VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Anjana Manoj (1BM23CS038)

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 **Anjana Manoj (1BM23CS038),** 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.

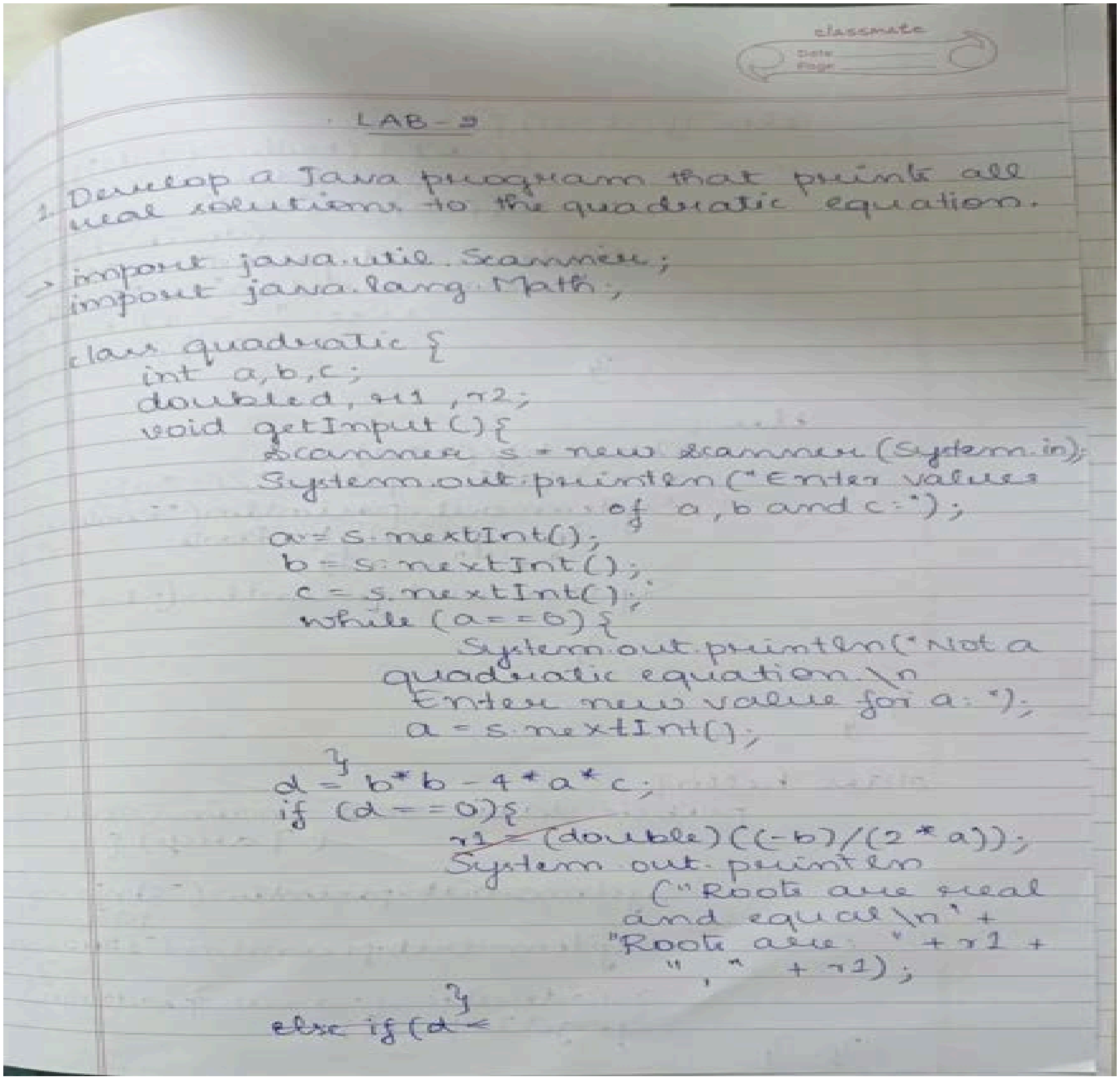
Sheetal V.A Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link:
https://github.com/AnjanaManoj-05/Java-lab

