VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Kavana M A (1BM23CS145)

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

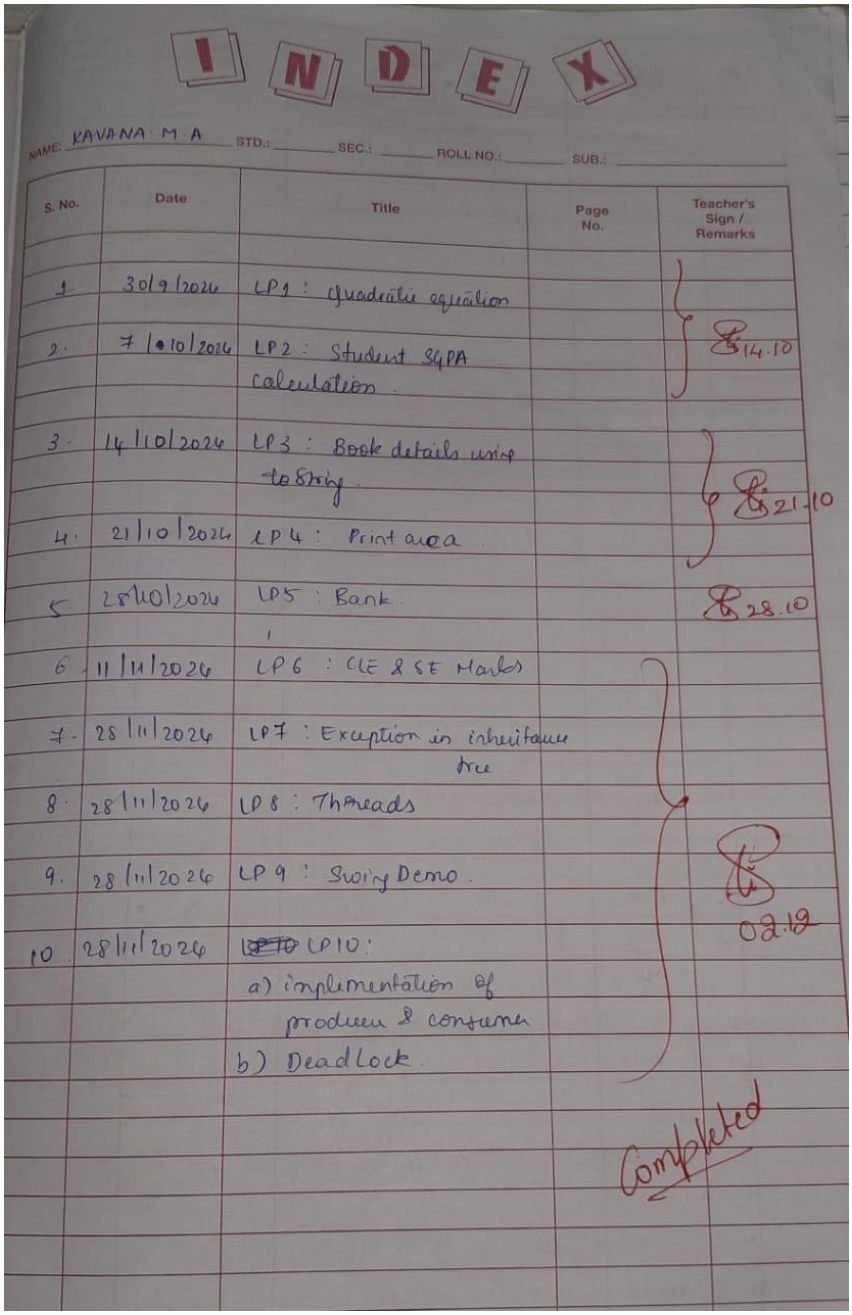
This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by **Kavana M A (1BM23CS145),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** 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.

Surabhi S Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link:
https://github.com/kavana-ma/OOJ-Lab-Programs

INDEX of Observation:



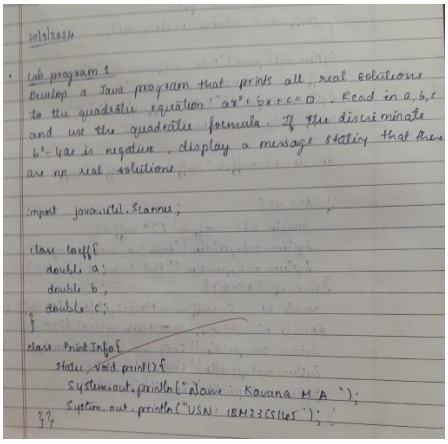
INDEX

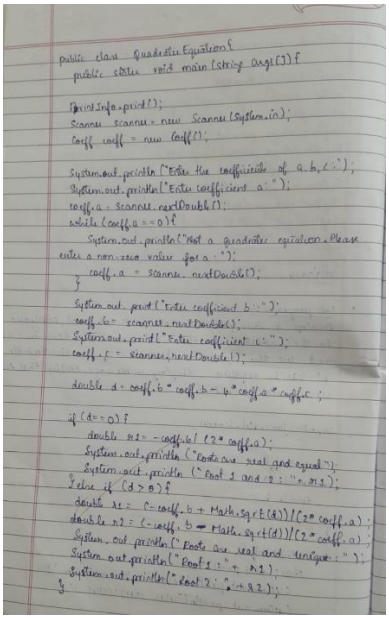
NAME: KAVANA M A STD.: SEC.: ROLL NO.: SUB.:

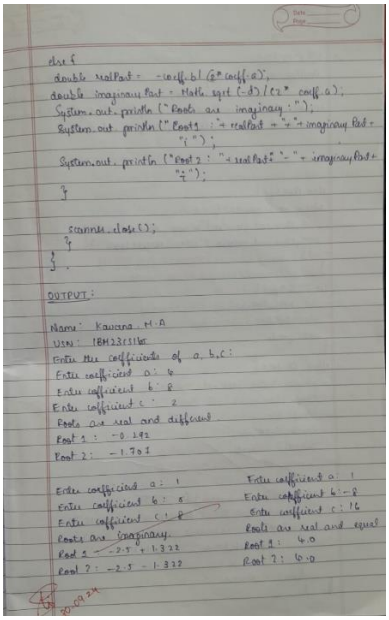
S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	30/9/2024	LP1 : Quadratic equation		14.10
2.	7/10/2024	LP2 : Student SGPA calculation		
3.	14/10/2024	LP3 : Book details using toString		21.10
4.	21/10/2024	LP4 : Print area		
5	28/10/2024	LP5 : Bank		28.10
6.	11/11/2024	LP6 : CIE & SE Marks		
7.	28/11/2024	LP7 : Exception in inheritance tree		
8.	28/11/2024	LP8 : Threads		
9.	28/11/2024	LP9 : Swing Demo.		02.12
10	28/11/2024	LP10 : a) implementation of producer & consumer b) Deadlock.		Completed

Program 1
Implement Quadratic Equation

Observation:







Code:

```
import java.util.Scanner;

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

class PrintInfo {
  static void print() {
      System.out.println("Name: Kavana M A");
      System.out.println("USN: 1BM23CS145");
   }
}

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

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

      System.out.println("Enter the coefficients of a, b, c:");
```

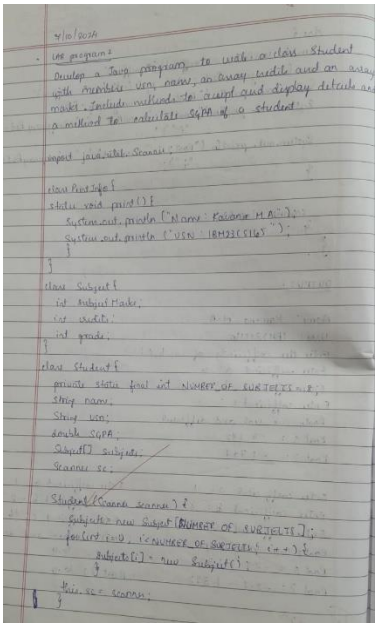
```
        System.out.print("Enter coefficient a: ");
        coeff.a = scanner.nextDouble();
        while (coeff.a == 0) {
            System.out.println("Not a quadratic equation. Please enter a non-zero value for a:");
            coeff.a = scanner.nextDouble();
        }

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

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

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

        scanner.close();
    }
}
```

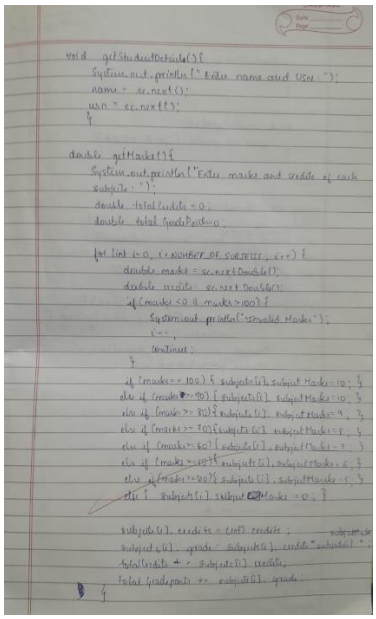
Output :

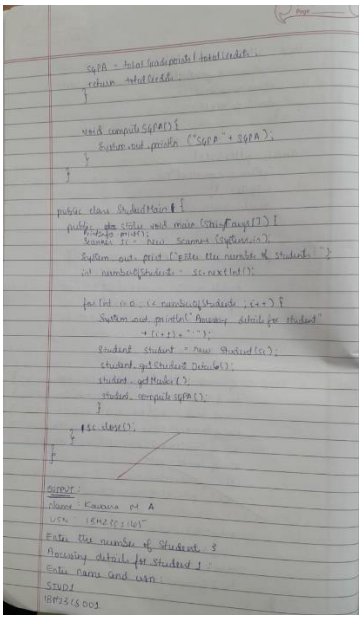
```
D:\1BM23CS145>java QuadraticEquation
D:\1BM23CS145>javac QuadraticEquation.java

D:\1BM23CS145>java QuadraticEquation
Name: Kavana M A
USN: 1BM23CS145
Enter the coefficients of a, b, c:
Enter coefficient a: 1
Enter coefficient b: 3
Enter coefficient c: 2
Roots are real and different.
Root 1: -1.0
Root 2: -2.0
```

