VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Kanishka Sharma (1BM23CS138)

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 **Kanishka Sharma (1BM23CS138) ,** 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/KanishkaSharma18/JAVA-LAB-PROGRAMS.git

Program 1

Implement Quadratic Equation

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

Algorithm:

30/9/21 Lab Program-1

1) Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message that there are no real solutions.

Source Code:-

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

class Quadratic1
{
    int a, b, c;
    double d, r1, r2;

    void getd()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter values of a, b c:");
        a= scanner.nextInt();
        b= scanner.nextInt();
        c= scanner.nextInt();

    while (a==0)
    {
        System.out.println("Not a quadratic Equation");
        System.out.println("Enter value of a:");
        a=scanner.nextInt();
    }
```

```
        d = b*b-4*a*c;
        if(d==0)
        {
          System.out.println("Roots are real");
          r1 = (-b)/2*a;
          System.out.println("Root1 = Root2 = "+r1);
        }
        else if(d>0)
        {
          System.out.println("Roots are distinct");
          r1 = (-b)+Math.sqrt(d)/double(2*a);
          r2 = (-b)-Math.sqrt(d)/double(2*a);
          System.out.println("Root1 = "+r1+" and
                              Root2 = "+r2);
        }
        else
        {
          System.out.println("Roots are Imaginary");
          r1 = (-b)/2*a;
          r2 = Math.sqrt(d)/2*a;
          System.out.println("Root1 = "+r1+" and
                              Root2 = "+r2);
        }
        scanner.close();
    }
}
```

```
class Quadratic
{
    public static void main(String args[])
    {
        Quadratic1 myobj = new Quadratic1();
        myobj.getd();
        System.out.println("Name : Kanishka Sharma");
        System.out.println("USN : 1BM23CS138");
    }
}
```

Output :-

```
Enter values of a,b,c :
1 2 8
Roots are imaginary
Root1 = -1.0 and Root2 = 2.64575
Name : Kanishka Sharma
USN : 1BM23CS138
```

Or

```
Enter values of a,b,c :
1 2 1
Roots are real
Root1 = Root2 = -1.0
Name : Kanishka Sharma
USN : 1BM23CS138
```

30.09

Code:

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

class Quadratic1
{
   int a,b,c;
   double d,r1,r2;

void getd()
{
        Scanner scanner=new Scanner(System.in);

        System.out.println("Enter the values of a , b and c : ");
        a=scanner.nextInt();
        b=scanner.nextInt();
        c=scanner.nextInt();


        while(a==0)
        {
           System.out.println("Not a quadratic equation");
           System.out.println("Enter the values of a : ");
           a=scanner.nextInt();
        }


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

        if(d==0)
        {
          System.out.println("Roots are real and equal");
          r1=(-b)/2*a;
          System.out.println("Root1 = Root2 = "+r1);
        }

        else if(d>0)
        {
          System.out.println("Roots are real and distinct");
          r1=(-b)+Math.sqrt(d)/(double)2*a;
          r2=(-b)-Math.sqrt(d)/(double)2*a;
          System.out.println("Root1 = "+r1+" and Root2 = "+r2);
        }

         else
        {
          System.out.println("Roots are imaginary");
```

```
            r1=(-b)/2*a;
            r2=Math.sqrt(-d)/2*a;
            System.out.println("Root1 = "+r1+" and Root2 = "+r2);
        }
      scanner.close();
}
}

class Quadratic
{
   public static void main(String args[])
   {
      Quadratic1 myobj = new Quadratic1();
      myobj.getd();
      System.out.println("Name : Kanishka Sharma");
      System.out.println("USN : 1BM23CS138");
   }
}
```

Output:

```
D:\1BM23CS138>javac Quadratic.java

D:\1BM23CS138>java Quadratic
Enter the values of a , b and c :
1 2 8
Roots are imaginary
Root1 = -1.0 and Root2 = 2.6457513110645907
Name : Kanishka Sharma
USN : 1BM23CS138
```

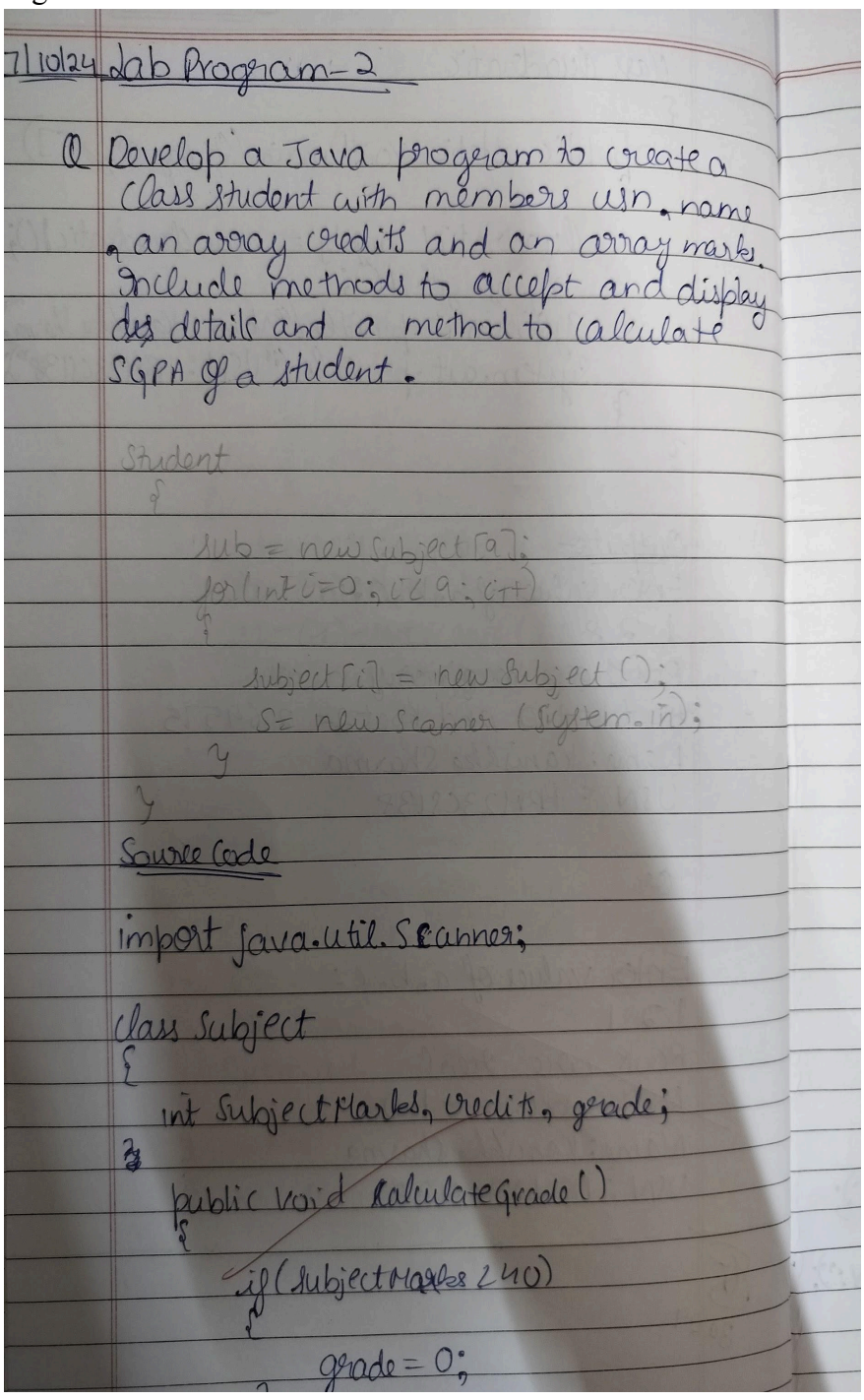
```
D:\1BM23CS138>javac Quadratic.java

D:\1BM23CS138>java Quadratic
Enter the values of a , b and c :
1 2 1
Roots are real and equal
Root1 = Root2 = -1.0
Name : Kanishka Sharma
USN : 1BM23CS138

D:\1BM23CS138>
```

Program 2

Calculation of SGPA

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

Algorithm:



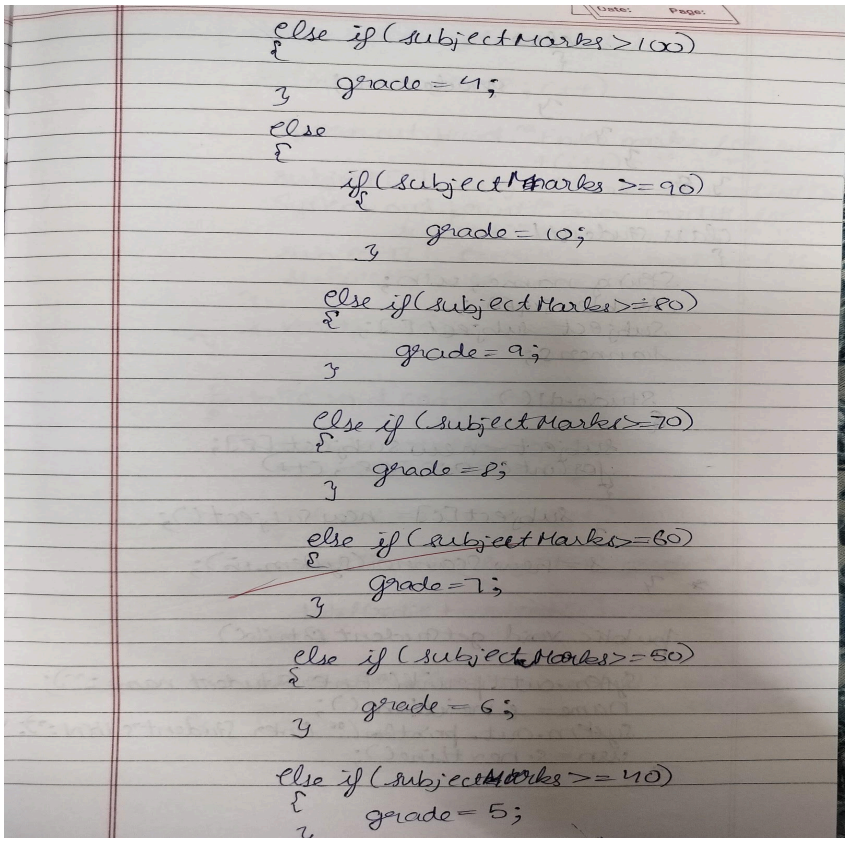
7/10/24 Lab Program-2

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

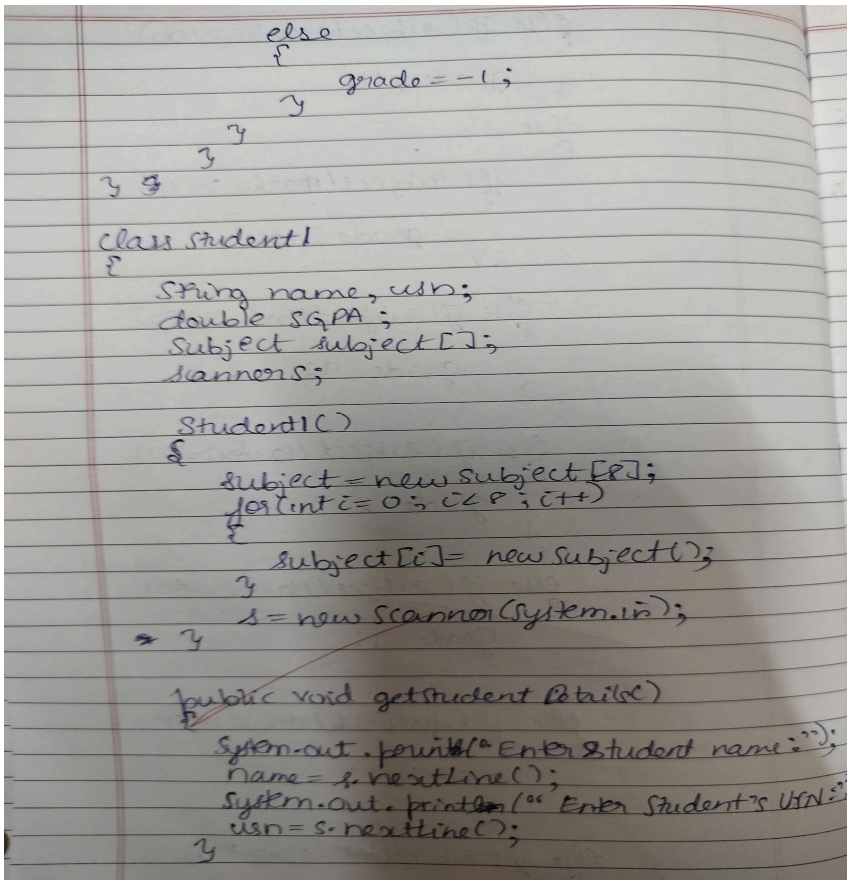
Student
{
sub = new Subject[a];
for(int i=0; i<a; i++)
{
subject[i] = new Subject();
S= new Scanner(System.in);
}
}

Source Code

```
import java.util.Scanner;

class Subject
{
  int SubjectMarks, credits, grade;
}
  public void CalculateGrade()
  {
    if(SubjectMarks<40)
    {
      grade = 0;
```



```
else if(subjectMarks>100)
{
    grade=4;
}
else
{
    if(subjectMarks>=90)
    {
        grade=10;
    }
    else if(subjectMarks>=80)
    {
        grade=9;
    }
    else if(subjectMarks>=70)
    {
        grade=8;
    }
    else if(subjectMarks>=60)
    {
        grade=7;
    }
    else if(subjectMarks>=50)
    {
        grade=6;
    }
    else if(subjectMarks>=40)
    {
        grade=5;
    }
```



