BM23CS146
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


Got: 2
Intimate Producer
consumed:2
Put: 3
Intimate Consumer
Producer waiting
Got: 3
Intimate Producer
consumed:3
Put: 4
Intimate Consumer
Producer waiting
Got: 4
Intimate Producer
consumed:4
Put: 5
Intimate Consumer
Producer waiting
Got: 5
Intimate Producer
consumed:5
Put: 6
Intimate Consumer
Producer waiting


Producer waiting
Got: 7
Intimate Producer
consumed:7
Put: 8
Intimate Consumer
Producer waiting
Got: 8
Intimate Producer
consumed:8
Put: 9
Intimate Consumer
Producer waiting
Got: 9
Intimate Producer
consumed:9
Put: 10
Intimate Consumer
Producer waiting
Got: 10


Got: 10
Intimate Producer
consumed:10
Put: 11
Intimate Consumer
Producer waiting
Got: 11
Intimate Producer
consumed:11
Put: 12
Intimate Consumer
Producer waiting
Got: 12
Intimate Producer
consumed:12
Put: 13
Intimate Consumer
Producer waiting
Got: 13
Intimate Producer
consumed:13
Put: 14
Intimate Consumer


Intimate Producer
consumed:13
Put: 14
Intimate Consumer
Got: 14
Intimate Producer
consumed:14