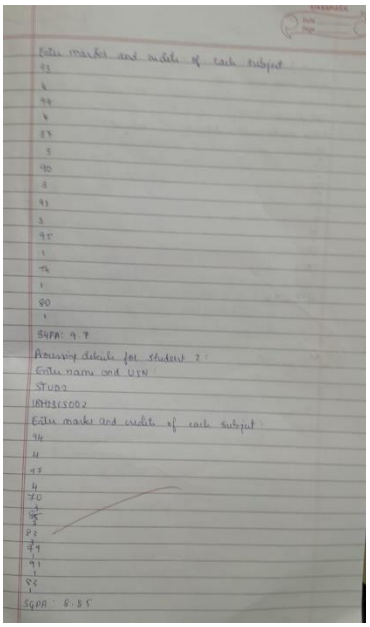
Program 2
Student SGPA Calculation

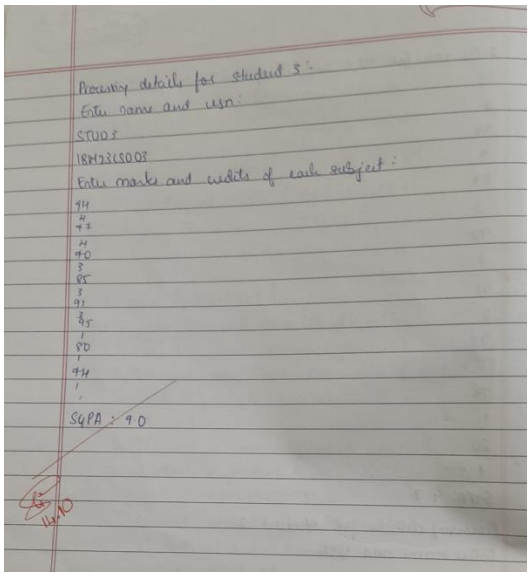
Observation:











Code:

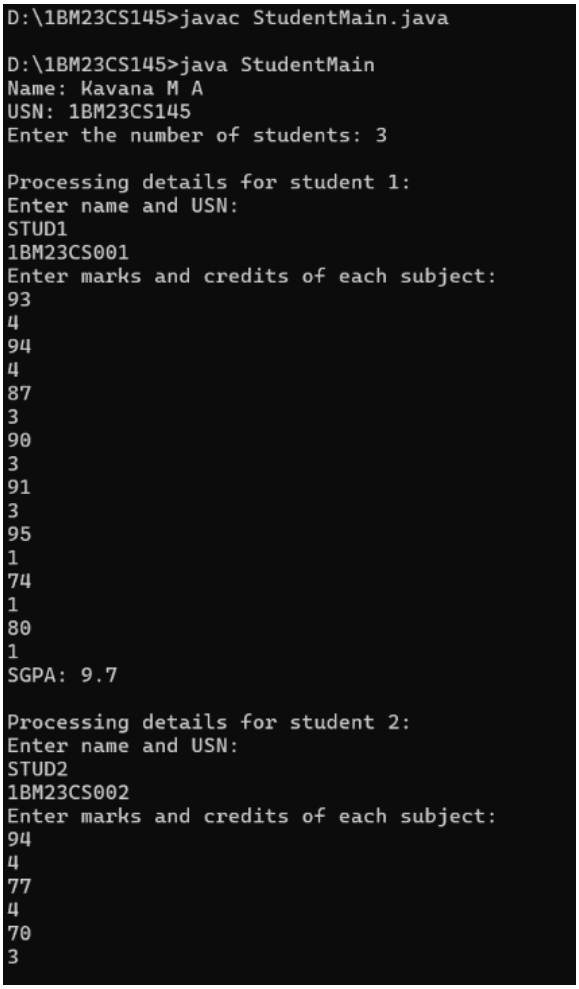
```
import java.util.Scanner;

class PrintInfo {
    static void print() {
```

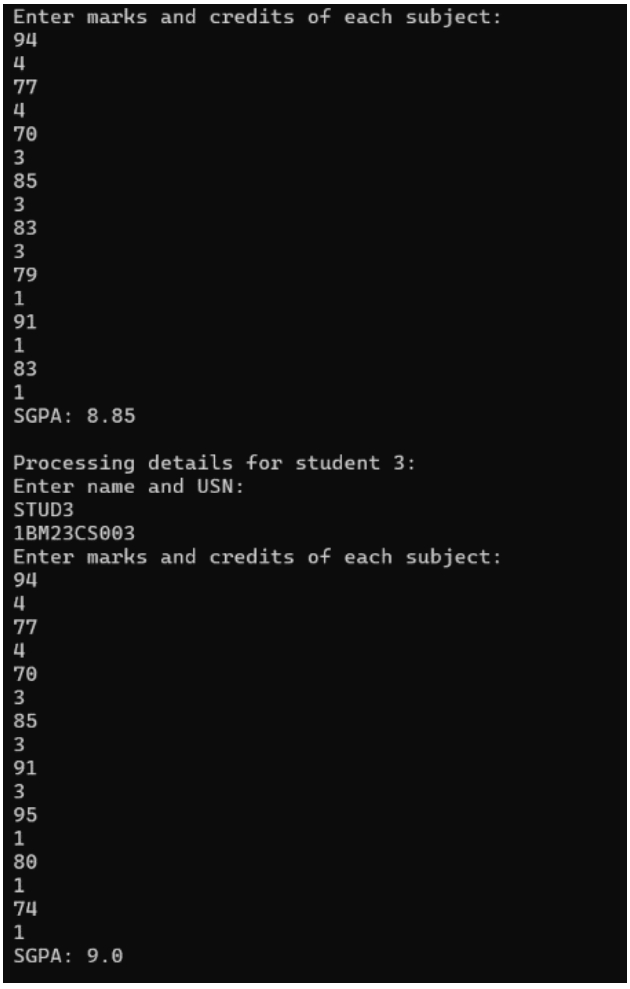
```
        System.out.println("Name: Kavana M A");
        System.out.println("USN: 1BM23CS145");
    }
}

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) {
        PrintInfo.print();
        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();
        }

        sc.close();
    }
}
```

Output:

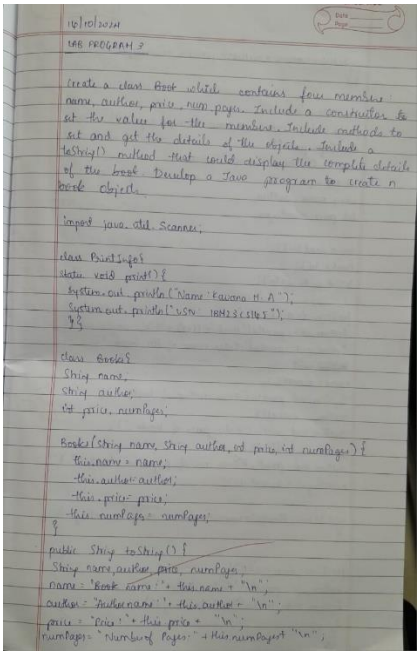


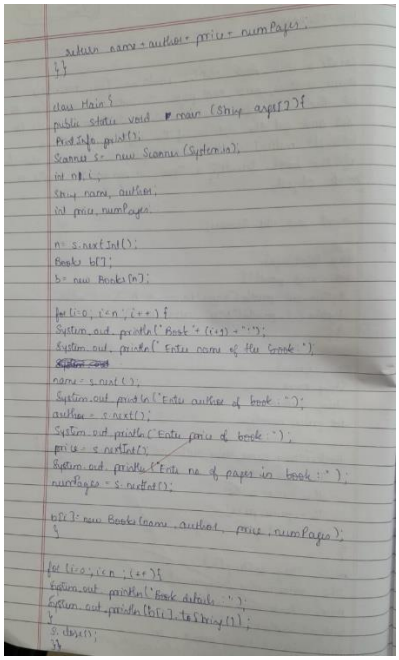


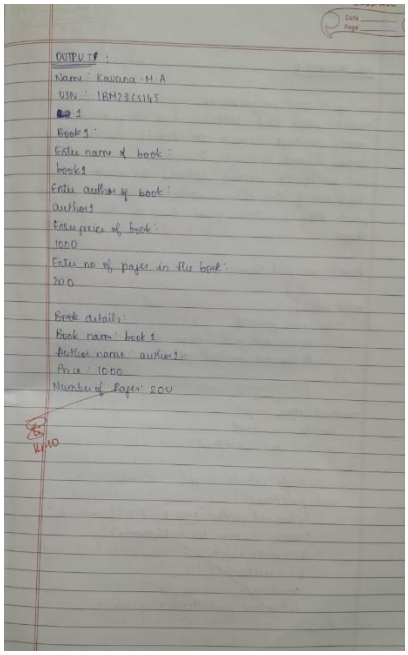
Program 3

Book details using toString

Observation:







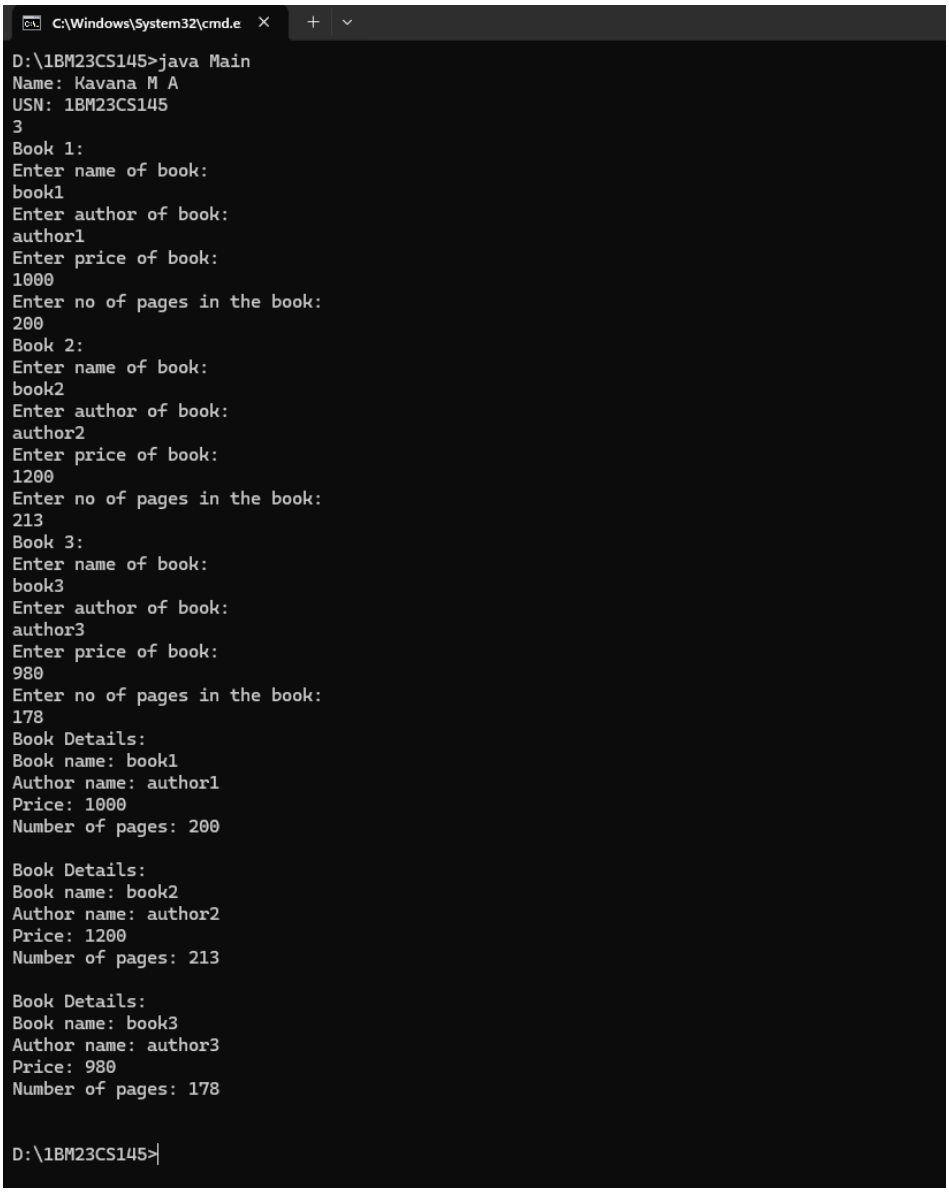
Code:

```
import java.util.Scanner;

class PrintInfo {
  static void print() {
      System.out.println("Name: Kavana M A");
      System.out.println("USN: 1BM23CS145");
   }
}

class Books {
   String name;
   String author;
   int price;
   int numPages;

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


public String toString(){
String name, author, price, numPages;
```

```
name = "Book name: " + this.name + "\n";
author = "Author name: " + this.author + "\n";
price = "Price: " + this.price + "\n";
numPages = "Number of pages: " + this.numPages + "\n";

return name + author + price + numPages;
}
}

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

PrintInfo.print();

Scanner s = new Scanner(System.in);
int n;
String name;
String author;
int price;
int numPages;
System.out.println("Enter number of books");
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());
}

s.close();
```

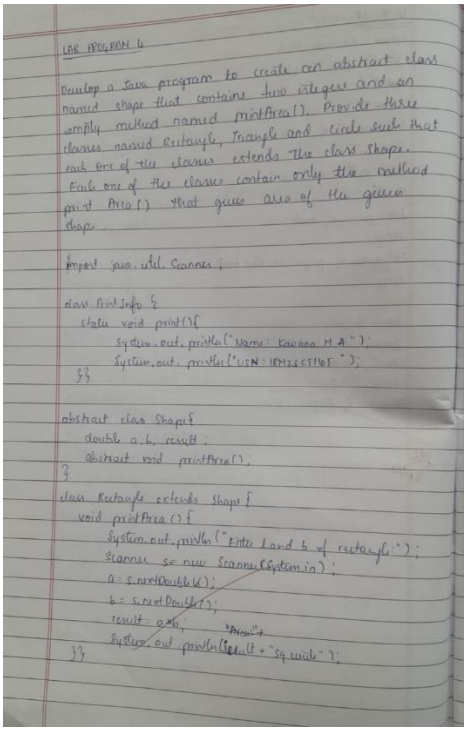
}}

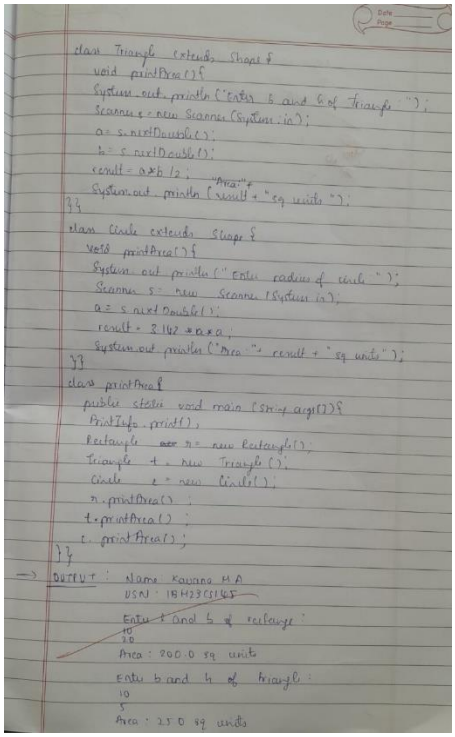
Output:

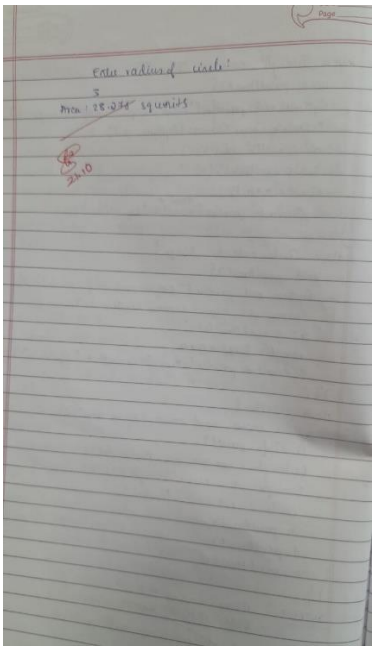

C:\Windows\System32\cmd.e
D:\1BM23CS145>java Main
Name: Kavana M A
USN: 1BM23CS145
3
Book 1:
Enter name of book:
book1
Enter author of book:
author1
Enter price of book:
1000
Enter no of pages in the book:
200
Book 2:
Enter name of book:
book2
Enter author of book:
author2
Enter price of book:
1200
Enter no of pages in the book:
213
Book 3:
Enter name of book:
book3
Enter author of book:
author3
Enter price of book:
980
Enter no of pages in the book:
178
Book Details:
Book name: book1
Author name: author1
Price: 1000
Number of pages: 200
Book Details:
Book name: book2
Author name: author2
Price: 1200
Number of pages: 213
Book Details:
Book name: book3
Author name: author3
Price: 980
Number of pages: 178
D:\1BM23CS145>

Program 4 :
PrintArea

Observation:





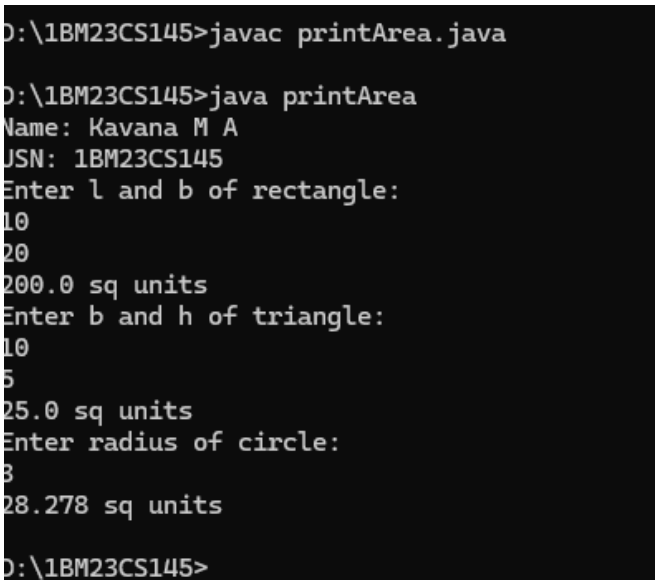


Code:

```
import java.util.Scanner;

class PrintInfo {
  static void print() {
      System.out.println("Name: Kavana M A");
      System.out.println("USN: 1BM23CS145");
   }
}

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();
```