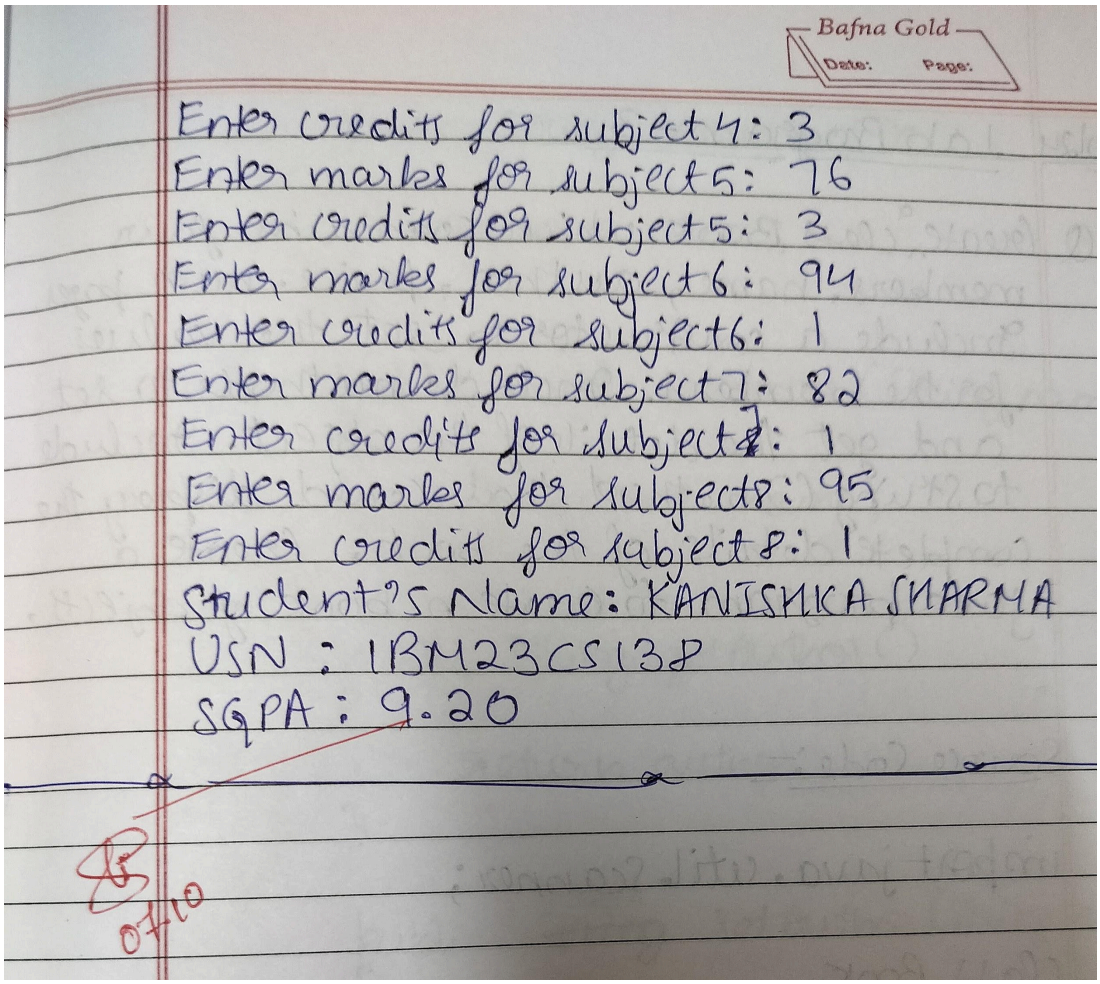
```
        else
        {
            grade=-1;
        }
    }
}
}

class Student1
{
    String name, usn;
    double SGPA;
    Subject subject[];
    Scanner s;

    Student1()
    {
        subject=new subject[8];
        for(int i=0; i<8; i++)
        {
            subject[i]= new Subject();
        }
        s=new Scanner(System.in);
    }

    public void getStudentDetails()
    {
        System.out.println("Enter Student name:");
        name=s.nextLine();
        System.out.println("Enter Student's USN:");
        usn=s.nextLine();
    }
```

```
public void getMarks()
{
    for(int i=0; i<8; i++)
    {
        System.out.print("Enter marks of subject "+(i+1)+" : ");
        subject[i].subjectMarks = s.nextInt();
        System.out.print("Enter credits of subject "+(i+1)+" : ");
        subject[i].credits = s.nextInt();
        subject[i].calculateGrade();
    }
    s.nextLine();
}

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

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

    SGPA = (totalCredits==0) ? 0 : totalPoints/totalCredits;
}
```

```
public void displayResults()
{
    System.out.println("Student's Name : "+name);
    System.out.println("USN : "+usn);
    System.out.println("SGPA : "+String.format("%.2f",SGPA));
}
}

public class Student
{
    public static void main(String[] args)
    {
        Student1 s1 = new Student1();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResults();
    }
}
```

Output :-

```
Enter student name : KANISHKA SHARMA
Enter student USN : 1BM23CS138
Enter marks for subject 1 : 91
Enter credits for subject 1 : 4
Enter marks for subject 2 : 92
Enter credits for subject 2 : 4
Enter marks for subject 3 : 76
Enter credits for subject 3 : 3
Enter marks for subject 4 : 84
```



Code:

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;
    public void calculateGrade() {
        if (subjectMarks < 40) {
            grade = 0;
        } else if (subjectMarks > 100) {
            grade = 4;
        } else {
            if (subjectMarks >= 90) {
                grade = 10;
            } else if (subjectMarks >= 80) {
                grade = 9;
            } else if (subjectMarks >= 70) {
                grade = 8;
```

```
            } else if (subjectMarks >= 60) {
                grade = 7;
            } else if (subjectMarks >= 50) {
                grade = 6;
            } else if (subjectMarks >= 40) {
                grade = 5;

            } else {
                grade = -1;
            }
        }
    }
}
class Student1 {
    String name;
    String usn;
    double SGPA;
    Subject subject[];
    Scanner s;

        Student1() {
        subject = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
        s = new Scanner(System.in);
    }
    public void getStudentDetails() {
        System.out.print("Enter student name: ");
        name = s.nextLine();
        System.out.print("Enter student USN: ");
        usn = s.nextLine();
    }
    public void getMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.print("Enter marks for subject " + (i + 1) + ": ");
            subject[i].subjectMarks = s.nextInt();
            System.out.print("Enter credits for subject " + (i + 1) + ": ");
            subject[i].credits = s.nextInt();
            subject[i].calculateGrade();
        }
        s.nextLine();
    }
    public void computeSGPA() {
        double totalPoints = 0;
        int totalCredits = 0;
```

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

        SGPA = (totalCredits == 0) ? 0 : totalPoints / totalCredits;
    }
    public void displayResults() {
        System.out.println("Student Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: %.2f%n"+ SGPA);
    }
}
public class Student {
    public static void main(String[] args) {
        Student1 s1 = new Student1();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResults();
    }
}
```

Output:

```
C:\1BM23CS138>javac Student.java

C:\1BM23CS138>java Student
Enter student name: KANISHKA SHARMA
Enter student USN: 1BM23CS138
Enter marks for subject 1: 92
Enter credits for subject 1: 4
Enter marks for subject 2: 91
Enter credits for subject 2: 4
Enter marks for subject 3: 76
Enter credits for subject 3: 3
Enter marks for subject 4: 84
Enter credits for subject 4: 3
Enter marks for subject 5: 76
Enter credits for subject 5: 3
Enter marks for subject 6: 95
Enter credits for subject 6: 1
Enter marks for subject 7: 82
Enter credits for subject 7: 1
Enter marks for subject 8: 94
Enter credits for subject 8: 1
Student Name: KANISHKA SHARMA
USN: 1BM23CS138
SGPA: 9.20
```

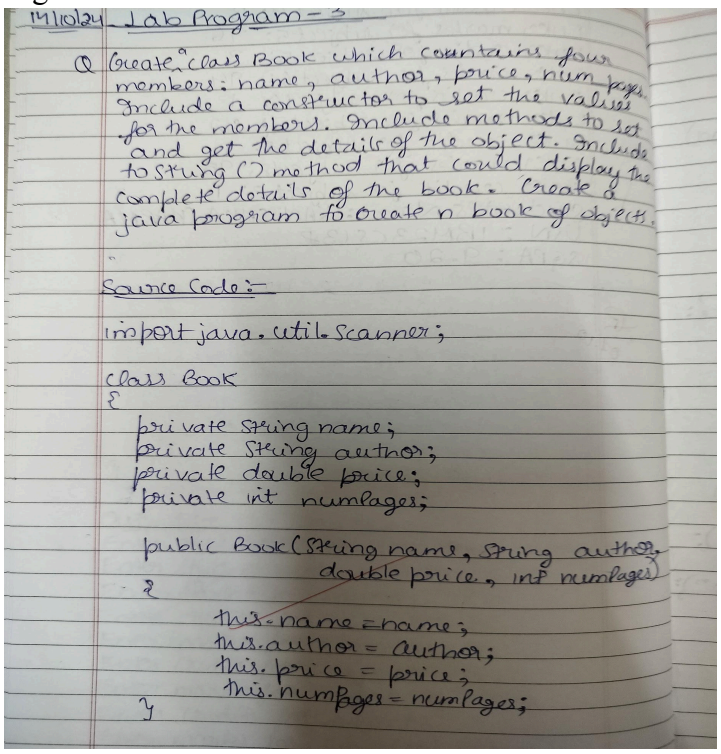
```
C:\1BM23CS138>javac Student.java

C:\1BM23CS138>java Student
Enter student name: CHITRASHREE K
Enter student USN: 1BM23CS081
Enter marks for subject 1: 87
Enter credits for subject 1: 4
Enter marks for subject 2: 98
Enter credits for subject 2: 4
Enter marks for subject 3: 95
Enter credits for subject 3: 3
Enter marks for subject 4: 86
Enter credits for subject 4: 3
Enter marks for subject 5: 91
Enter credits for subject 5: 3
Enter marks for subject 6: 93
Enter credits for subject 6: 1
Enter marks for subject 7: 95
Enter credits for subject 7: 1
Enter marks for subject 8: 87
Enter credits for subject 8: 1
Student Name: CHITRASHREE K
USN: 1BM23CS081
SGPA: 9.60
```

Program 3

Book Details using toString()

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

Algorithm:

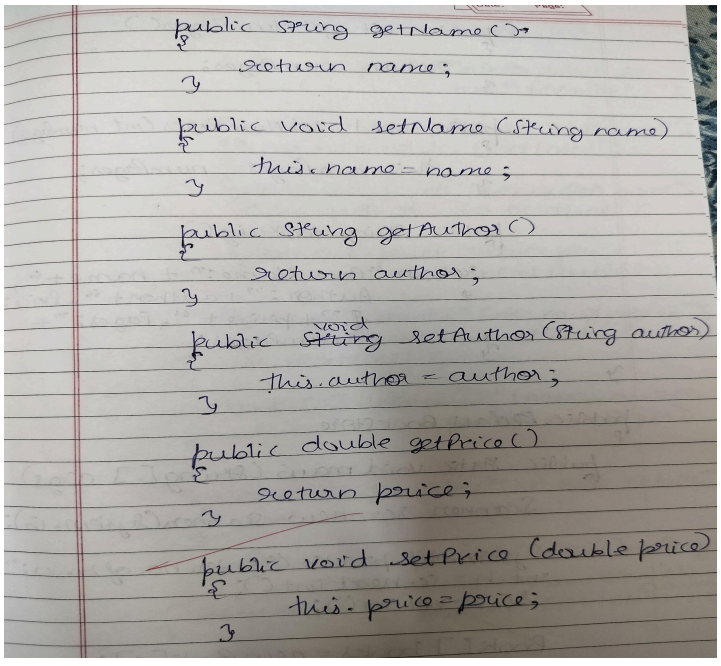


14/10/24 Lab Program - 3

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

Source Code:-

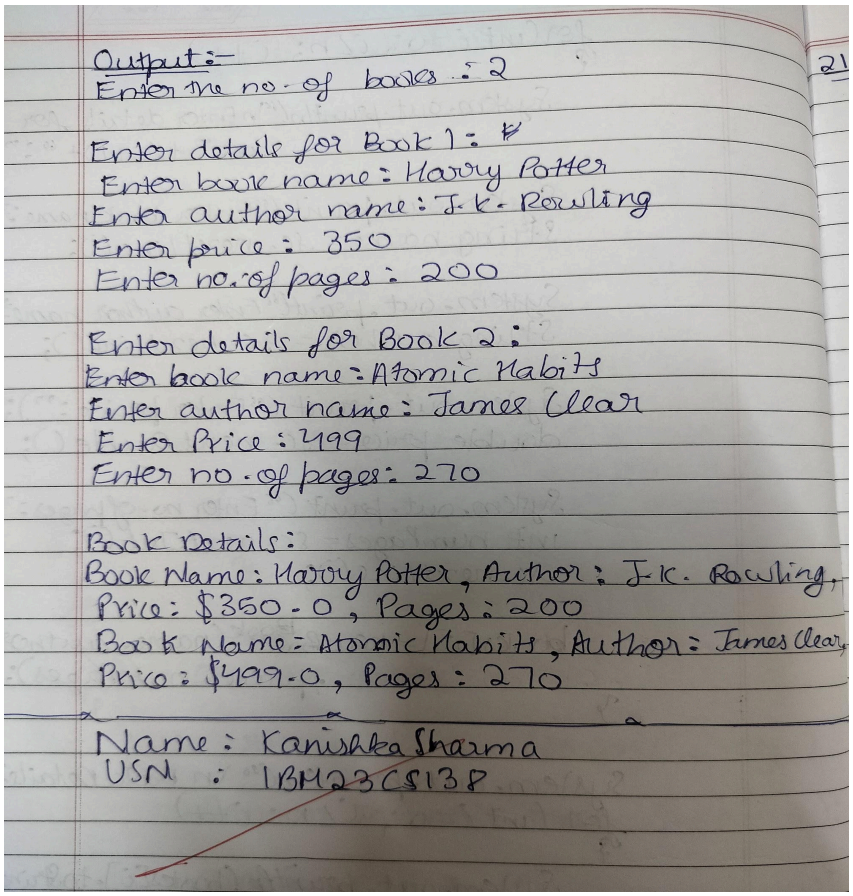
```
import java.util.Scanner;

class Book
{
    private String name;
    private String author;
    private double price;
    private int numPages;

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



```
    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

    public String getAuthor()
    {
        return author;
    }

    public void setAuthor(String author)
    {
        this.author = author;
    }

    public double getPrice()
    {
        return price;
    }

    public void setPrice(double price)
    {
        this.price = price;
    }
```

```
public int getNumPages()
{
    return numPages;
}

public void setNumPages(int numPages)
{
    this.numPages = numPages;
}

public String toString()
{
    return "Book Name:" + name + "
    Author:" + author + ", Price:
    $" + price + ", Pages:" +
    numPages;
}
}