Program 1
Implement Quadratic Equation

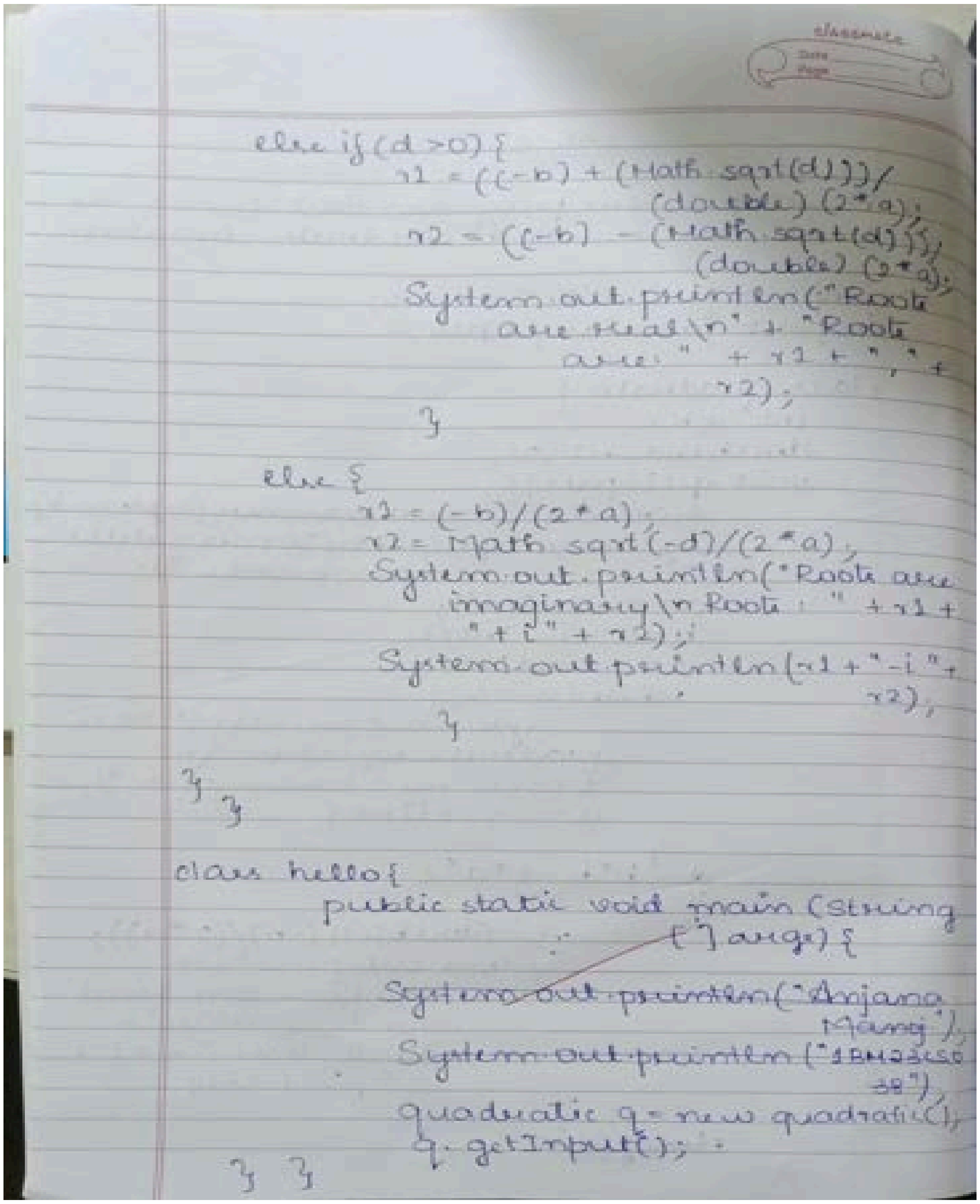
Algorithm:



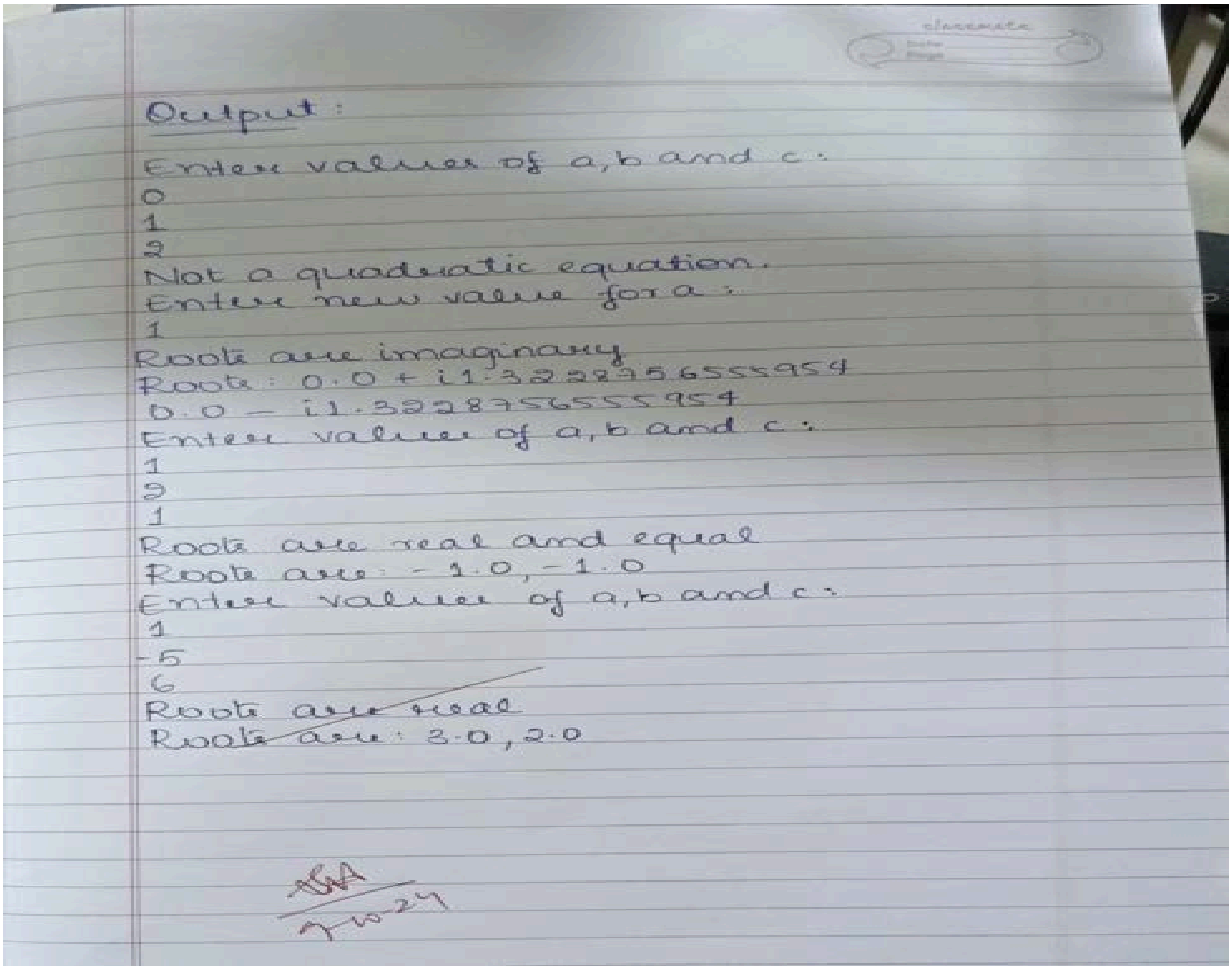
LAB-2

1. Develop a Java program that prints all real solutions to the quadratic equation.

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

class quadratic {
     int a,b,c;
     double d, r1 ,r2;
     void getInput(){
            Scanner s = new Scanner(System.in);
            System.out.println("Enter values
                          of a, b and c:");
            a = s.nextInt();
             b = s.nextInt();
              c = s.nextInt();
              while (a==0){
                     System.out.println("Not a
                quadratic equation.\n
                 Enter new value for a: ");
                 a = s.nextInt();
             }
            d = b*b - 4*a*c;
            if (d==0){
                     r1 = (double)((-b)/(2*a));
                     System.out.println
                           ("Roots are real
                          and equal\n" +
                         "Roots are: " + r1 +
                            " , " + r1);
                  }
            else if(d<
```