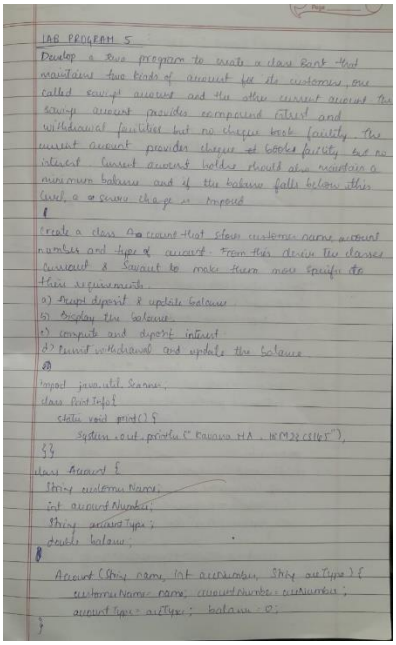
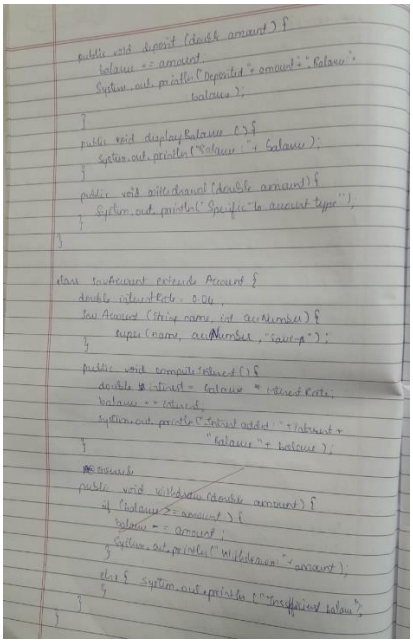
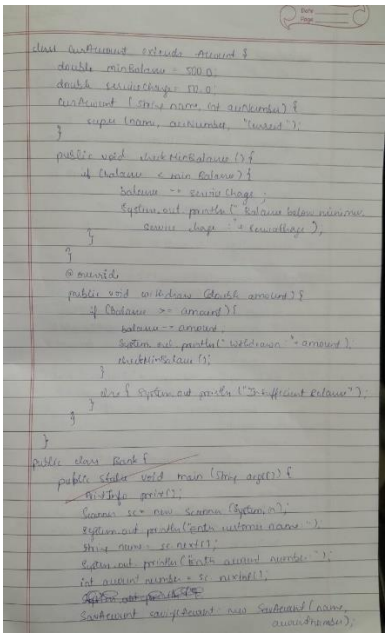
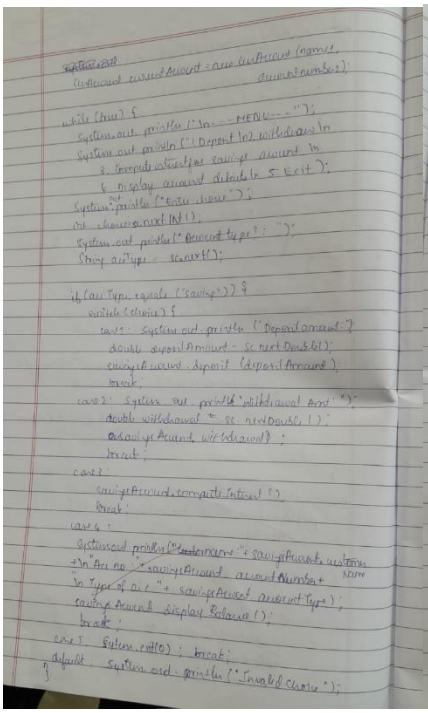
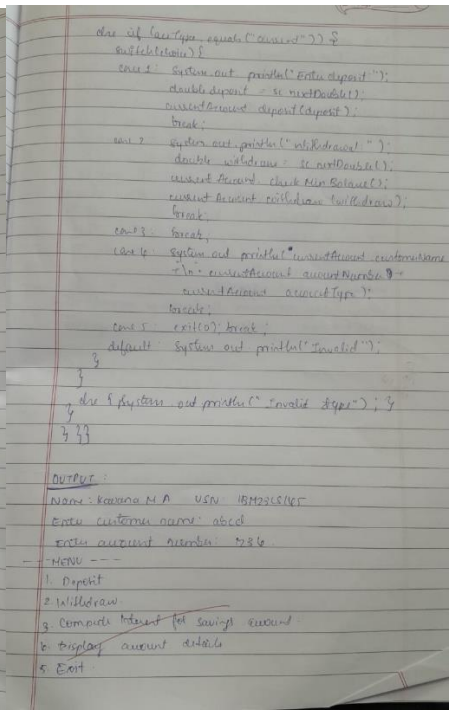
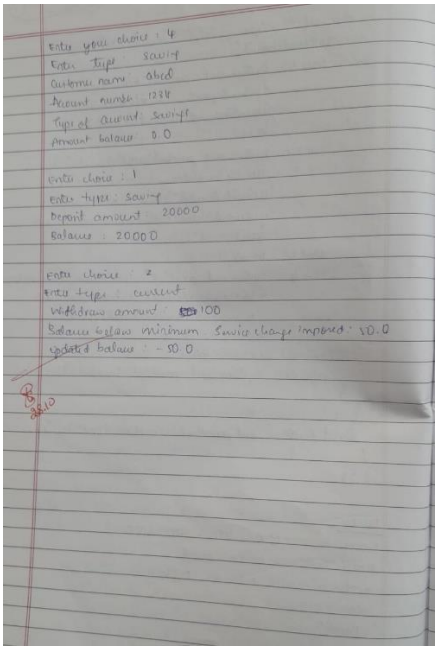
```
b=s.nextDouble();
result=a*b;
System.out.println(result+" sq units");
}
}
class Triangle extends Shape{
void printArea(){
System.out.println("Enter b and h of triangle:");
Scanner s=new Scanner(System.in);
a=s.nextDouble();
b=s.nextDouble();
result=a*b/2;
System.out.println(result+" sq units");
}
}
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[]){
PrintInfo.print();
Rectangle r=new Rectangle();
Triangle t=new Triangle();
Circle c=new Circle();
r.printArea();
t.printArea();
c.printArea();
}
}
```

Output:


D:\1BM23CS145>javac printArea.java
D:\1BM23CS145>java printArea
Name: Kavana M A
USN: 1BM23CS145
Enter l and b of rectangle:
10
20
200.0 sq units
Enter b and h of triangle:
10
5
25.0 sq units
Enter radius of circle:
3
28.278 sq units
D:\1BM23CS145>

Program 5
Bank

Observation:

Code:

```
import java.util.Scanner;

class PrintInfo {
```

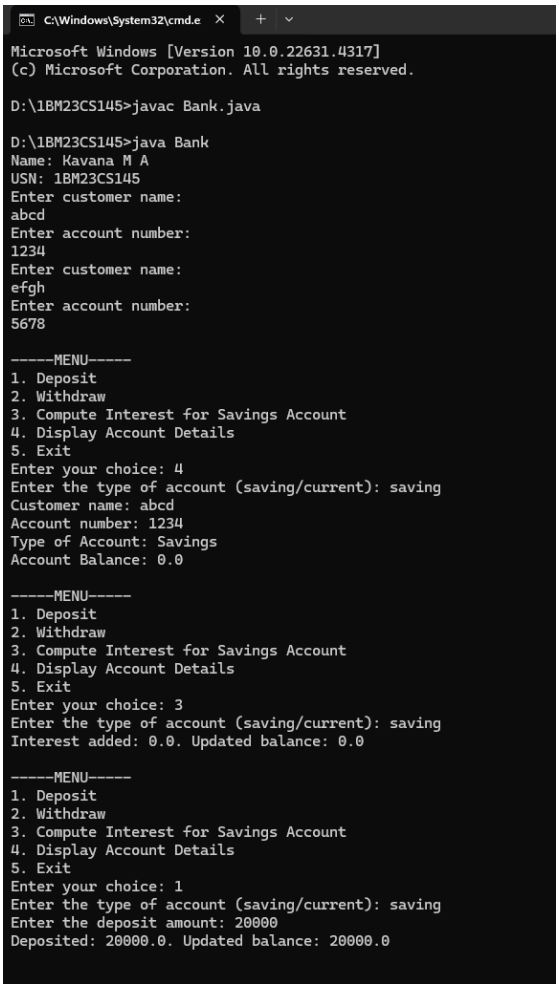
```
   static void print() {
       System.out.println("Name: Kavana M A");
       System.out.println("USN: 1BM23CS145");
    }
}

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;
```

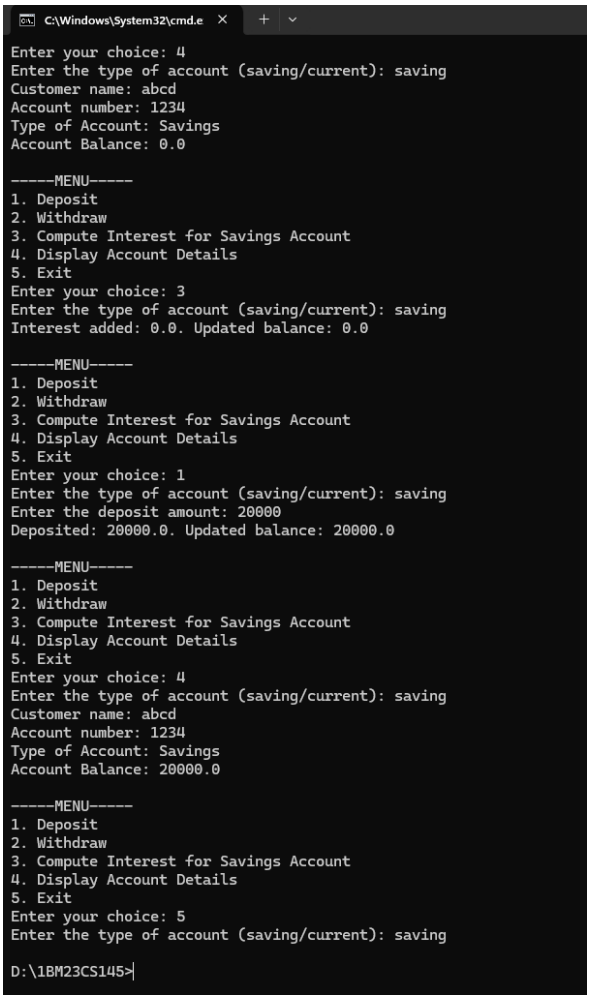
```
        balance += interest;
        System.out.println("Interest added: " + interest + ". Updated balance: " + balance);
    }


    @Override
    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);
        }
    }


    @Override
    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.");
        }
    }
  }
}
```

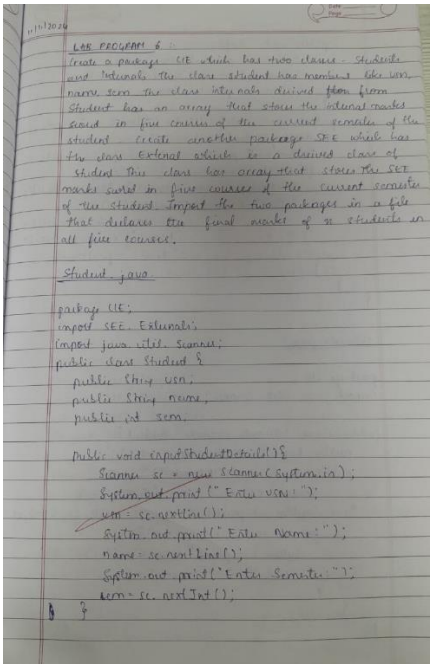
Output:

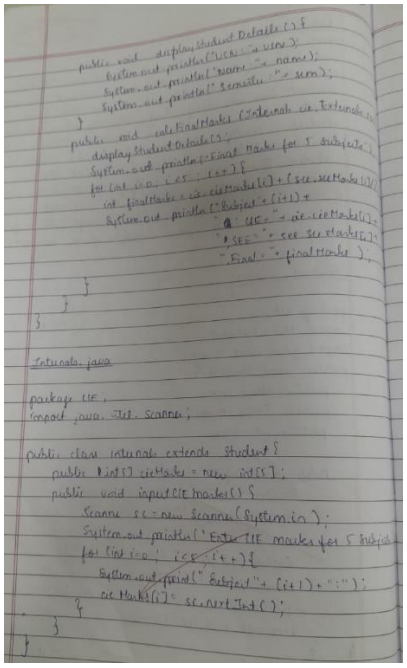

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.
D:\1BM23CS145>javac Bank.java
D:\1BM23CS145>java Bank
Name: Kavana M A
USN: 1BM23CS145
Enter customer name:
abcd
Enter account number:
1234
Enter customer name:
efgh
Enter account number:
5678
-----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: 0.0
-----MENU-----
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 3
Enter the type of account (saving/current): saving
Interest added: 0.0. Updated balance: 0.0
-----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: 20000
Deposited: 20000.0. Updated balance: 20000.0

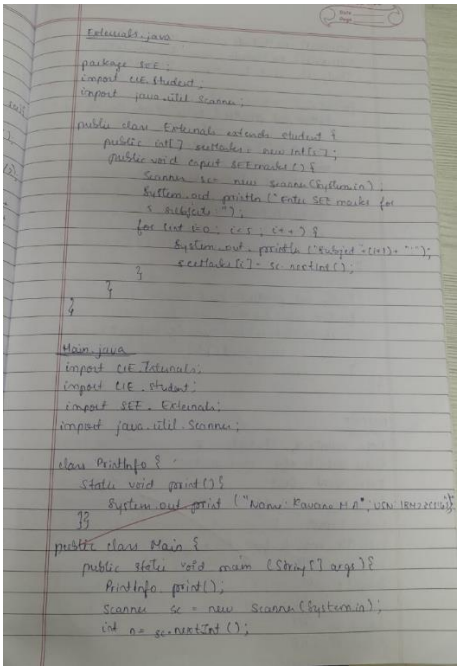

C:\Windows\System32\cmd.e
Enter your choice: 4
Enter the type of account (saving/current): saving
Customer name: abcd
Account number: 1234
Type of Account: Savings
Account Balance: 0.0
-----MENU-----
1. Deposit
2. Withdraw
3. Compute Interest for Savings Account
4. Display Account Details
5. Exit
Enter your choice: 3
Enter the type of account (saving/current): saving
Interest added: 0.0. Updated balance: 0.0
-----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: 20000
Deposited: 20000.0. Updated balance: 20000.0
-----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: 20000.0
-----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): saving
D:\1BM23CS145>

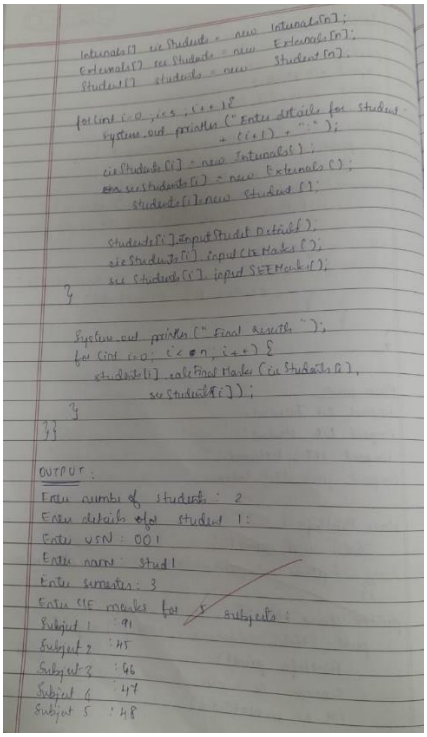
Program 6
CIE & SEE Marks

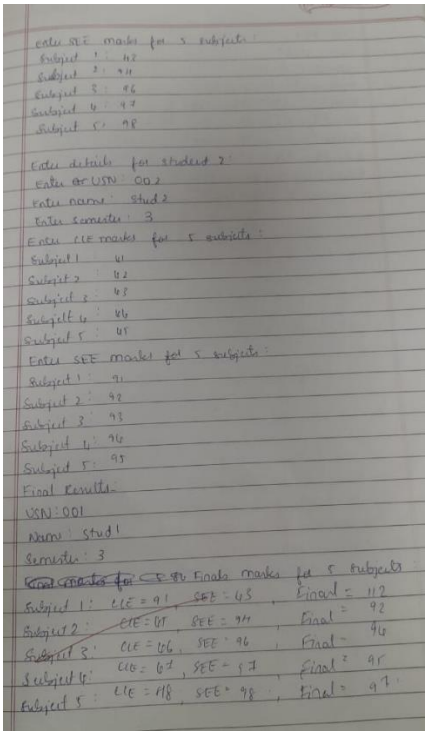
Observation:

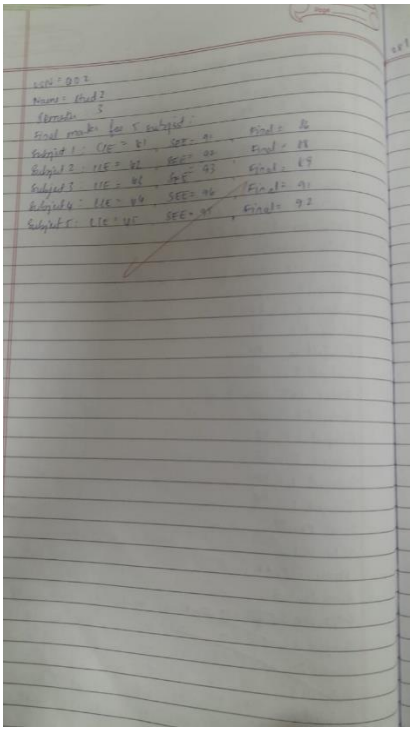












Code:

Student.java

```
Package CIE;
Import SEE.Externals;

Import java.uѲl.Scanner;

Public class Student {

 Public String usn;

 Public String name;

 Public int sem;

 Public void inputStudentDetails() {

 Scanner sc = new Scanner(System.in);

 System.out.print(“Enter USN: “);

 Usn = sc.nextLine();

 System.out.print(“Enter Name: “);

 Name = sc.nextLine();

 System.out.print(“Enter Semester: “);

 Sem = sc.nextInt();

 }

 Public void displayStudentDetails() {

 System.out.println(“USN: “ + usn);

 System.out.println(“Name: “ + name);

 System.out.println(“Semester: “ + sem);

 }

 Public void calcFinalMarks(Internals cie, Externals see) {

 displayStudentDetails();
```

```
System.out.println(“Final Marks for 5 subjects:”);

For (int I = 0; I < 5; i++) {

Int finalMarks = cie.cieMarks[i] + (see.seeMarks[i] / 2);

System.out.println(“Subject “ + (I + 1) + “: CIE = “ + cie.cieMarks[i] +

“, SEE = “ + see.seeMarks[i] +

“, Final = “ + finalMarks);

}

}

}
```

Internals.java

```
Package CIE;

Import java.uΘl.Scanner;

Public class Internals extends Student {

Public int[] cieMarks = new int[5];

Public void inputCIEmarks() {

Scanner sc = new Scanner(System.in);
System.out.println(“Enter CIE marks for 5 subjects:”);

For (int I = 0; I < 5; i++) {

System.out.print(“Subject “ + (I + 1) + “: “);

cieMarks[i] = sc.nextInt();

}

}

}
```

Externals.java

```
Package SEE;

Import CIE.Student;

Import java.uѲl.Scanner;

Public class Externals extends Student {

 Public int[] seeMarks = new int[5];

 Public void inputSEEmarks() {

 Scanner sc = new Scanner(System.in);

 System.out.println(“Enter SEE marks for 5 subjects:”);

 For (int I = 0; I < 5; i++) {

 System.out.print(“Subject “ + (I + 1) + “: “);

 seeMarks[i] = sc.nextInt();
 }
 }
}
```