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

        System.out.println("Enter no. of books:");
        int n = sc.nextInt();
        sc.nextLine();

        Book[] books = new Book[n];
```

```
        for(int i=0; i<n; i++)
        {
            System.out.println("\n Enter details for
                book" + (i+1) + ":");

            System.out.print("Enter book name:");
            String name = sc.nextLine();

            System.out.print("Enter author name:");
            String author = sc.nextLine();

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

            System.out.print("Enter no. of pages:");
            int numPages = sc.nextInt();
            sc.nextLine();

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

        System.out.println("\n Book Details:");
        for(int i=0; i<n; i++)
        {
            System.out.println(books[i].toString());
        }
        System.out.println("Name: Kanishka Sharma");
        sc.close();
        System.out.println("USN: 1BM23CS138");
    }
}
```



Code:

```
import java.util.Scanner;

class Book {

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


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


    public String getName() {
        return name;
    }
```

```
    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public int getNumPages() {
        return numPages;
    }

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }

    public String toString() {
        return "Book Name: " + name + ", Author: " + author + ", Price: $" + price + ", Pages: " +
numPages;
    }
}

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

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

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
```

```
            System.out.print("Enter book name: ");
            String name = sc.nextLine();

            System.out.print("Enter author name: ");
            String author = sc.nextLine();

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

            System.out.print("Enter number of pages: ");
            int numPages = sc.nextInt();
            sc.nextLine(); // Consume the newline

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

        System.out.println("\nBook Details:");
        for (int i = 0; i < n; i++) {
            System.out.println(books[i].toString());
        }

        System.out.println("Name : Kanishka Sharma");
        System.out.println("USN : 1BM23CS138");


        sc.close();
    }
}
```

Output:

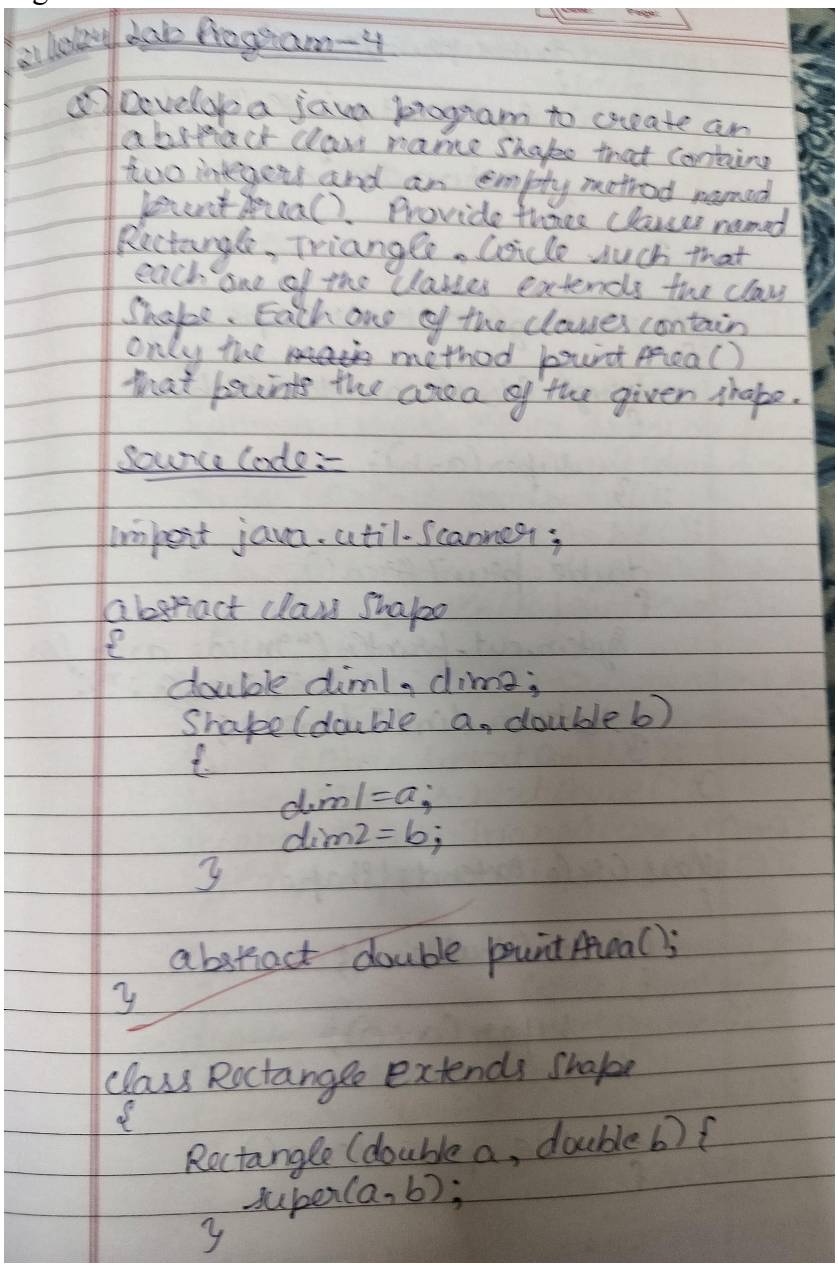
```
D:\1BM23CS138>javac BookStore.java

D:\1BM23CS138>java BookStore
Enter the number of books: 2

Enter details for Book 1:
Enter book name:  Harry Potter
Enter author name: J.K  Rowling
Enter price: 350
Enter number of pages: 200

Enter details for Book 2:
Enter book name: Atomic Habits
Enter author name: James Clear
Enter price: 499
Enter number of pages: 270

Book Details:
Book Name:  Harry Potter, Author: J.K  Rowling, Price: $350.0, Pages: 200
Book Name: Atomic Habits, Author: James Clear, Price: $499.0, Pages: 270
Name : Kanishka Sharma
USN : 1BM23CS138
```

Program 4

Shape and areas (Abstract Classes)

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

Algorithm:



Lab Program-4

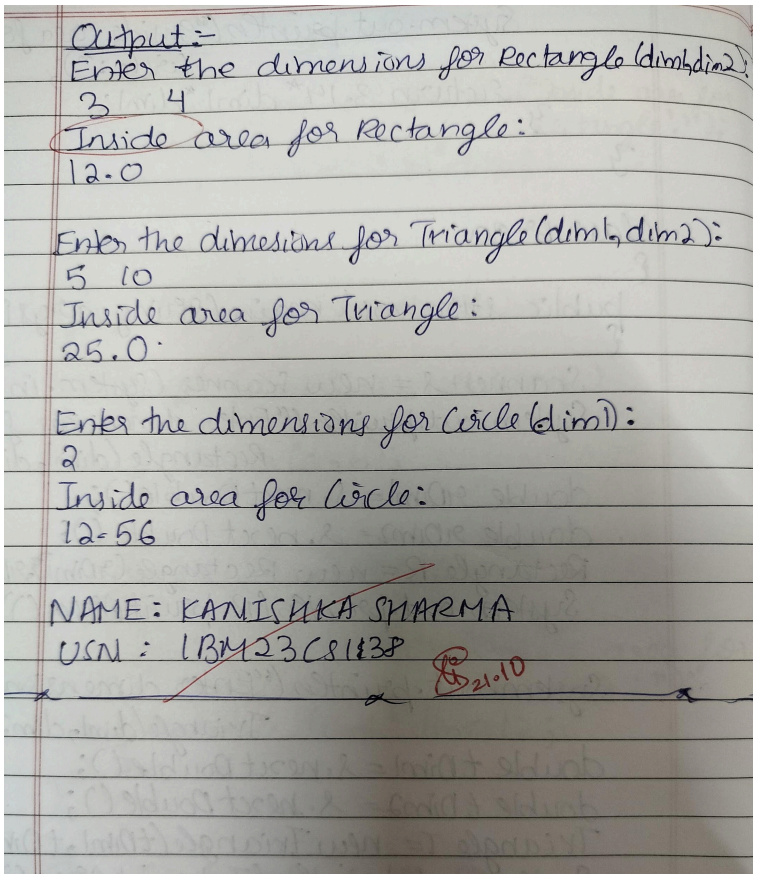
8) Develop a java program to create an abstract class name Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle, Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the ~~main~~ method printArea() that prints the area of the given shape.

Source Code:-

```
import java.util.Scanner;

abstract class Shape
{
    double dim1, dim2;
    Shape(double a, double b)
    {
        dim1=a;
        dim2=b;
    }

    abstract double printArea();
}

class Rectangle extends Shape
{
    Rectangle(double a, double b){
        super(a,b);
    }
```

```
    double printArea()
    {
        System.out.println("Inside area for Rectangle:");
        return dim1*dim2;
    }
}

class Triangle extends Shape
{
    Triangle(double a, double b)
    {
        super(a,b);
    }

    double printArea()
    {
        System.out.println("Inside area for Triangle:");
        return dim1*dim2/2;
    }
}

class Circle extends Shape
{
    Circle(double a)
    {
        super(a,0);
    }

    double printArea()
    {
```

```
        System.out.println("Inside area for Circle:");
        return 3.14*dim1*dim1;
    }
}

class Area
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in)
        System.out.println("Enter dimensions for Rectangle(dim1,dim2):");
        double rDim1 = s.nextDouble();
        double rDim2 = s.nextDouble();
        Rectangle R = new Rectangle(rDim1,rDim2);
        System.out.println(R.printArea());

        System.out.println("Enter dimensions for Triangle(dim1,dim2):");
        double tDim1 = s.nextDouble();
        double tDim2 = s.nextDouble();
        Triangle T = new Triangle(tDim1,tDim2);
        System.out.println(T.printArea());

        System.out.println("Enter dimensions for Circle(dim1):");
        double cDim1 = s.nextDouble();
        Circle C = new Circle(cDim1);
        System.out.println(C.printArea());

        System.out.println("NAME: KANISHKA SHARMA");
        System.out.println("USN: 1BM23CS138");
    }
}
```


Output:-
Enter the dimensions for Rectangle (dim1,dim2):
3 4
Inside area for Rectangle:
12.0

Enter the dimesions for Triangle (dim1, dim2):
5 10
Inside area for Triangle:
25.0

Enter the dimensions for Circle (dim1):
2
Inside area for Circle:
12.56

NAME: KANISHKA SHARMA
USN : 1BM23CS138
21.10

Code:

```
import java.util.Scanner;

abstract class Shape
{
double dim1,dim2;

Shape(double a,double b)
{
dim1=a;
dim2=b;
}

abstract double printArea();
}

class Rectangle extends Shape
{
Rectangle (double a,double b)
{
super(a,b);
}
```

```
double printArea()
{
System.out.println("Inside area for Rectangle :");
return dim1*dim2;
}
}

class Triangle extends Shape
{
Triangle (double a,double b)
{
super(a,b);
}

double printArea()
{
System.out.println("Inside area for Triangle :");
return dim1*dim2/2;
}
}
class Circle extends Shape
{
Circle (double a)
{
super(a,0);
}
double printArea()
{
System.out.println("Inside area for Circle :");
return 3.14*dim1*dim1;
}
}
class Area
{
public static void main(String args[])
{
Scanner s = new Scanner(System.in);
System.out.println("Enter the dimensions for Rectangle (dim1,dim2):");
double rDim1 = s.nextDouble();
double rDim2 = s.nextDouble();
Rectangle R = new Rectangle(rDim1, rDim2);
System.out.println(R.printArea());

System.out.println("Enter the dimensions for Triangle (dim1,dim2):");
double tDim1 = s.nextDouble();
double tDim2 = s.nextDouble();
Triangle T = new Triangle(tDim1, tDim2);
```

```
System.out.println(T.printArea());

System.out.println("Enter the dimension for Circle (dim1):");
double cDim1 = s.nextDouble();
Circle C = new Circle(cDim1);
System.out.println(C.printArea());

System.out.println("NAME : KANISHKA SHARMA");
System.out.println("USN : 1BM23CS138");
}
}
```

Output:

```
D:\>cd "1BM23CS138"