```
else if(d>0){
        r1 = ((-b) + (Math.sqrt(d)))/
                        (double) (2*a);
        r2 = ((-b) - (Math.sqrt(d)))/
                          (double) (2*a);
        System.out.println("Roots
             are real\n" + "Roots
                 are: " + r1 + ", " +
                         r2);
     }

  else {
        r1 = (-b)/(2*a);
        r2 = Math.sqrt(-d)/(2*a);
        System.out.println("Roots are
             imaginary\n Roots : " +r1 +
             " + i" + r2);
        System.out.println(r1 + " -i "+
                                  r2);
     }

}
  }

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

            System.out.println("Anjana
                              Manoj");
            System.out.println("1BM23CS0
                               38");
            quadratic q = new quadratic();
            q.getInput();
   } }
```



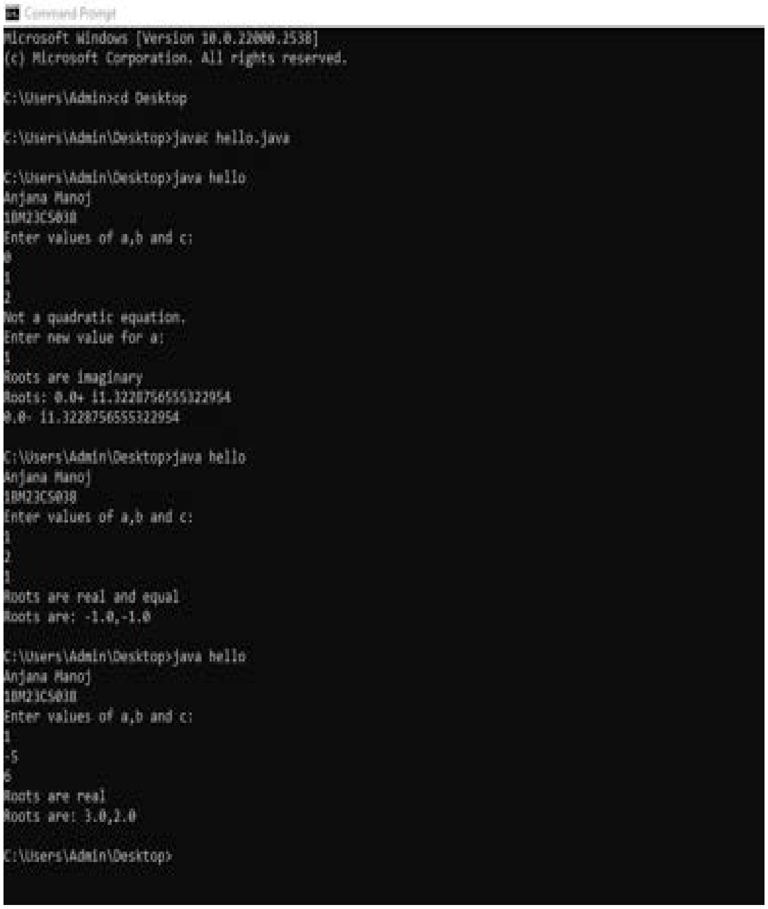
Output:

```
Enter values of a,b and c:
0
1
2
Not a quadratic equation.
Enter new value for a:
1
Roots are imaginary
Roots: 0.0 + i1.3228756555954
0.0 - i1.3228756555954
Enter values of a,b and c:
1
2
1
Roots are real and equal
Roots are: -1.0, -1.0
Enter values of a,b and c:
1
-5
6
Roots are real
Roots are: 3.0, 2.0
```

7-10-24

Code:

```
import java.util.Scanner;
import java.lang.Math;
class quadratic{
int a,b,c;
double d,r1,r2;
void getInput(){
Scanner s =new Scanner(System.in);
System.out.println("Enter values of a,b and c:");
a=s.nextInt();
b=s.nextInt();
c=s.nextInt();
while(a==0){
System.out.println("Not a quadratic equation.\nEnter new value for a:");
a=s.nextInt();