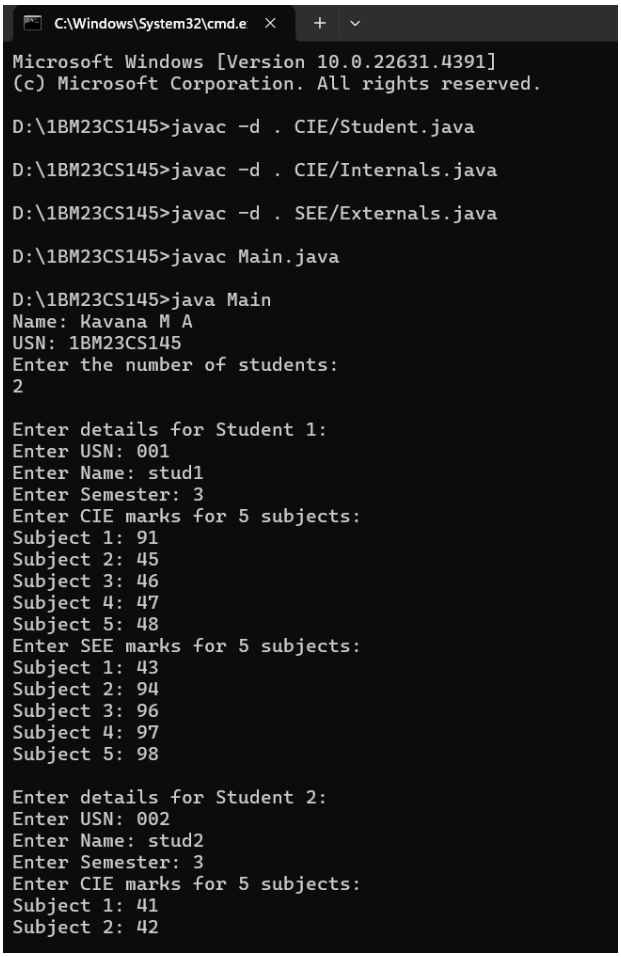
Main.java

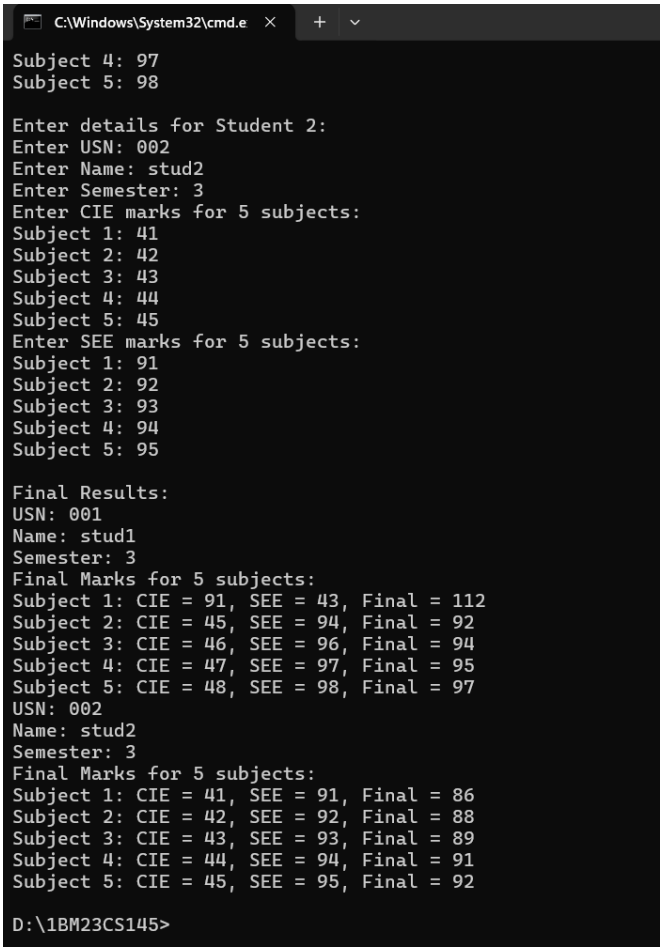
```
Import CIE.Internals;

Import SEE.Externals;

Import CIE.Student;

Import java.uѲl.Scanner;

Class PrintInfo {

 staѲc void print() {

 System.out.println(“Name: Kavana M A”);

 System.out.println(“USN: 1BM23CS145”);

 }
```

```
}
Public class Main {
 Public staϴc void main(String[] args) {
 PrintInfo.print();
 Scanner sc = new Scanner(System.in);
 System.out.println(“Enter the number of students: “);
 Int n = sc.nextInt();
 Internals[] cieStudents = new Internals[n];
 Externals[] seeStudents = new Externals[n];
 Student[] students = new Student[n];
// Input details and marks for each student
 For (int I = 0; I < n; i++) {
 System.out.println(“\nEnter details for Student “ + (I + 1) + “:”);
 cieStudents[i] = new Internals();
 seeStudents[i] = new Externals();
 students[i] = new Student();
 students[i].inputStudentDetails();
 cieStudents[i].inputCIEmarks();
 seeStudents[i].inputSEEmarks();
 }
 // Display final results
 System.out.println(“\nFinal Results:”);
 For (int I = 0; I < n; i++) {
 Students[i].calcFinalMarks(cieStudents[i], seeStudents[i]);
```

```
 }

 }
}
```

Output:

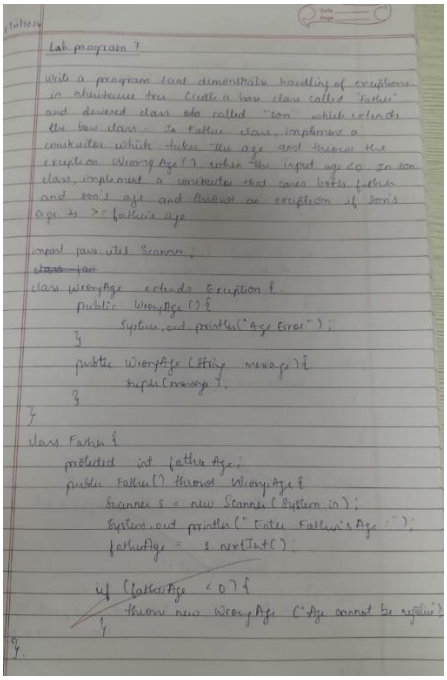

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4391]
(c) Microsoft Corporation. All rights reserved.
D:\1BM23CS145>javac -d . CIE/Student.java
D:\1BM23CS145>javac -d . CIE/Internals.java
D:\1BM23CS145>javac -d . SEE/Externals.java
D:\1BM23CS145>javac Main.java
D:\1BM23CS145>java Main
Name: Kavana M A
USN: 1BM23CS145
Enter the number of students:
2
Enter details for Student 1:
Enter USN: 001
Enter Name: stud1
Enter Semester: 3
Enter CIE marks for 5 subjects:
Subject 1: 91
Subject 2: 45
Subject 3: 46
Subject 4: 47
Subject 5: 48
Enter SEE marks for 5 subjects:
Subject 1: 43
Subject 2: 94
Subject 3: 96
Subject 4: 97
Subject 5: 98
Enter details for Student 2:
Enter USN: 002
Enter Name: stud2
Enter Semester: 3
Enter CIE marks for 5 subjects:
Subject 1: 41
Subject 2: 42


C:\Windows\System32\cmd.e
Subject 4: 97
Subject 5: 98
Enter details for Student 2:
Enter USN: 002
Enter Name: stud2
Enter Semester: 3
Enter CIE marks for 5 subjects:
Subject 1: 41
Subject 2: 42
Subject 3: 43
Subject 4: 44
Subject 5: 45
Enter SEE marks for 5 subjects:
Subject 1: 91
Subject 2: 92
Subject 3: 93
Subject 4: 94
Subject 5: 95
Final Results:
USN: 001
Name: stud1
Semester: 3
Final Marks for 5 subjects:
Subject 1: CIE = 91, SEE = 43, Final = 112
Subject 2: CIE = 45, SEE = 94, Final = 92
Subject 3: CIE = 46, SEE = 96, Final = 94
Subject 4: CIE = 47, SEE = 97, Final = 95
Subject 5: CIE = 48, SEE = 98, Final = 97
USN: 002
Name: stud2
Semester: 3
Final Marks for 5 subjects:
Subject 1: CIE = 41, SEE = 91, Final = 86
Subject 2: CIE = 42, SEE = 92, Final = 88
Subject 3: CIE = 43, SEE = 93, Final = 89
Subject 4: CIE = 44, SEE = 94, Final = 91
Subject 5: CIE = 45, SEE = 95, Final = 92
D:\1BM23CS145>

Program 7

Exception in inheritance tree

Observation:



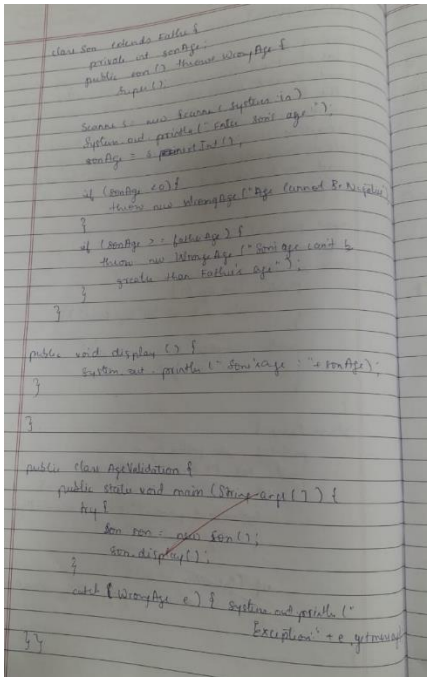
Lab program 7

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

import java.util.Scanner;

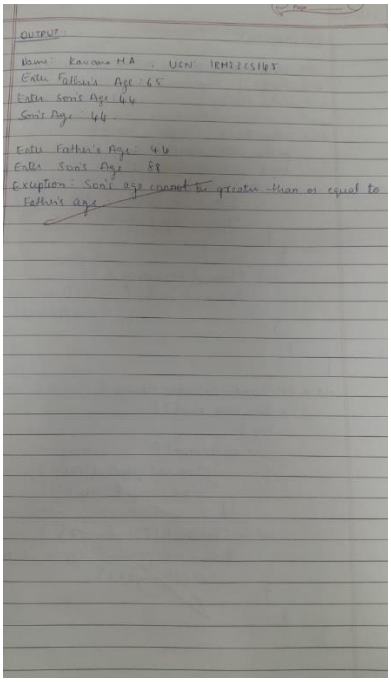
class WrongAge extends Exception {
public WrongAge() {
System.out.println("Age Error");
}
public WrongAge(String message) {
super(message);
}
}

class Father {
protected int fatherAge;
public Father() throws WrongAge {
Scanner s = new Scanner(System.in);
System.out.println("Enter Father's Age:");
fatherAge = s.nextInt();
if (fatherAge < 0) {
throw new WrongAge("Age cannot be negative");
}
}



class Son extends Father {
private int sonAge;
public Son() throws WrongAge {
super();
Scanner s = new Scanner(System.in);
System.out.println("Enter Son's age:");
sonAge = s.nextInt();
if (sonAge < 0) {
throw new WrongAge("Age cannot be negative");
}
if (sonAge >= fatherAge) {
throw new WrongAge("Son's age can't be greater than Father's age");
}
}
public void display() {
System.out.println("Son's age:" + sonAge);
}
}