D:\1BM23CS138>javac Area.java

D:\1BM23CS138>java Area
Enter the dimensions for Rectangle (dim1,dim2):
3 4
Inside area for Rectangle :
12.0
Enter the dimensions for Triangle (dim1,dim2):
5 10
Inside area for Triangle :
25.0
Enter the dimension for Circle (dim1):
2
Inside area for Circle :
12.56
NAME : KANISHKA SHARMA
USN : 1BM23CS138
```

Program 5

Bank Inheritance

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

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

a)Accept deposit from customer and update the balance.

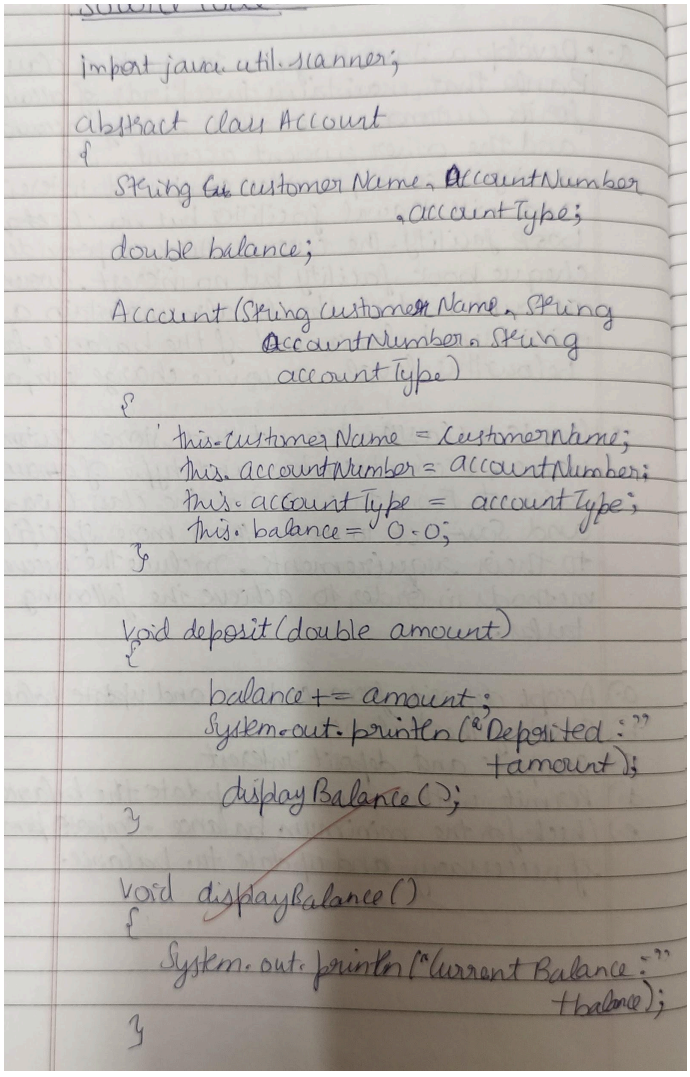
b)Display the balance.

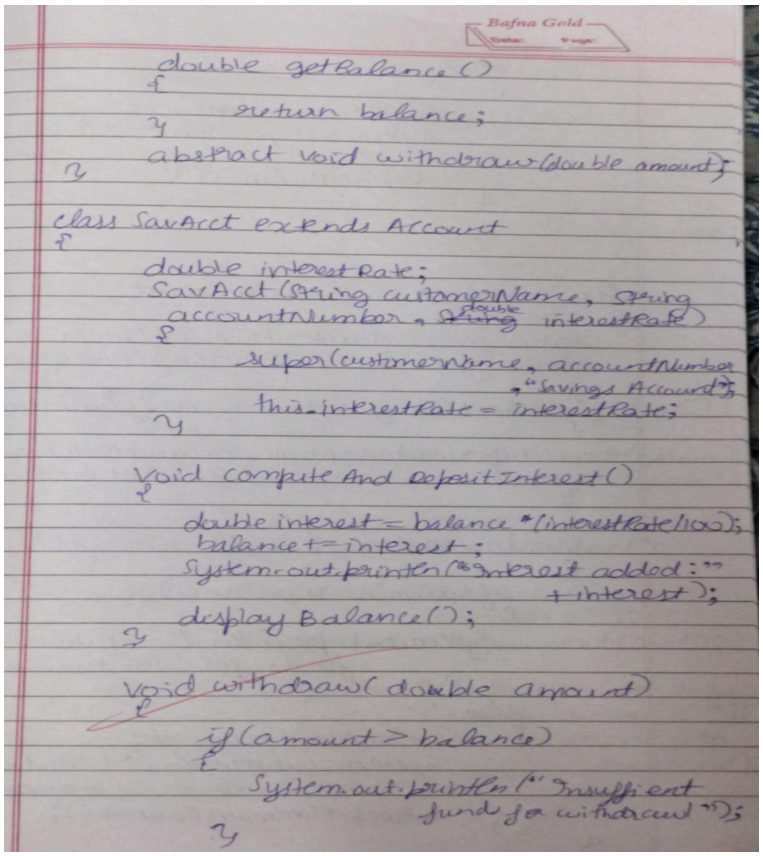
c)Compute and deposit interest

d)Permit withdrawal and update the balance

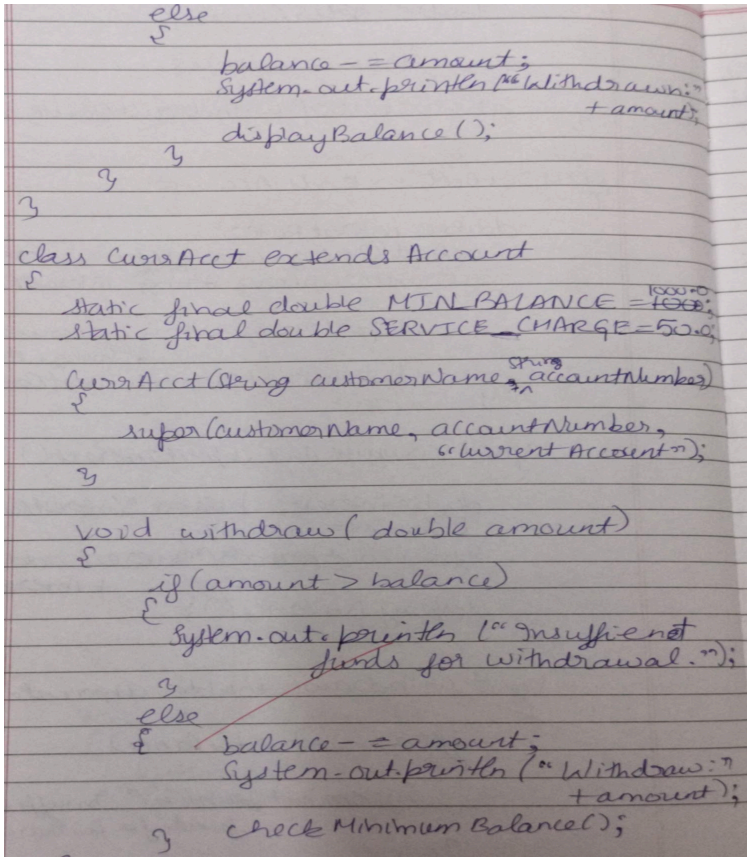
e) Check for the minimum balance, impose penalty if necessary and update the balance.

Algorithm:

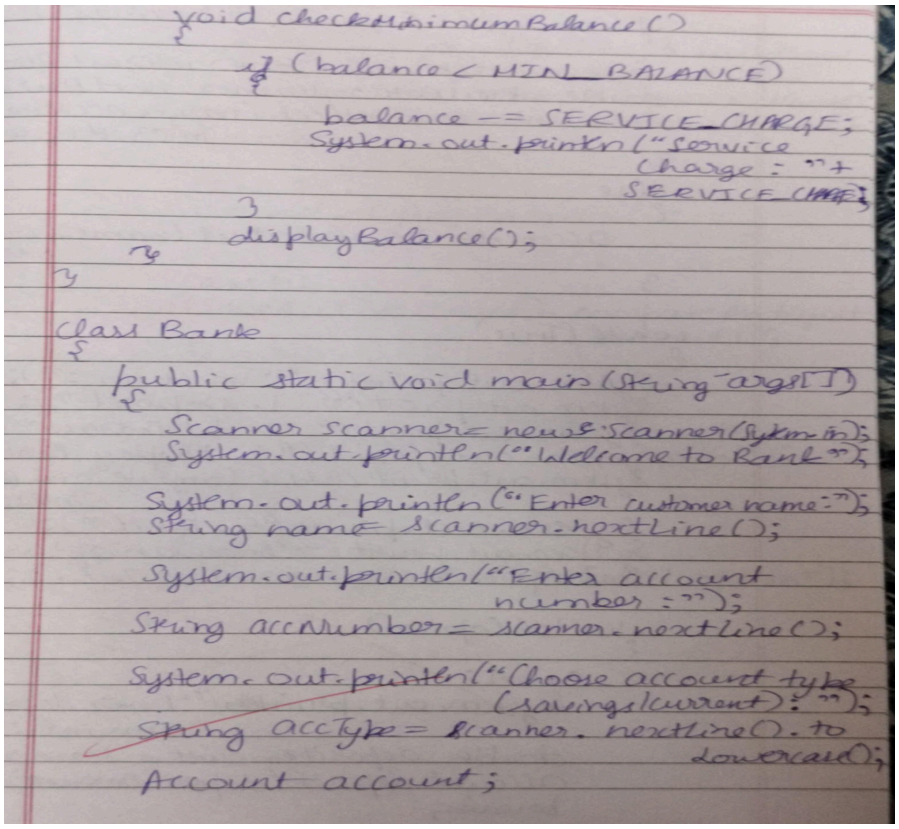




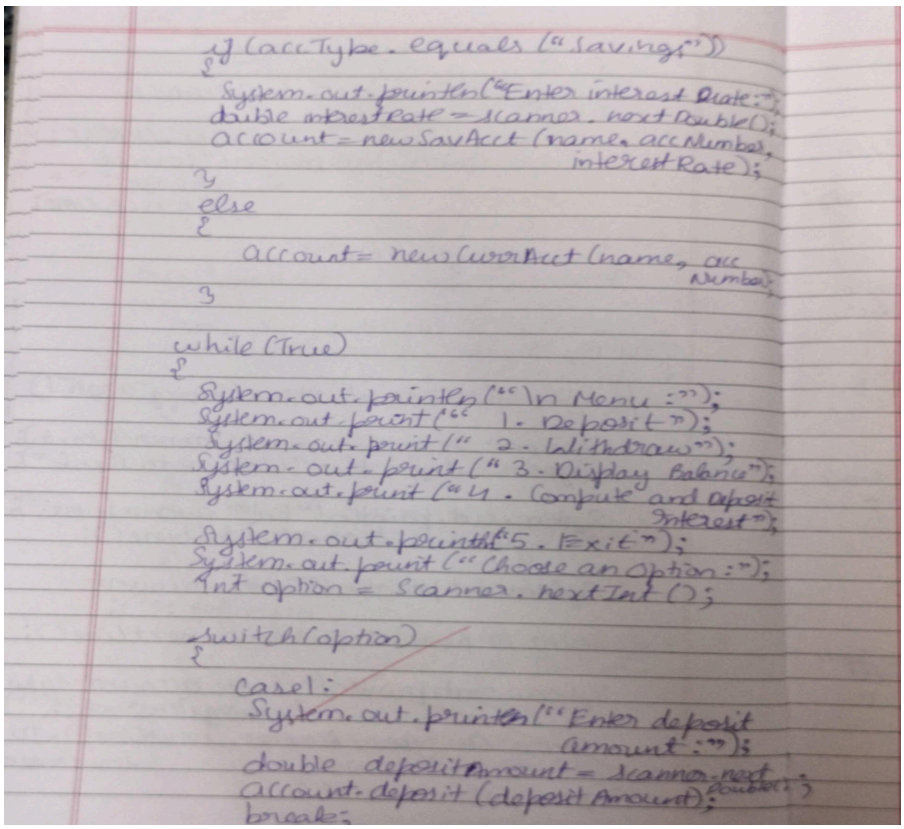
```
double getBalance()
{
    return balance;
}
abstract void withdraw(double amount);
}

class SavAcct extends Account
{
    double interestRate;
    SavAcct(String customerName, String
    accountNumber, double interestRate)
    {
        super(customerName, accountNumber
                ,"Savings Account");
        this.interestRate = interestRate;
    }

    Void computeAndDepositInterest()
    {
        double interest = balance *(interestRate/100);
        balance += interest;
        System.out.println("Interest added: "
                + interest);
        displayBalance();
    }

    Void withdraw(double amount)
    {
        if(amount > balance)
        {
            System.out.println("Insufficient
                funds for withdrawal");
        }
```



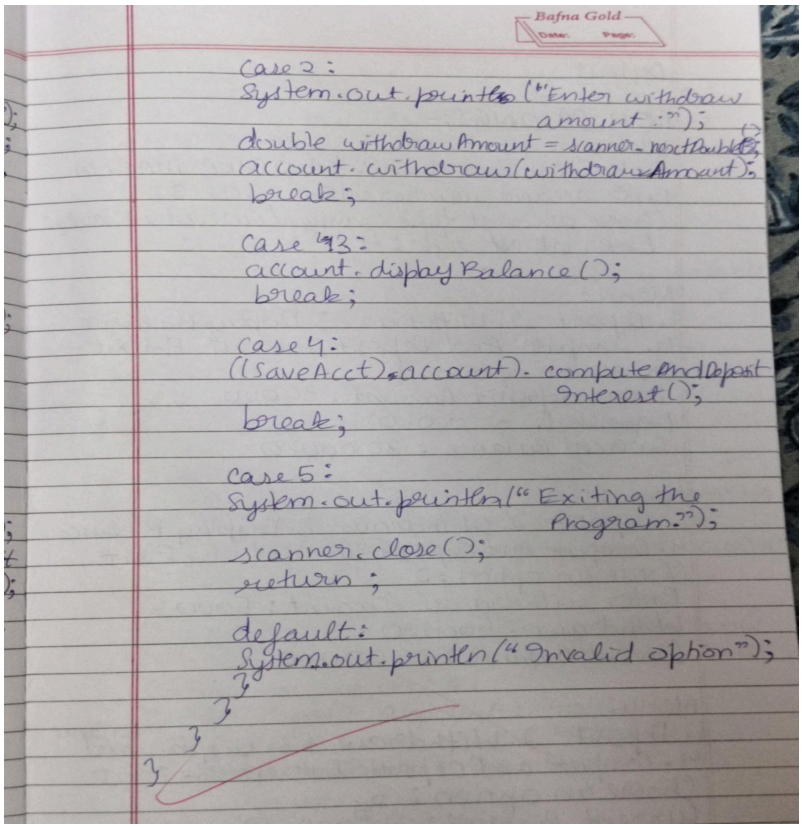
```
        else
        {
            balance -= amount;
            System.out.println("Withdrawn:"
                    + amount);
            displayBalance();
        }
    }
}

class CurrAcct extends Account
{
    Static final double MIN_BALANCE = 1000.0;
    static final double SERVICE_CHARGE = 50.0;

    CurrAcct(String customerName, String accountNumber)
    {
        super(customerName, accountNumber,
                "Current Account");
    }

    Void withdraw(double amount)
    {
        if(amount > balance)
        {
            System.out.println("Insufficient
                funds for withdrawal.");
        }
        else
        {
            balance -= amount;
            System.out.println("Withdraw:"
                    + amount);
            checkMinimumBalance();
        }
```



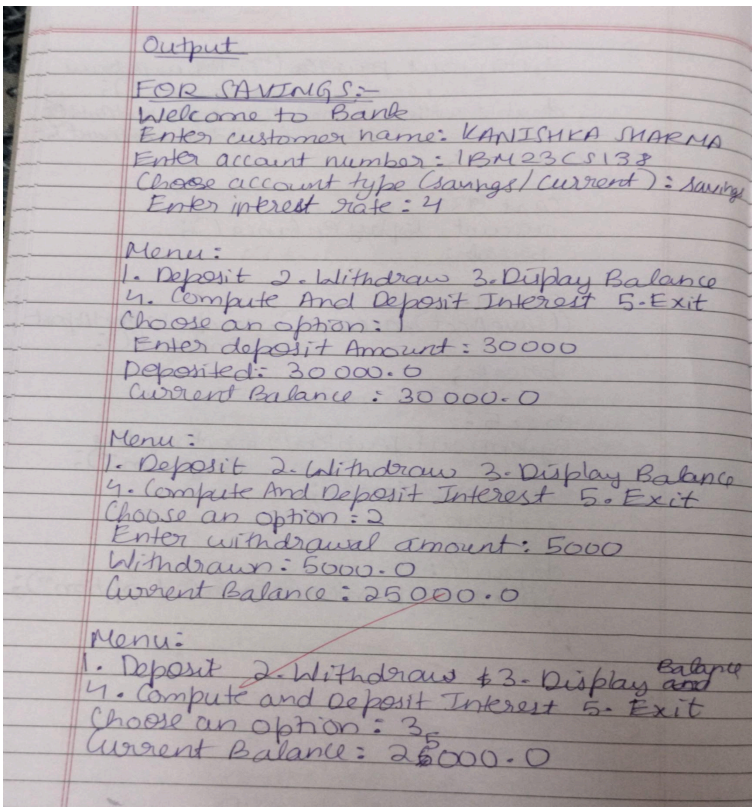
```
void checkMinimumBalance()
{
    if (balance < MIN_BALANCE)
    {
        balance -= SERVICE_CHARGE;
        System.out.println("Service
                Charge : " +
                SERVICE_CHARGE);
    }
    displayBalance();
}
}