}
d=b*b-4*a*c;
if(d==0){
  r1=(double)((-b)/(2*a));
  System.out.println("Roots are real and equal\n" + "Roots are: " + r1 +","+r1);
}
else if(d>0){
r1= ((-b) + (Math.sqrt(d)))/(double)(2*a);
r2= ((-b) - (Math.sqrt(d)))/(double)(2*a);
System.out.println("Roots are real\n" + "Roots are: " + r1 +","+r2 );
}
else{
r1= (-b)/(2*a);
r2 = Math.sqrt(-d)/(2*a);
System.out.println("Roots are imaginary\nRoots: " + r1 + "+ i" +r2 );
System.out.println(r1 +"- i"+r2);
}


}}

class hello{
public static void main(String [] args){
System.out.println("Anjana Manoj");
System.out.println("1BM23CS038");
quadratic q=new quadratic();
q.getInput();




}}
```

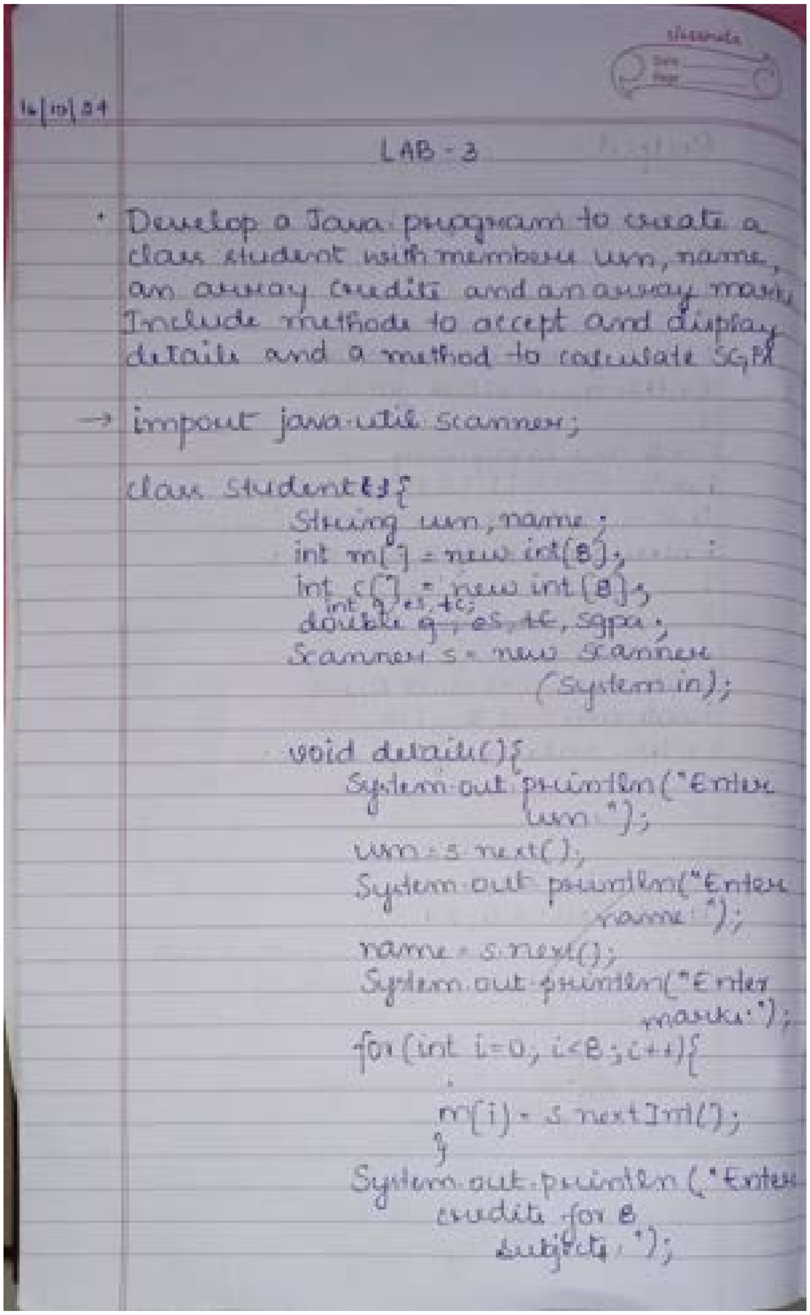


```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd Desktop

C:\Users\Admin\Desktop>javac hello.java

C:\Users\Admin\Desktop>java hello
Anjana Manoj
1BM23CS038
Enter values of a,b and c:
0
1
2
Not a quadratic equation.
Enter new value for a:
1
Roots are imaginary
Roots: 0.0+ i1.3228756555322954
0.0- i1.3228756555322954