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



OUTPUT

Name: Kavana M A USN: 1EM22CS168

Enter Father's Age: 65
Enter Son's Age: 44
Son's Age: 44

Enter Father's Age: 46
Enter Son's Age: 88
Exception: Son's age cannot be greater than or equal to Father's age

Code:

```
import java.util.Scanner;

// Custom exception class
class WrongAge extends Exception {
    // Default constructor
    public WrongAge() {
        super("Age Error");
    }

    // Parameterized constructor
    public WrongAge(String message) {
        super(message);
    }
}

// Father class
class Father {
    protected int fatherAge;

    // Constructor
    public Father() throws WrongAge {
        Scanner s = new Scanner(System.in);
```

```
        System.out.print("Enter Father's Age: ");
        fatherAge = s.nextInt();

        // Throw exception if age is negative
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
}

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

    // Constructor
    public Son() throws WrongAge {
        super(); // Call the parent class constructor

        Scanner s = new Scanner(System.in);
        System.out.print("Enter Son's Age: ");
        sonAge = s.nextInt();

        // Throw exception if son's age is invalid
        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to Father's age");
        }
    }

    // Method to display the son's age
    public void display() {
        System.out.println("Son's Age: " + sonAge);
    }
}

// Main class
public class AgeValidation {
    public static void main(String[] args) {
        System.out.println("Name: Kavana M A, USN: 1BM23CS145");
        try {
            // Create a Son object
            Son son = new Son();
            son.display(); // Display son's age
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
```

```
    }
  }
}
```

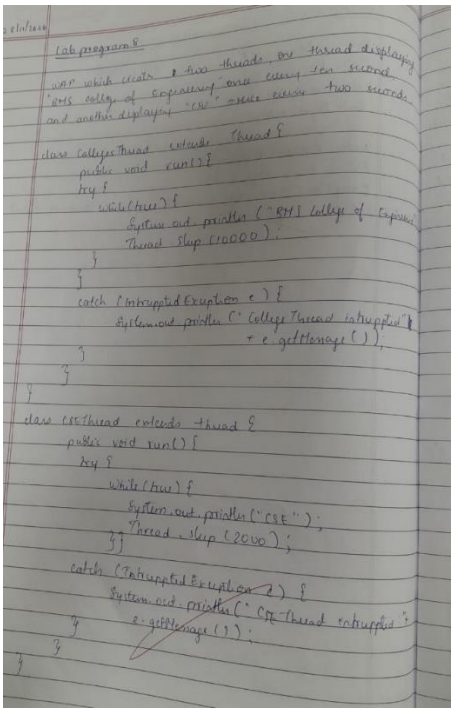
Output:

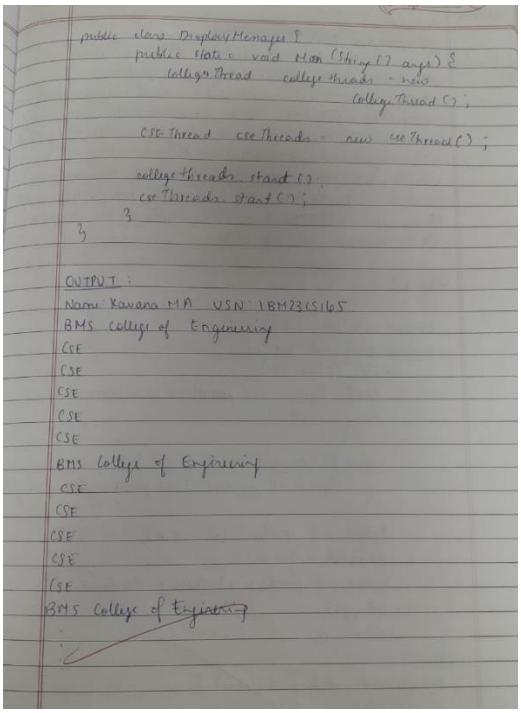
```
D:\1BM23CS145>java AgeValidation
Name: Kavana M A, USN: 1BM23CS145
Enter Father's Age: 65
Enter Son's Age: 44
Son's Age: 44

D:\1BM23CS145>java AgeValidation
Name: Kavana M A, USN: 1BM23CS145
Enter Father's Age: 44
Enter Son's Age: 88
Exception: Son's age cannot be greater than or equal to Father's age
```

Program 8

Threads

Observation:





Code:

```
Class CollegeThread 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("CollegeThread interrupted: " + e.getMessage());
      }
   }
}
// Thread to display "CSE" every 2 seconds
Class CSEThread extends Thread {
   Public void run() {
      Try {
         While (true) {
            System.out.println("CSE");
            Thread.sleep(2000); // Sleep for 2 seconds
         }
      } catch (InterruptedException e) {
```

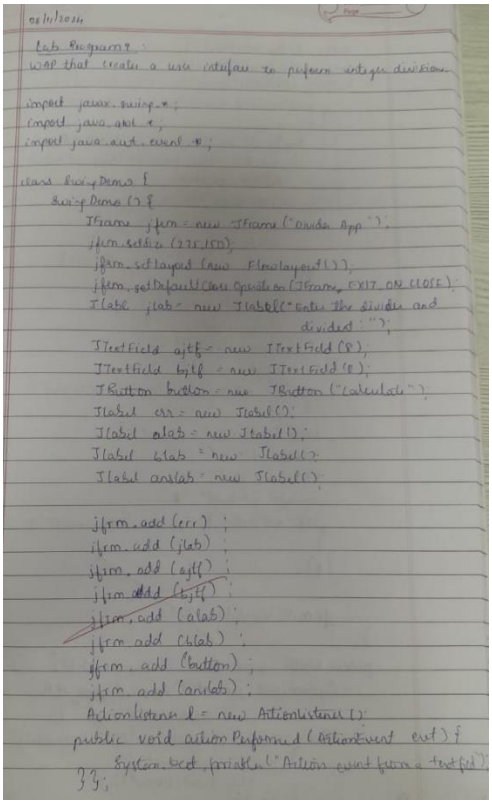
```
            System.out.println("CSEThread interrupted: " + e.getMessage());
        }
    }
}
// Main class to run the threads
Public class DisplayMessages {
    Public static void main(String[] args) {
        System.out.print("Name: Kavana M A, USN: 1BM23CS145");
        // Create threads
        CollegeThread collegeThread = new CollegeThread();
        CSEThread cseThread = new CSEThread();

        // Start threads
        collegeThread.start();
        cseThread.start();
    }
}
```

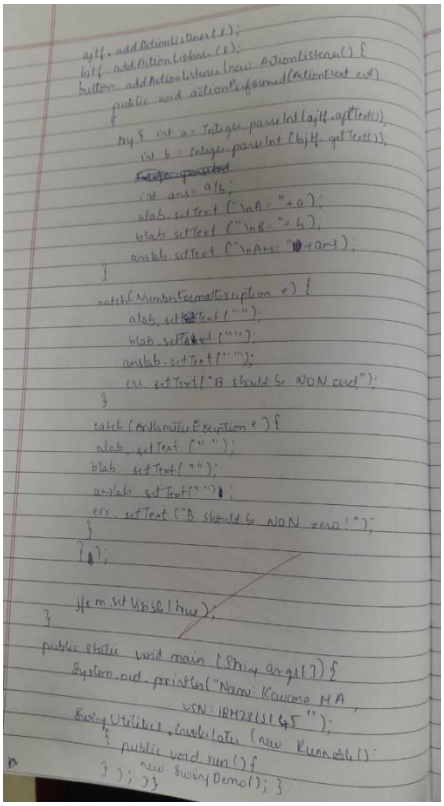
Output:

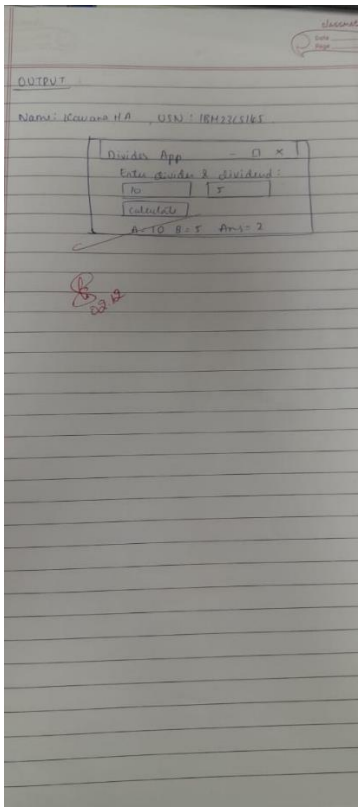
```
D:\1BM23CS145>java DisplayMessages
Name: Kavana M A, USN: 1BM23CS145BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
```

Program 9
Swing Demo

Observation:







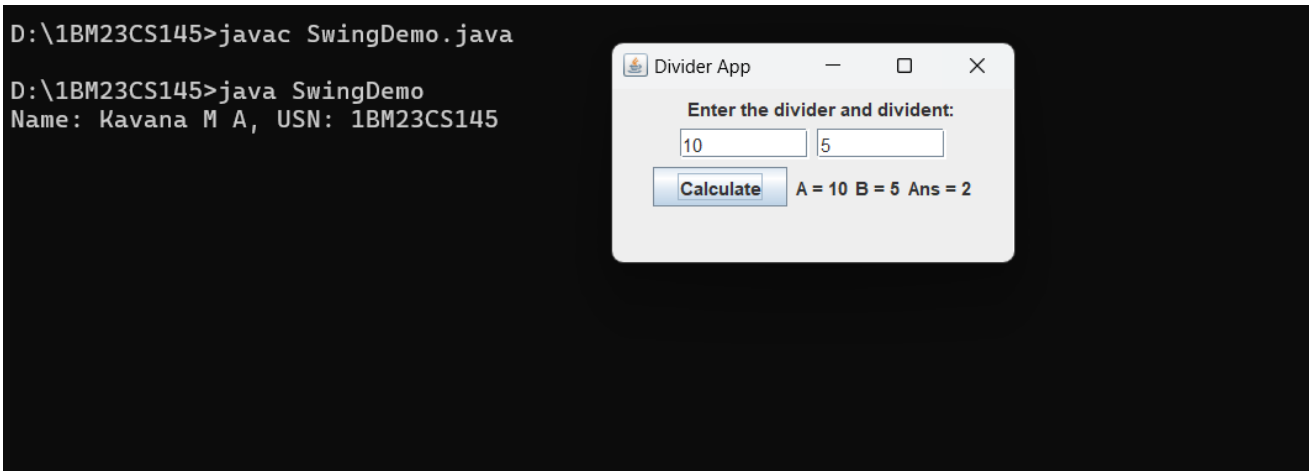
Code:

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

Class SwingDemo{
SwingDemo(){
// create jframe container
JFrame jfrm = new JFrame(“Divider App”);
Jfrm.setSize(275, 150);
Jfrm.setLayout(new FlowLayout());
// to terminate on close
Jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
JLabel jlab = new JLabel(“Enter the divider and divident:”);
// add text field for both numbers
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
```

```
// calc button
JButton button = new JButton(“Calculate”);
// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();

JLabel blab = new JLabel();
JLabel anslab = new JLabel();
// add in order ☺
Jfrm.add(err); // to display error bois
Jfrm.add(jlab);
Jfrm.add(ajtf);
Jfrm.add(bjtf);
Jfrm.add(button);
Jfrm.add(alab);
Jfrm.add(blab);
Jfrm.add(anslab);
ActionListener l = new ActionListener() {
Public void actionPerformed(ActionEvent evt) {
System.out.println(“Action event from a text field”); }
};
Ajtf.addActionListener(l);
Bjtf.addActionListener(l);
Button.addActionListener(new ActionListener() {
Public void actionPerformed(ActionEvent evt) { try{
Int a = Integer.parseInt(ajtf.getText()); int b =
Integer.parseInt(bjtf.getText()); int ans = a/b;
Alab.setText(“\nA = “ + a);
Blab.setText(“\nB = “ + b);
Anslab.setText(“\nAns = “+ ans);
}
Catch(NumberFormatException e){
Alab.setText(“”);
Blab.setText(“”);
Anslab.setText(“”);
Err.setText(“Enter Only Integers!”); }

Catch(ArithmeticException e){
Alab.setText(“”);
Blab.setText(“”);
Anslab.setText(“”);
Err.setText(“B should be NON zero!”); }
}
});
// display frame
Jfrm.setVisible(true);
}
```

```
Public static void main(String args[]){ // create  frame on event dispatching thread
System.out.println("Name: Kavana M A, USN: 1BM23CS145");
SwingUtilities.invokeLater(new Runnable(){
Public void run(){
New SwingDemo();
}
});
}}
```

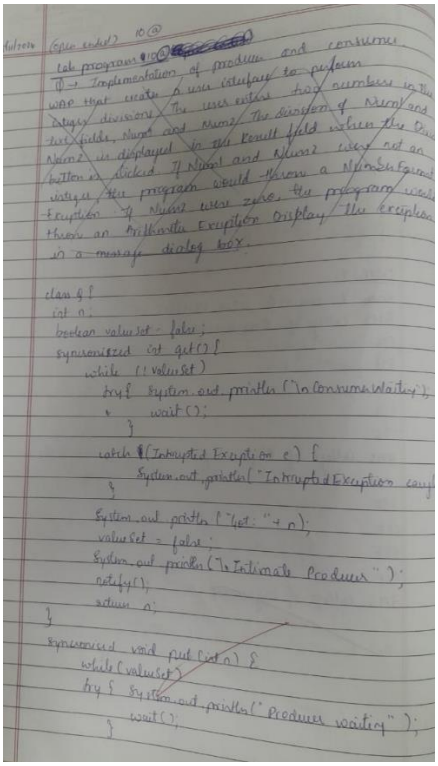
Output:

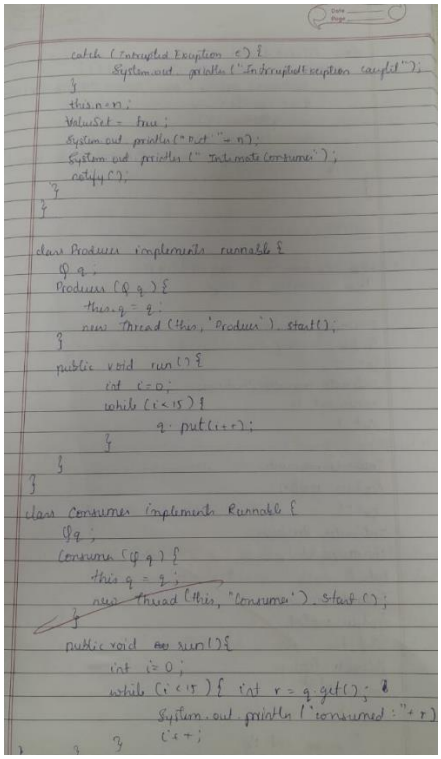


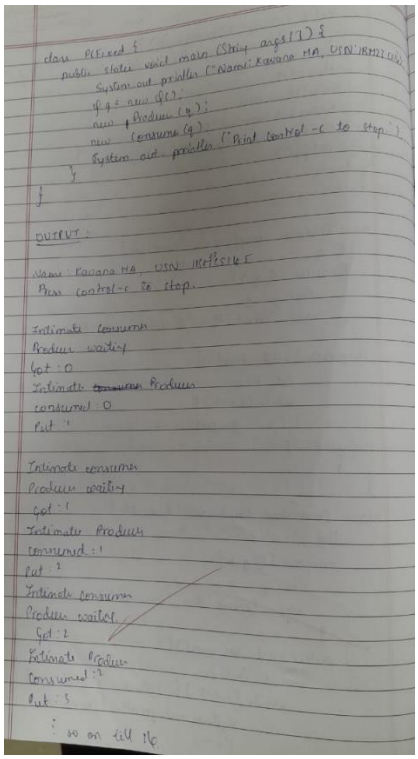
Program 10 (a)

Implementation of producer & consumer

Observation:







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;
```

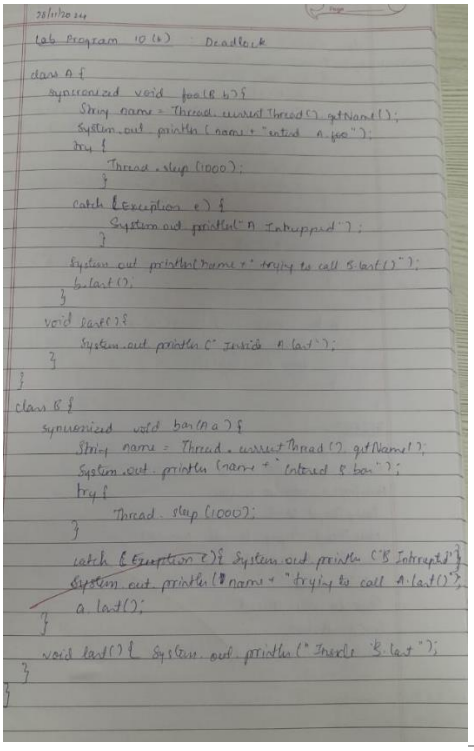
```
}

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++;
}
```

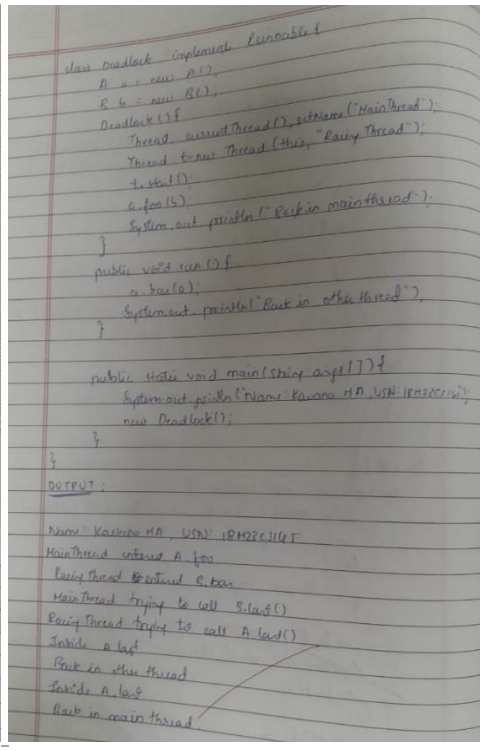
```
}
}

Class PCFixed {
Public static void main(String args[]) {
System.out.println("Name: Kavana M A, USN:1BM23CS145");
Q q = new Q();
New Producer(q);
New Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```

Output:

Program 10 (b)
Deadlock

Observation:





Code:

```
Class A {
Synchronized void foo(B b) {
String name = Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
Try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("A Interrupted");
}
System.out.println(name + " trying to call B.last()");
b.last();
}
Void last() {
System.out.println("Inside A.last");
}
}
```

```
Class B {
Synchronized void bar(A a) {
String name = Thread.currentThread().getName();
System.out.println(name + " entered B.bar");
Try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("B Interrupted");
}
System.out.println(name + " trying to call A.last()");
a.last();
}
Void last() {
System.out.println("Inside A.last");
}
}

Class Deadlock implements Runnable
{
A a = new A();
B b = new B();
Deadlock() {
Thread.currentThread().setName("MainThread");
Thread t = new Thread(this,"RacingThread");
t.start();
a.foo(b); // get lock on a in this thread.
System.out.println("Back in mainthread");
}

Public void run() {
b.bar(a); // get lock on b in otherthread.
System.out.println("Back in otherthread");
}

Public static void main(String args[]) {
System.out.println("Name: Kavana M A, USN: 1BM23CS145");
New Deadlock();
}
}
```

Output:

```
D:\1BM23CS145>java Deadlock
Name: Kavana M A, USN: 1BM23CS145
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
RacingThread trying to call A.last()
Inside A.last
Back in otherthread
Inside A.last
Back in mainthread
```