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

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

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

        System.out.println("Choose account type
                (savings/current) :");
        String accType = scanner.nextLine().to
                LowerCase();
        Account account;
```



```
        if (accType.equals("savings"))
        {
            System.out.println("Enter interest Rate:");
            double interestRate = scanner.nextDouble();
            account = new SavAcct(name, accNumber,
                    interestRate);
        }
        else
        {
            account = new CurrAcct(name, acc
                    Number);
        }

        while (True)
        {
            System.out.println("\n Menu :");
            System.out.print(" 1. Deposit");
            System.out.print(" 2. Withdraw");
            System.out.print(" 3. Display Balance");
            System.out.print(" 4. Compute and deposit
                    interest");
            System.out.println(" 5. Exit");
            System.out.print("Choose an Option :");
            int option = scanner.nextInt();

            switch(option)
            {
                case 1:
                System.out.println("Enter deposit
                        Amount :");
                double depositAmount = scanner.next
                        Double();
                account.deposit(depositAmount);
                break;
```



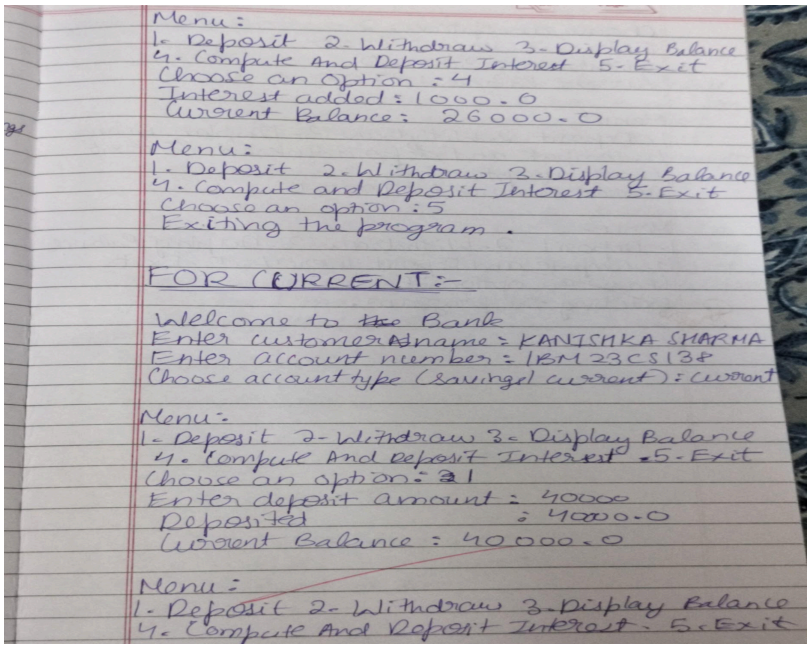
```
case 2:
System.out.println("Enter withdraw amount:");
double withdrawAmount = scanner.nextDouble();
account.withdraw(withdrawAmount);
break;

case 3:
account.displayBalance();
break;

case 4:
((SaveAcct)account).computeAndDepositInterest();
break;

case 5:
System.out.println("Exiting the Program:");
scanner.close();
return;

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



Output

FOR SAVINGS:-

```
Welcome to Bank
Enter customer name: KANISHKA SHARMA
Enter account number: IBM23CS138
Choose account type (Savings/Current): Savings
Enter interest rate: 4

Menu:
1. Deposit 2. Withdraw 3. Display Balance
4. Compute And Deposit Interest 5. Exit
Choose an option:
Enter deposit Amount: 30000
Deposited: 30000.0
Current Balance: 30000.0

Menu:
1. Deposit 2. Withdraw 3. Display Balance
4. Compute And Deposit Interest 5. Exit
Choose an option: 2
Enter withdrawal amount: 5000
Withdrawn: 5000.0
Current Balance: 25000.0

Menu:
1. Deposit 2. Withdraw & 3. Display Balance and
4. Compute and Deposit Interest 5. Exit
Choose an option: 3
Current Balance: 25000.0
```


Menu:
1. Deposit 2. Withdraw 3. Display Balance
4. Compute And Deposit Interest 5. Exit
Choose an option : 4
Interest added : 1000.0
Current Balance : 26000.0

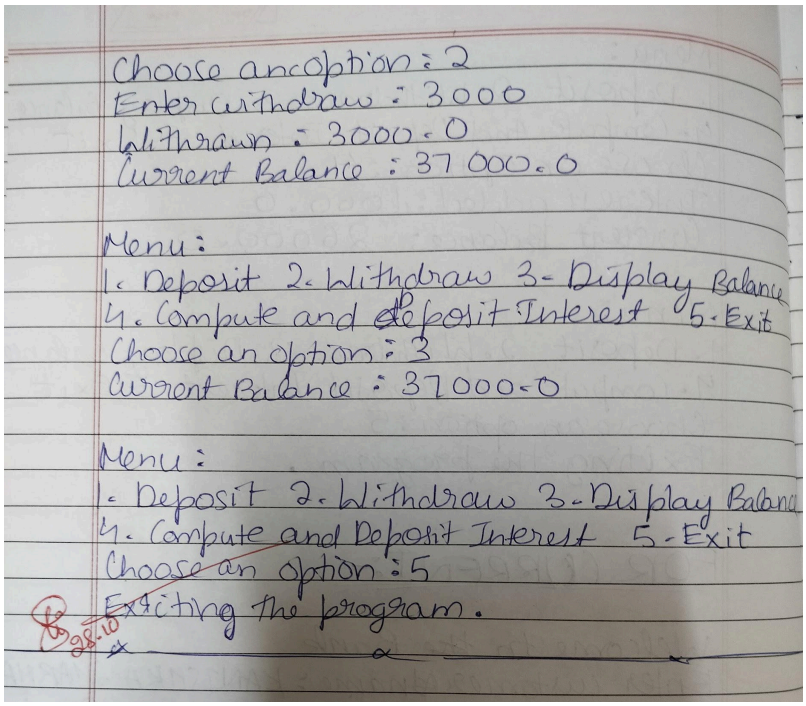
Menu:
1. Deposit 2. Withdraw 3. Display Balance
4. Compute and Deposit Interest 5. Exit
Choose an option : 5
Exiting the program.

FOR CURRENT:-

Welcome to the Bank
Enter customer name : KANISHKA SHARMA
Enter account number : IBM23CS138
Choose account type (Saving/current) : current

Menu:
1. Deposit 2. Withdraw 3. Display Balance
4. Compute And Deposit Interest 5. Exit
Choose an option : 1
Enter deposit amount : 40000
Deposited : 40000.0
Current Balance : 40000.0

Menu:
1. Deposit 2. Withdraw 3. Display Balance
4. Compute And Deposit Interest 5. Exit


Choose an option : 2
Enter withdraw : 3000
Withdrawn : 3000.0
Current Balance : 37000.0

Menu:
1. Deposit 2. Withdraw 3. Display Balance
4. Compute and Deposit Interest 5. Exit
Choose an option : 3
Current Balance : 37000.0

Menu:
1. Deposit 2. Withdraw 3. Display Balance
4. Compute and Deposit Interest 5. Exit
Choose an option : 5
Exiting the program.

Code:

```
import java.util.Scanner;

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

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

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

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

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

    double getBalance() {
        return balance;
    }

    abstract void withdraw(double amount);
}

class SavAcct extends Account {
    double interestRate;

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

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

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds for withdrawal.");
        } else {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
            displayBalance();
        }
    }
}
```

```
class CurAcct extends Account {
   static final double MIN_BALANCE = 1000.0;
   static final double SERVICE_CHARGE = 50.0;

    CurAcct(String customerName, String accountNumber) {
       super(customerName, accountNumber, "Current Account");
    }

    void withdraw(double amount) {
       if (amount > balance) {
          System.out.println("Insufficient funds for withdrawal.");
       } else {
          balance -= amount;
          System.out.println("Withdrawn: " + amount);
          checkMinimumBalance();
       }
    }

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

class Bank {
    public static void main(String[] args) {
       Scanner scanner = new Scanner(System.in);
       System.out.println("Welcome to the Bank!");

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

       System.out.print("Enter account number: ");
       String accNumber = scanner.nextLine();

       System.out.print("Choose account type (savings/current): ");
       String accType = scanner.nextLine().toLowerCase();

       Account account;
       if (accType.equals("savings")) {
          System.out.print("Enter interest rate: ");
          double interestRate = scanner.nextDouble();
          account = new SavAcct(name, accNumber, interestRate);
```

```
        } else {
            account = new CurAcct(name, accNumber);
        }

        while (true) {
            System.out.println("\nMenu:");
            System.out.print(" 1. Deposit");
            System.out.print(" 2. Withdraw");
            System.out.print(" 3. Display Balance");
            System.out.print(" 4. Compute and Deposit Interest");
            System.out.println(" 5. Exit");
            System.out.print("Choose an option: ");
            int option = scanner.nextInt();

            switch (option) {
                case 1:
                    System.out.print("Enter deposit amount: ");
                    double depositAmount = scanner.nextDouble();
                    account.deposit(depositAmount);
                    break;
                case 2:
                    System.out.print("Enter withdrawal amount: ");
                    double withdrawAmount = scanner.nextDouble();
                    account.withdraw(withdrawAmount);
                    break;
                case 3:
                    account.displayBalance();
                    break;
                case 4:
                        ((SavAcct) account).computeAndDepositInterest();
                    break;
                case 5:
                    System.out.println("Exiting the program.");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid option. Please try again.");
            }
        }
    }
}
```

Output:

```
D:\1BM23CS138>javac Bank.java

D:\1BM23CS138>java Bank
Welcome to the Bank!
Enter customer name: KANISHKA SHARMA
Enter account number: 1BM23CS138
Choose account type (savings/current): current

Menu:
 1. Deposit 2. Withdraw 3. Display Balance 4. Compute and Deposit Interest 5. Exit
Choose an option: 1
Enter deposit amount: 40000
Deposited: 40000.0
Current Balance: 40000.0

Menu:
 1. Deposit 2. Withdraw 3. Display Balance 4. Compute and Deposit Interest 5. Exit
Choose an option: 2
Enter withdrawal amount: 3000
Withdrawn: 3000.0
Current Balance: 37000.0

Menu:
 1. Deposit 2. Withdraw 3. Display Balance 4. Compute and Deposit Interest 5. Exit
Choose an option: 3
Current Balance: 37000.0

Menu:
 1. Deposit 2. Withdraw 3. Display Balance 4. Compute and Deposit Interest 5. Exit
Choose an option: 5
Exiting the program.

D:\1BM23CS138>
```

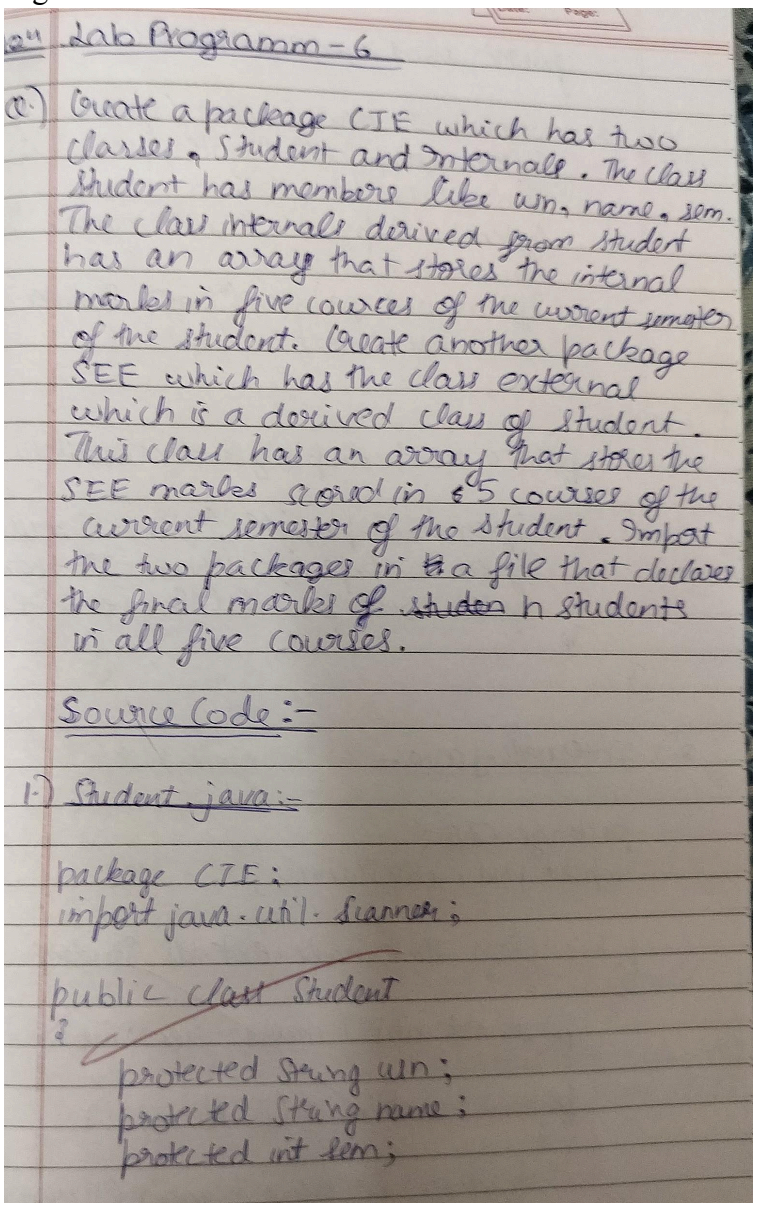
```
D:\1BM23CS138>javac Bank.java

D:\1BM23CS138>java Bank
Welcome to the Bank!
Enter customer name: KANISHKA SHARMA
Enter account number: 1BM23CS138
Choose account type (savings/current): savings
Enter interest rate: 4

Menu:
 1. Deposit 2. Withdraw 3. Display Balance 4. Compute and Deposit Interest 5. Exit
Choose an option: 1
Enter deposit amount: 30000
Deposited: 30000.0
Current Balance: 30000.0

Menu:
 1. Deposit 2. Withdraw 3. Display Balance 4. Compute and Deposit Interest 5. Exit
Choose an option: 2
Enter withdrawal amount: 5000
Withdrawn: 5000.0
Current Balance: 25000.0

Menu:
 1. Deposit 2. Withdraw 3. Display Balance 4. Compute and Deposit Interest 5. Exit
Choose an option: 3
Current Balance: 25000.0

Menu:
 1. Deposit 2. Withdraw 3. Display Balance 4. Compute and Deposit Interest 5. Exit
Choose an option: 4
Interest added: 1000.0
Current Balance: 26000.0

Menu:
 1. Deposit 2. Withdraw 3. Display Balance 4. Compute and Deposit Interest 5. Exit
Choose an option: 5
Exiting the program.
```

Program 6

Marks card of Student using Packages.

Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Algorithm:



Data Programm - 6

a) Create a package CIE which has two classes, Student and Internals. The class Student has members like usn, name, sem. The class internals derived from student has an array that stores the internal marks in five courses of the current semester of the student. Create another package SEE which has the class external which is a derived class of Student. This class has an array that stores the SEE marks scored in 5 courses of the current semester of the student. Import the two packages in a file that declares the final marks of ~~student~~ n students in all five courses.

Source Code :-

1.) 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.println("Enter USN : ");
    usn = s.nextLine();
    System.out.println("Enter name : ");
    name = s.nextLine();
    System.out.println("Enter Semester : ");
    sem = s.nextInt();
}

public void displayStudentDetails()
{
    System.out.println("USN : " + usn);
    System.out.println("Name : " + name);
    System.out.println("Semester : " + sem);
}
}
```

2.) 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.println("Enter marks for
                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]);
        }
    }
}
```

3.) External.java :-

```
package SEE;

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

public class External extends Internals
{
    protected int[] externalMarks = new int[5];
    protected int[] 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.println("Enter marks for
      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();
  display CIE marks();
  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) + 6" :"
                        + finalMarks[i]);
  }
}
```

4) Main.Java :-

```
import SEE.External;
import java.util.Scanner;

public class Main
{
  public static void main(String[] args)
  {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter no. of students:");
    int n = sc.nextInt();
    sc.nextLine();

    External[] students = new Externals[n];

    for(int i=0; i<n; i++)
    {
      students[i] = new External();
      System.out.println("Enter details for student
                          + (i+1));
      students[i].inputStudentDetails();
      students[i].inputCIEmarks();
      students[i].inputSEEmarks();
    }
```

```
        for (int i=0; i<n; i++)
        {
            students[i].calculateFinalMarks();
            students[i].displayFinalMarks();
        }
        sc.close();
    }
}
```

Output:-

```
Enter USN:
1BM23CS138
Enter Name:
Kanishka
Enter Semester:
3

Enter Internal marks for 5 courses:
Enter marks for course 1:
96
Enter marks for course 2:
96
Enter marks for course 3:
96
Enter marks for course 4:
96
Enter marks for course 5:
96
Enter external marks for 5 courses:
Enter marks for course 1:
100
Enter marks for course 2:
100
Enter marks for course 3:
100
Enter marks for course 4:
100
Enter marks for course 5:
100
USN: 1BM23CS138
Name: Kanishka
Semester: 3
Internal Marks:
Course 1: 96
Course 2: 96
Course 3: 96
Course 4: 96
Course 5: 96
External Marks:
Course 1: 100
Course 2: 100
Course 3: 100
Course 4: 100
Course 5: 100
Final Marks:
Course 1: 196
Course 2: 196
Course 3: 196
Course 4: 196
Course 5: 196
```

Code:

Main.java:

```
import SEE.External;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
        sc.nextLine();
        External[] students = new External[n];
        for (int i = 0; i < n; i++) {
            students[i] = new External();
            System.out.println("Enter details for student " + (i + 1));
            students[i].inputStudentDetails();
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
        }
        for (int i = 0; i < n; i++) {
            students[i].calculateFinalMarks();
            students[i].displayFinalMarks();
        }
        sc.close();
    }
}
```

CIE Package:

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.println("Enter marks for 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.println("Enter USN: ");
        usn = s.nextLine();
        System.out.println("Enter Name: ");
        name = s.nextLine();
        System.out.println("Enter Semester: ");
        sem = s.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}
```

SEE Package:

External.java:

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

public class External extends Internals {
    protected int[] externalMarks = new int[5];
    protected int[] 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.println("Enter marks for 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]);
        }
    }
}
```

Output:

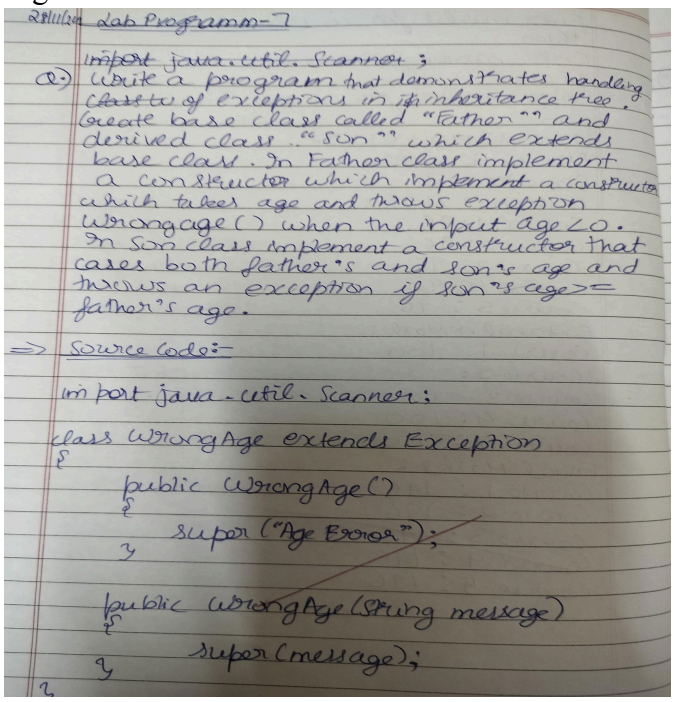
```
USN: 137
Name: Keerthi
Semester: 3
Internal Marks:
Course 1: 98
Course 2: 98
Course 3: 98
Course 4: 98
Course 5: 98
External Marks:
Course 1: 97
Course 2: 97
Course 3: 97
Course 4: 97
Course 5: 97
Final Marks:
Course 1: 195
Course 2: 195
Course 3: 195
Course 4: 195
Course 5: 195
USN: 138
Name: Kanishka
Semester: 3
Internal Marks:
Course 1: 96
Course 2: 96
Course 3: 96
Course 4: 96
Course 5: 96
External Marks:
Course 1: 100
Course 2: 100
Course 3: 100
Course 4: 100
Course 5: 100
Final Marks:
Course 1: 196
Course 2: 196
Course 3: 196
Course 4: 196
Course 5: 196

D:\1BM23CS138\CIE>
```

Program 7

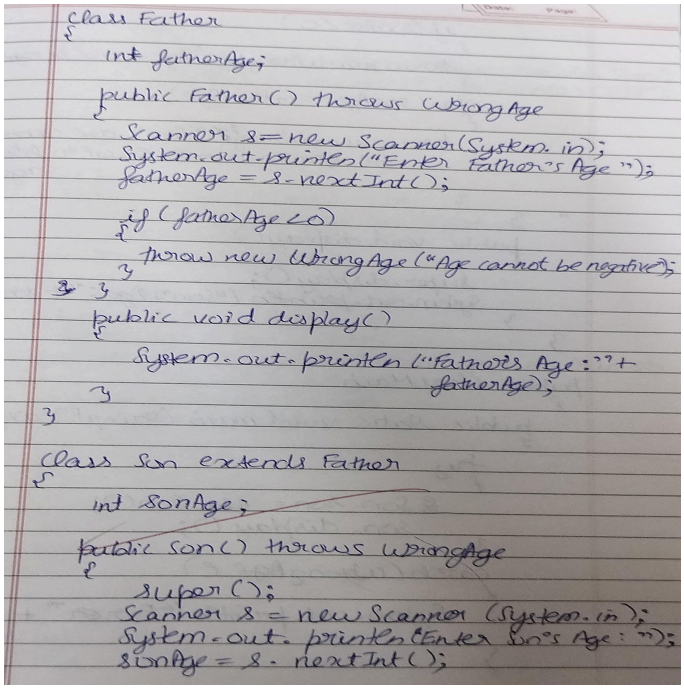
Father and Son age using Exceptions

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

Algorithm:



```
28/11/24 Lab Programm-7

import java.util.Scanner;
Q) Write a program that demonstrates handling
   classes of exceptions in an inheritance tree.
   Create base class called "Father" and
   derived class "Son" which extends
   base class. In Father class implement
   a constructor which implement a constructor
   which takes age and throws exception
   WrongAge() when the input age<0.
   In Son class implement a constructor that
   cases both father's and son's age and
   throws an exception if son's age>=
   father's age.
=> Source Code:-

import java.util.Scanner;

class WrongAge extends Exception
{
    public WrongAge()
    {
        super("Age Error");
    }

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



```
class Father
{
    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");
        }
    }

    public void display()
    {
        System.out.println("Father's Age :"+
                           fatherAge);
    }
}

class Son extends Father
{
    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");
        }
        else if (sonAge >= fatherAge)
        {
            throw new WrongAge("Son's age cannot
                be greater than or equal to father's
                age");
        }
    }
    public void display()
    {
        super.display();
        System.out.println("Son's Age :"+sonAge);
    }
}

public class Main
{
    public static void main(String[] args)
    {
        try
        {
            Son son = new Son();
            son.display();
        }
        catch(WrongAge e)
        {
            System.out.println("Error" + e.get
                                    Message());
        }
```

```
        finally
        {
            System.out.println("Name: Kanishka Sharma");
            System.out.println("USN: 1BM23CS138");
        }
    }
}
```

Output:-

Enter Father's Age : -45
Error : Age cannot be negative
Name : Kanishka Sharma
USN : 1BM23CS138

or

Enter Father's Age : 18
Enter Son's Age : 45
Error : Son's age cannot be greater than or equal to father's age.
Name: Kanishka Sharma
USN : 1BM23CS138

Code:

```
import java.util.Scanner;

class WrongAge extends Exception {
   public WrongAge() {
      super("Age Error");
   }

   public WrongAge(String message) {
      super(message);
   }
}
class Father {
   int fatherAge;

   public Father() throws WrongAge {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter Father's Age: ");
      fatherAge = s.nextInt();

      if (fatherAge < 0) {
         throw new WrongAge("Age cannot be negative");
      }
   }
   public void display() {
      System.out.println("Father's Age: " + fatherAge);
   }
}
class Son extends Father {
   int sonAge;

   public Son() throws WrongAge {

      super();

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


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

```
        super.display();
        System.out.println("Son's Age: " + sonAge);
    }
}
public class Main {
    public static void main(String[] args) {
        try {
             Son son = new Son();
            son.display();

        } catch (WrongAge e) {
         System.out.println("Error: " + e.getMessage());
        }
        finally
        {
            System.out.println("Name : Kanishka Sharma ");
            System.out.println("USN : 1BM23CS138 ");
        }
    }
}
```

Output:

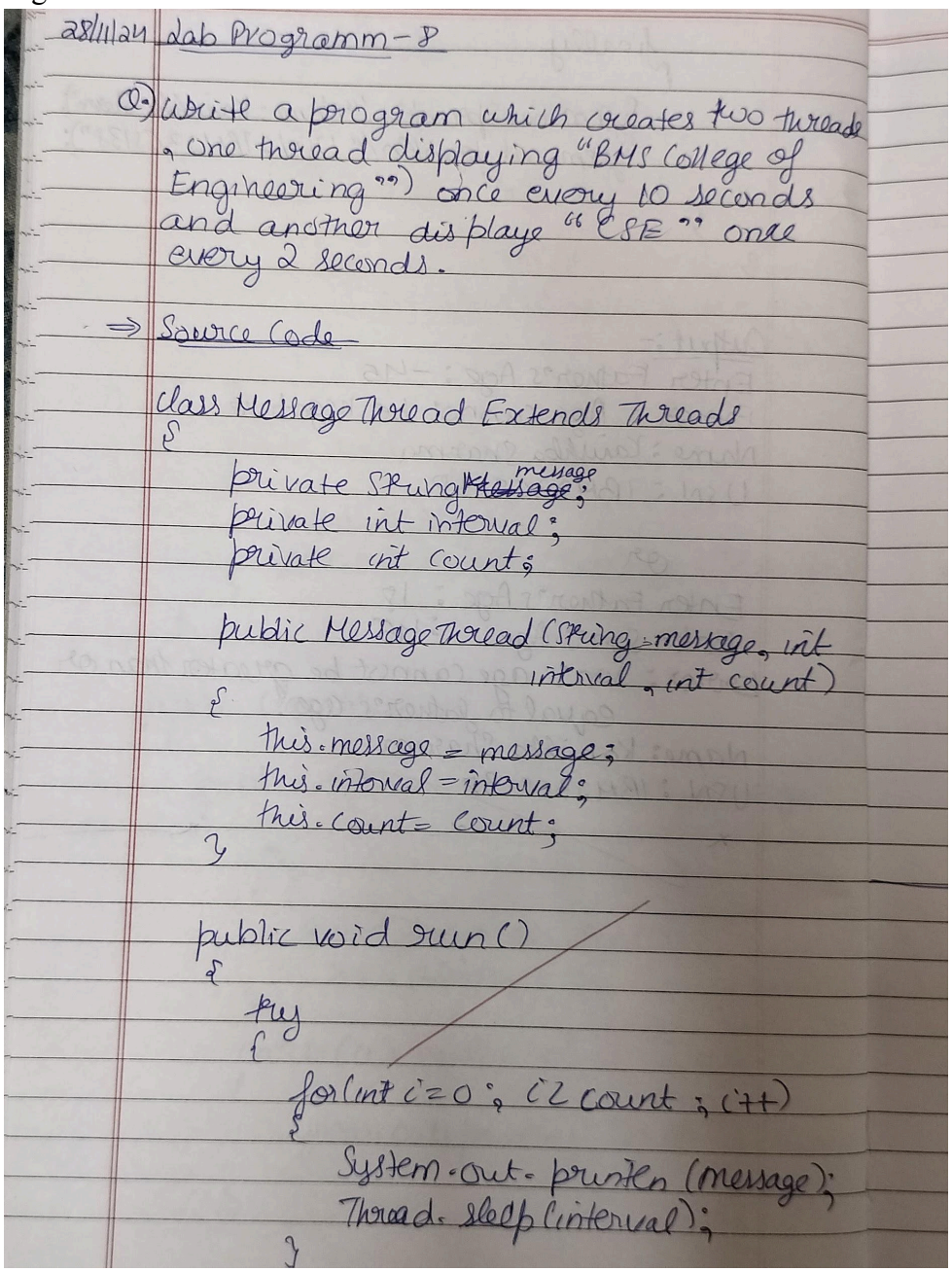
```
C:\1BM23CS138>javac Main.java

C:\1BM23CS138>java Main
Enter Father's Age: 45
Enter Son's Age: 18
Father's Age: 45
Son's Age: 18
Name : Kanishka Sharma
USN : 1BM23CS138

C:\1BM23CS138>javac Main.java

C:\1BM23CS138>java Main
Enter Father's Age: -45
Error: Age cannot be negative
Name : Kanishka Sharma
USN : 1BM23CS138

C:\1BM23CS138>javac Main.java

C:\1BM23CS138>java Main
Enter Father's Age: 18
Enter Son's Age: 45
Error: Son's age cannot be greater than or equal to father's age
Name : Kanishka Sharma
USN : 1BM23CS138
```

Program 8

Displaying college name and branch using Threads.

Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

Algorithm:



28/11/24 Lab Programm-8

Q. Write a program which creates two threads, one thread displaying "BMS College of Engineering") once every 10 seconds and another displaye "CSE" once every 2 seconds.

⇒ Source Code

```
class MessageThread Extends Threads
{
    private String message;
    private int interval;
    private int count;

    public MessageThread (String message, int
                          interval, int count)
    {
        this.message = message;
        this.interval = interval;
        this.count = count;
    }

    public void run()
    {
        try
        {
            for(int i=0; i<count; i++)
            {
                System.out.println(message);
                Thread.sleep(interval);
            }
```

```
        catch (InterruptedException e)
        {
            System.out.println(e);
        }
    }
}

public class Message2Threads
{
    public static void main(String[] args)
    {
        MessageThreads thread1 = new MessageThreads
                ("BMS College Of Engineering",
                10000, 5);
        MessageThreads thread2 = new MessageThreads
                ("CSE", 2000, 5);
        thread1.start();
        thread2.start();
        System.out.println("Name: Kanishka
                Sharma");
        System.out.println("USN: 1BM23CS138");
    }
}
```

Output:-

```
Name: Kanishka Sharma
USN: 1BM23CS138
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

```
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

Code:

```
class MessageThread extends Thread {
    private String message;
    private int interval;
    private int count;

    public MessageThread(String message, int interval, int count) {
        this.message = message;
        this.interval = interval;
        this.count = count;
    }
    public void run() {
        try {
            for (int i = 0; i < count; i++) {
                System.out.println(message);
                Thread.sleep(interval);
            }
        } catch (InterruptedException e) {
            System.out.println(e);
        }
    }
}
public class Message2Threads {
    public static void main(String[] args) {
        MessageThread thread1 = new MessageThread("BMS College of Engineering", 10000, 5); // 10
seconds, 5 times
        MessageThread thread2 = new MessageThread("CSE", 2000, 5); // 2 seconds, 5 times
        thread1.start();
        thread2.start();
        System.out.println("Name : Kanishka Sharma");
        System.out.println("USN : 1BM23CS138");

    }
}
```

Output:

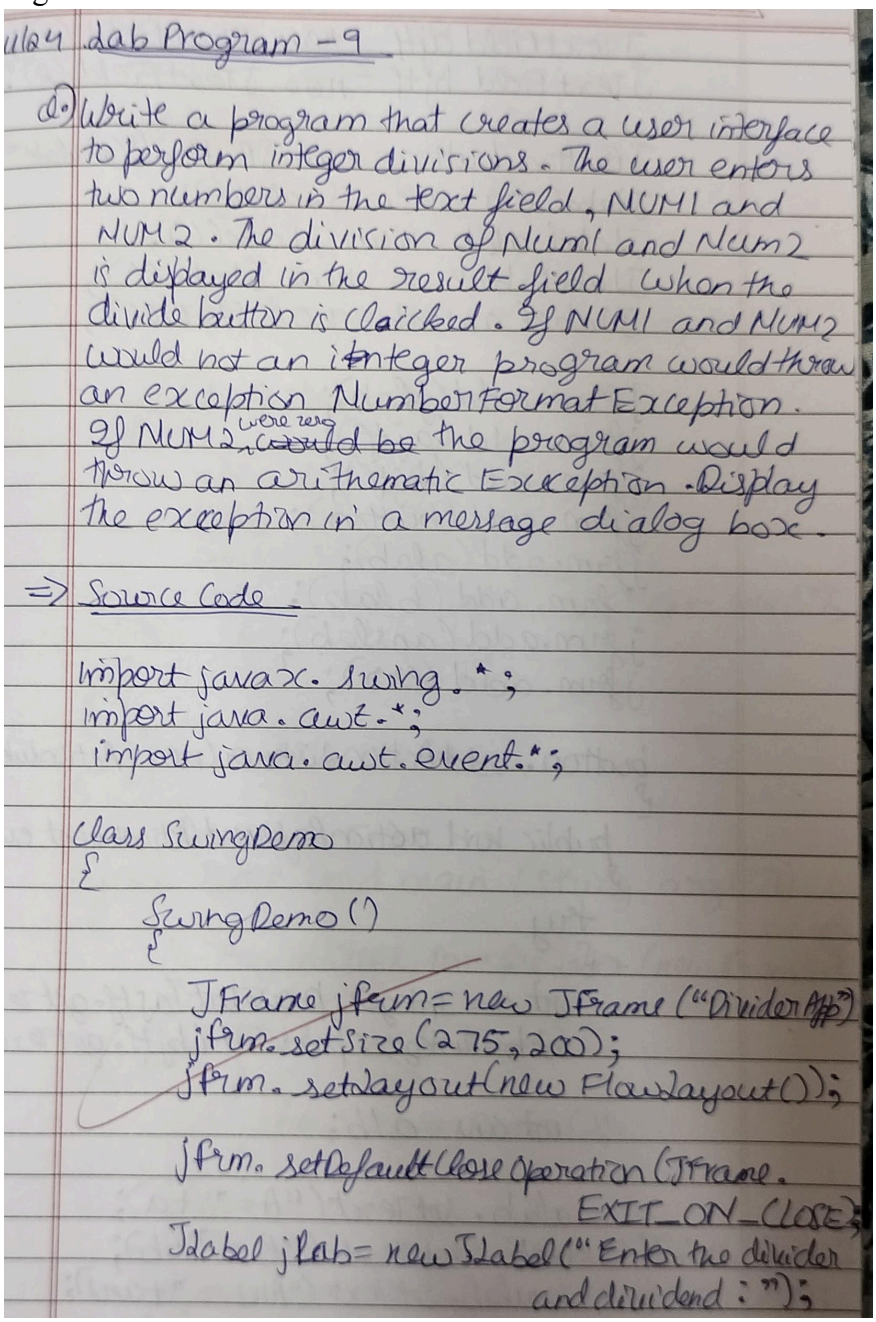
```
C:\1BM23CS138>java Message2Threads
Name : Kanishka Sharma
USN : 1BM23CS138
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

Program 9

Integer Divisions

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

Algorithm:



Lab Program - 9

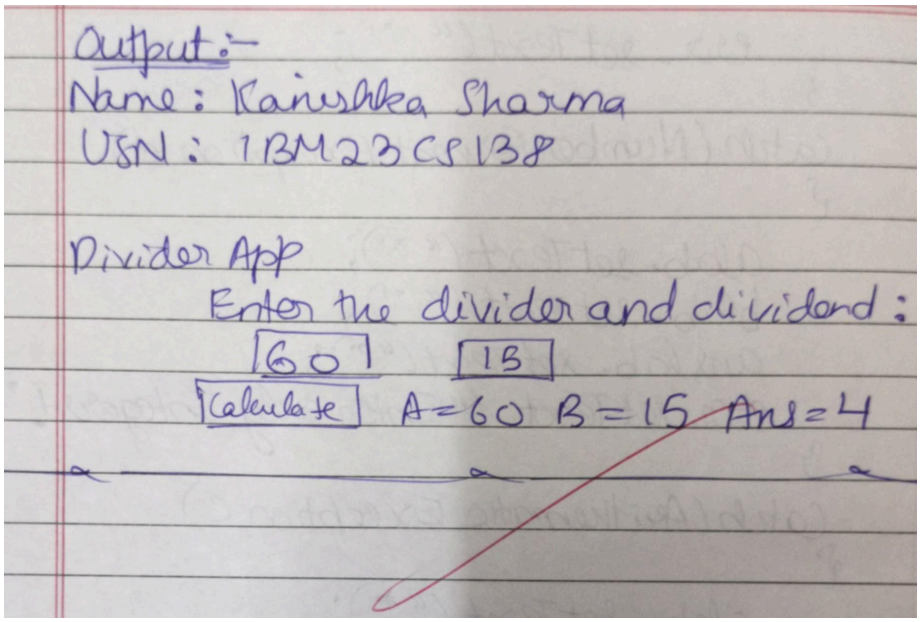
a) Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text field, NUM1 and NUM2. The division of Num1 and Num2 is displayed in the result field when the divide button is clicked. If NUM1 and NUM2 would not an integer program would throw an exception NumberFormatException. If NUM2 were zero the program would throw an arithematic Exception. Display the exception in a message dialog box.

=> Source Code

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

class SwingDemo
{
    SwingDemo()
    {
        JFrame jfrm = new JFrame("DividerApp");
        jfrm.setSize(275,200);
        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(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(ansLab);
jfrm.add(err);

button.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent ae)
  {
    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("");
    }
    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();
    }
  });
  System.out.println("Name: Kanishka Sharma");
  System.out.println("USN: 1BM23CS138");
}
}
```



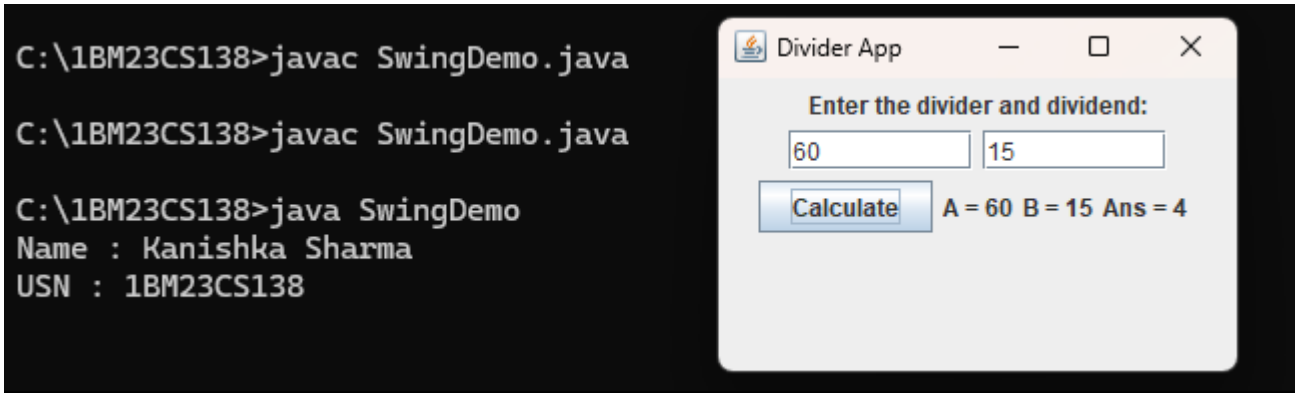
Code:

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

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

      // To terminate on close
      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      // Text label
      JLabel jlab = new JLabel("Enter the divider and dividend:");

      // Add text field for both numbers
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      // Calculate button
      JButton button = new JButton("Calculate");

      // Labels to display results and error messages
      JLabel err = new JLabel();
      JLabel alab = new JLabel();
      JLabel blab = new JLabel();
```

```
    JLabel anslab = new JLabel();

    // Add components to the frame
    jfrm.add(jlab);       // Label for the user input
    jfrm.add(ajtf);       // Text field for the first number (divider)
    jfrm.add(bjtf);       // Text field for the second number (dividend)
    jfrm.add(button);     // Calculate button
    jfrm.add(alab);       // Label to display the value of A (divider)
    jfrm.add(blab);       // Label to display the value of B (dividend)
    jfrm.add(anslab);     // Label to display the answer
    jfrm.add(err);        // Error message label

        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("");

            } catch (NumberFormatException e) {
                // Handling invalid input (non-integer values)
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("Enter Only Integers!");

            } catch (ArithmeticException e) {
                // Handling division by zero error
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("B should be NON-zero!");
            }
        }
    });
    jfrm.setVisible(true);
}

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

```
        }
    });
    System.out.println("Name : Kanishka Sharma");
    System.out.println("USN : 1BM23CS138");
  }
}
```

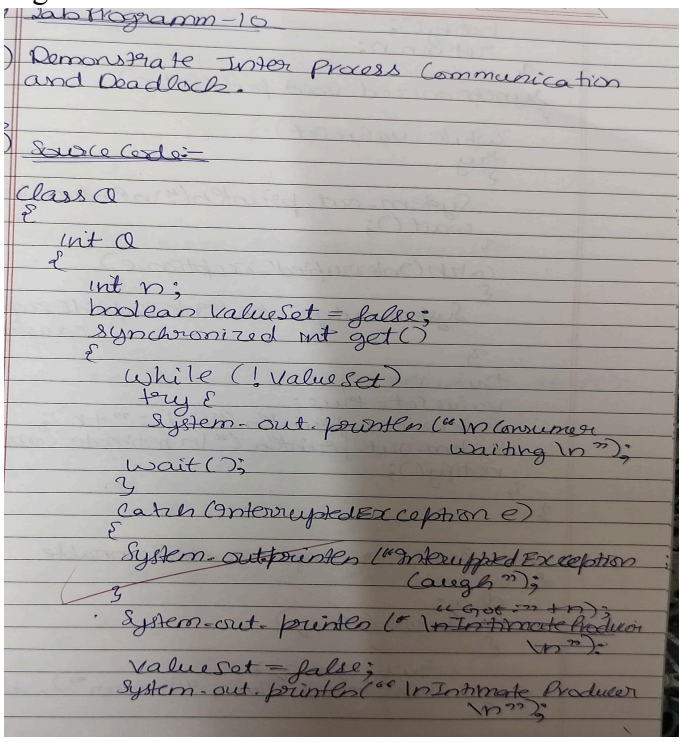
Output:



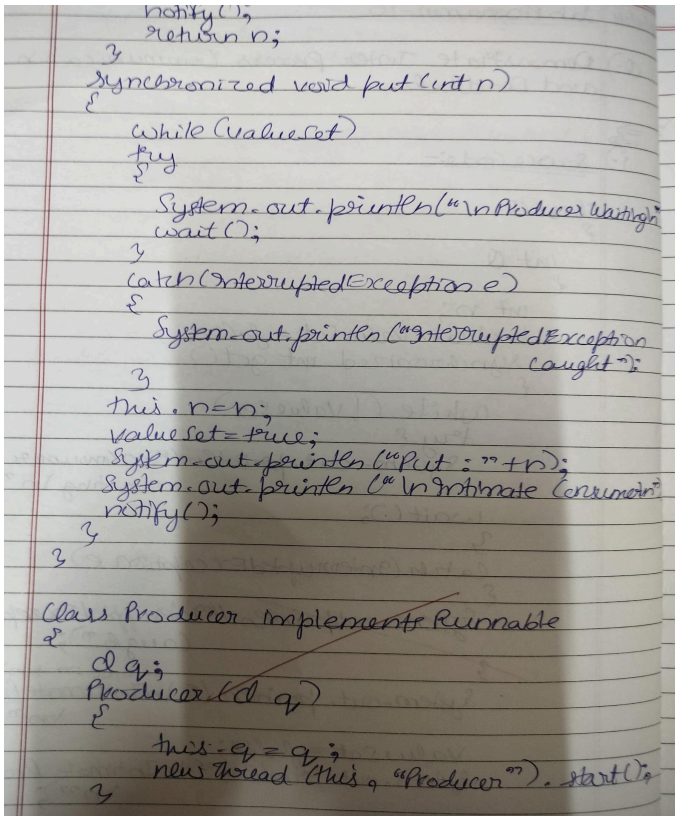
Program 10

Demonstrate Inter process Communication and Deadlock

Algorithm:



```
Lab Program-10

Demonstrate Inter Process Communication
and Deadlock.

Source Code:-

class Q
{
  int Q
  {
    int n;
    boolean valueSet = false;
    synchronized int get()
    {
      while (!valueSet)
      try {
        System.out.println("\n Consumer
                            waiting \n");
      wait();
      }
      catch (InterruptedException e)
      {
      System.out.println("InterruptedException
                          caught");
      }
      System.out.println("Got :" +n);
      valueSet = false;
      System.out.println("\n Intimate Producer
                          \n");
```



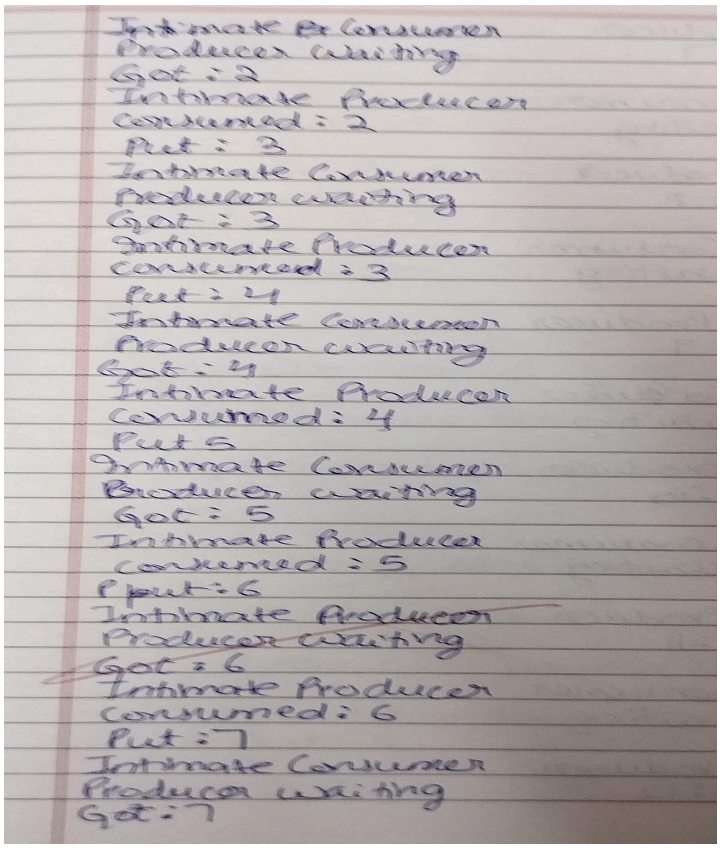
```
      notify();
      return n;
    }
    synchronized void put(int n)
    {
      while(valueSet)
      try
      {
        System.out.println("\n Producer Waiting");
        wait();
      }
      catch(InterruptedException e)
      {
        System.out.println("InterruptedException
                            caught");
      }
      this.n=n;
      valueSet=true;
      System.out.println("Put :" +n);
      System.out.println("\n Intimate Consumer");
      notify();
    }
  }

  class Producer implements Runnable
  {
    Q q;
    Producer(Q q)
    {
      this.q = q;
      new Thread(this, "Producer").start();
    }
```

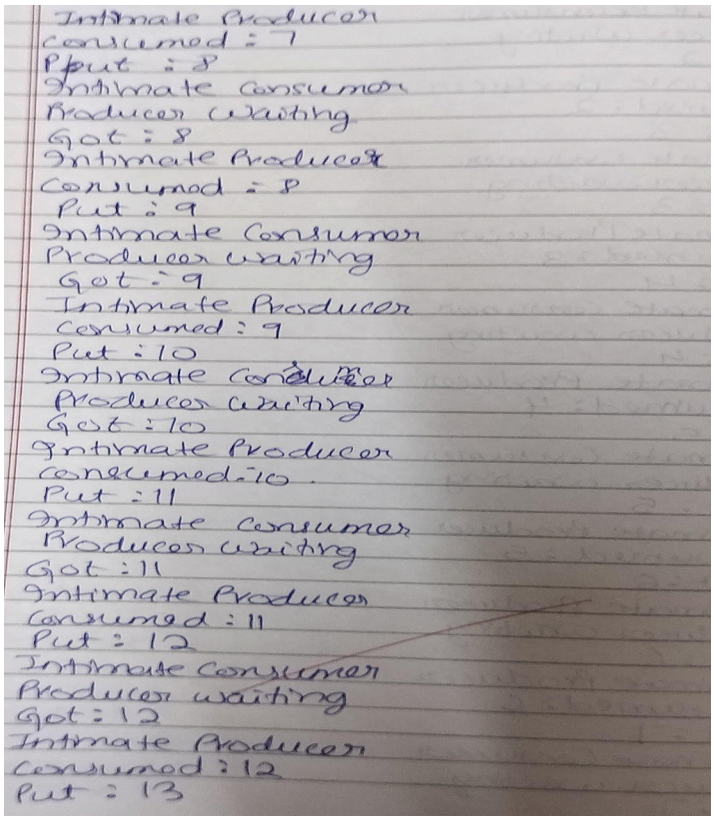
```
public void run()
{
    int i=0;
    while(i<15)
    {
        q.put(i++);
    }
}

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: Kanishka Sharma");
        System.out.println("USN: 1BM23CS138");

        Q q=new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}
```

Output:-

```
Name: Kanishka Sharma
USN : 1BM23CS138

Press Control-C to stop.
Put : 0
Intimate Consumer
Producer writing
Got : 0
Intimate Producer
Put : 1
Intimate Consumer
Producer writing
Consumed : 0
Got : 1
Intimate Producer
Consumed : 1
Put : 2
```

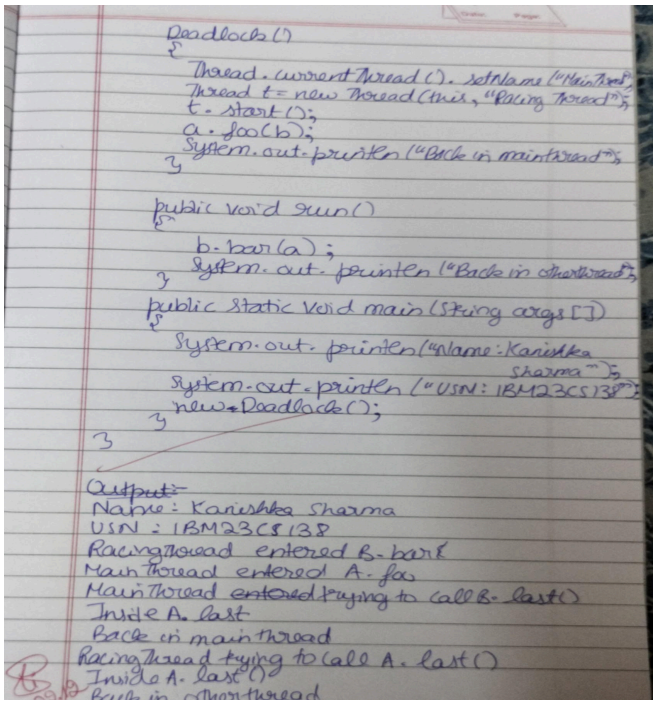

Intimate Consumer
Producer waiting
Got : 2
Intimate Producer
Consumed : 2
Put : 3
Intimate Consumer
Producer waiting
Got : 3
Intimate Producer
Consumed : 3
Put : 4
Intimate Consumer
Producer waiting
Got : 4
Intimate Producer
Consumed : 4
Put 5
Intimate Consumer
Producer waiting
Got : 5
Intimate Producer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


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

Intimate Consumer
Producer Waiting
Got : 13
Intimate Producer
Consumed : 13
Put : 14
Intimate Producer
Consumed : 14

ii.) Source Code :-

```
Class A
{
  synchronized void foo(B b)
  {
    String name = Thread.currentThread().
                                getName();
    System.out.println(name + "enter 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(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();
```



Code:

Inter process Communication:

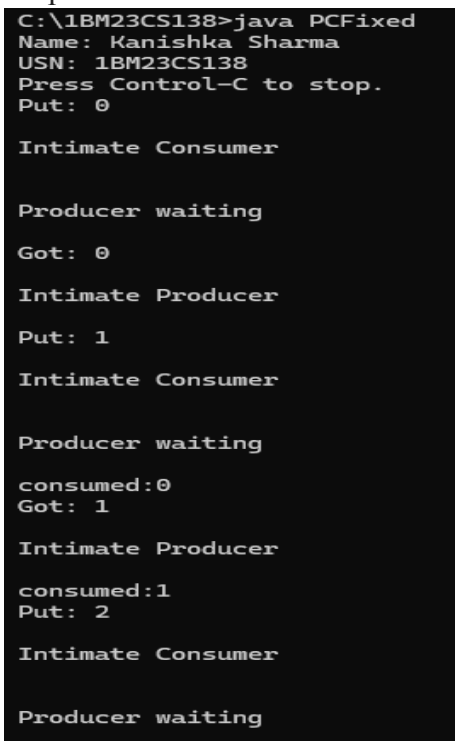
```
class Q {

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: Kanishka Sharma");
System.out.println("USN: 1BM23CS138");
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```

Output:

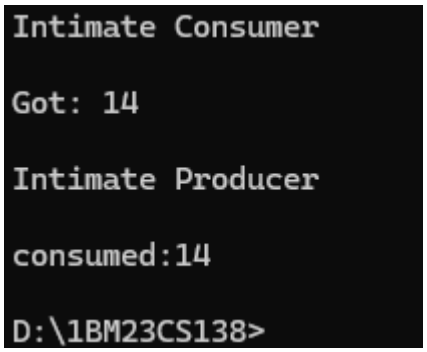


```
Intimate Producer

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
```

```
consumed:5
Put: 6

Intimate Consumer


Producer waiting

Got: 6

Intimate Producer

consumed:6
Put: 7

Intimate Consumer


Producer waiting

Got: 7

Intimate Producer

consumed:7
Put: 8

Intimate Consumer


Producer waiting

Got: 8
```

```
Intimate Producer

consumed:8
Put: 9

Intimate Consumer


Producer waiting

Got: 9

Intimate Producer

consumed:9
Put: 10

Intimate Consumer


Producer waiting

Got: 10

Intimate Producer

consumed:10
Put: 11
```

```
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
```



Deadlock:

```
class A {
synchronized void foo(B b) {
String name =
Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("A Interrupted");
}
System.out.println(name + " trying to call B.last()");
b.last();
}
void last() {
System.out.println("Inside A.last");
}
}
class B {
synchronized void bar(A a) {
String name =
Thread.currentThread().getName();
System.out.println(name + " entered B.bar");
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("B Interrupted");
}
System.out.println(name + " trying to call A.last()");
a.last();
}
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: Kanishka Sharma");
System.out.println("USN: 1BM23CS138");
new Deadlock();
}
}
```

```
C:\1BM23CS138>java Deadlock
Name: Kanishka Sharma
USN: 1BM23CS138
RacingThread entered B.bar
MainThread entered A.foo
MainThread trying to call B.last()
Inside A.last
Back in mainthread
RacingThread trying to call A.last()
Inside A.last
Back in otherthread
```