C:\Users\Admin\Desktop>java hello
Anjana Manoj
1BM23CS038
Enter values of a,b and c:
1
2
1
Roots are real and equal
Roots are: -1.0,-1.0

C:\Users\Admin\Desktop>java hello
Anjana Manoj
1BM23CS038
Enter values of a,b and c:
1
-5
6
Roots are real
Roots are: 3.0,2.0

C:\Users\Admin\Desktop>
```

Program 2

Calculation of Student SGPA

Algorithm:



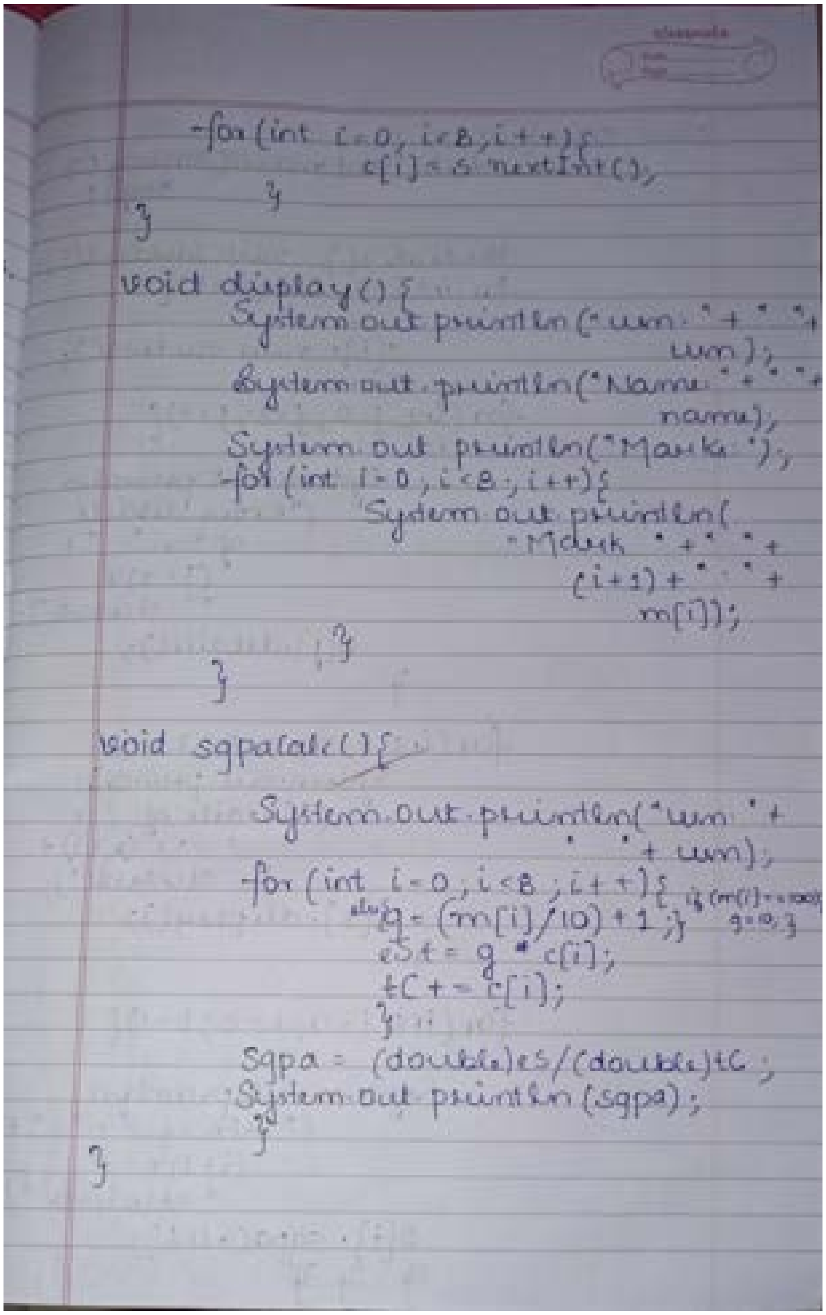
16/10/24

LAB - 3

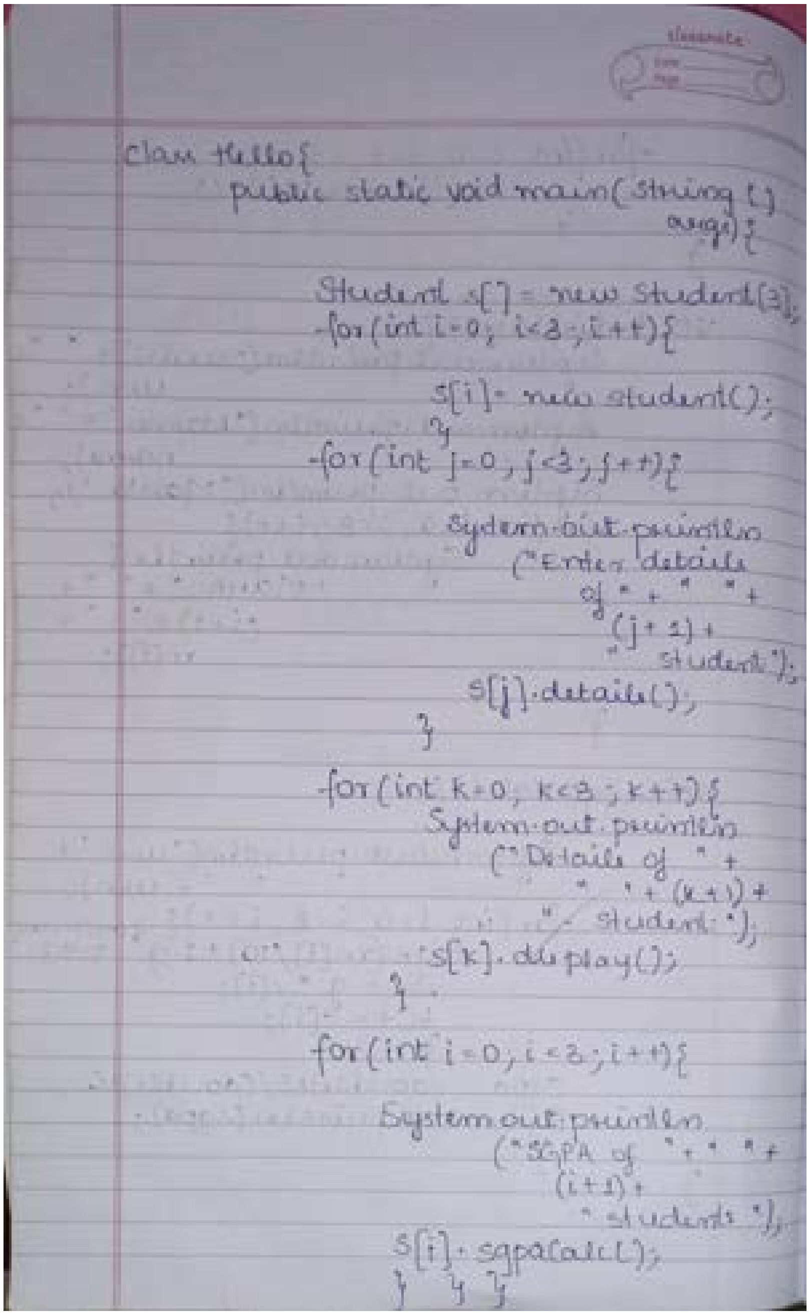
- Develop a Java program to create a class student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA

→ import java.util.Scanner;

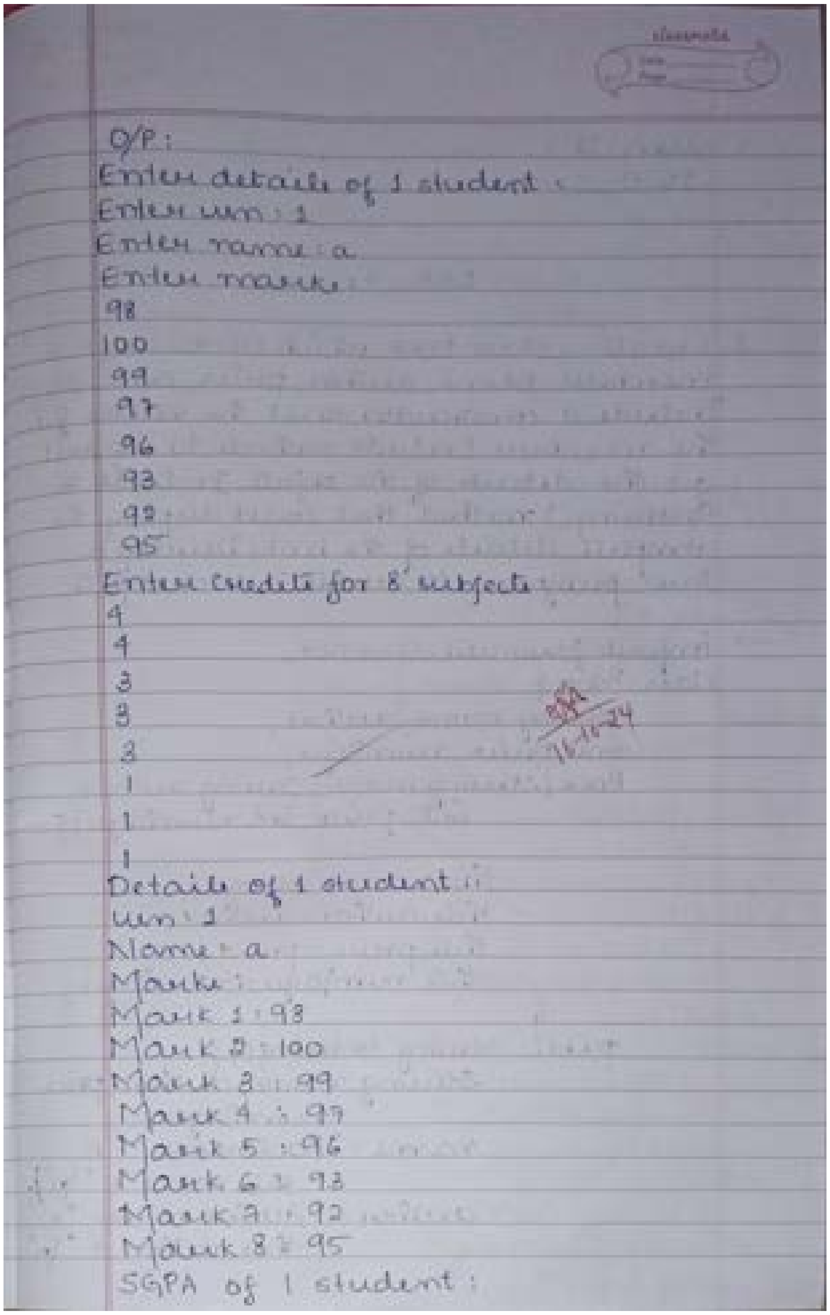
```
class Student~~()~~{
        String usn, name;
        int m[] = new int[8];
        int c[] = new int[8];
        int g, es, tc;
        double ~~g, es, tc,~~ sgpa;
        Scanner s = new Scanner
                        (System.in);

        void details(){
            System.out.println("Enter
                            usn:");
            usn = s.next();
            System.out.println("Enter
                            name:");
            name = s.next();
            System.out.println("Enter
                            marks:");
            for(int i=0; i<8; i++){

                m[i] = s.nextInt();
            }
            System.out.println("Enter
                credits for 8
                    subjects:");
```



```
    for(int i=0;i<8;i++){
            c[i]=s.nextInt();
        }
}

void display(){
    System.out.println("usn: "+" "+
                            usn);
    System.out.println("Name: "+" "+
                            name);
    System.out.println("Marks ");
    for(int i=0 ; i<8 ; i++){
            System.out.println(
                "Mark "+" "+
                    (i+1)+" "+
                        m[i]);
        }
    }

void sgpacalc(){
    System.out.println("usn "+
                    " "+ usn);
    for(int i=0 ; i<8 ; i++){ { if(m[i]==100){
          else g=(m[i]/10)+1;}      g=10;}
          eS+= g * c[i];
          tC+= c[i];
        }
    Sgpa = (double)eS/(double)tC ;
    System.out.println(sgpa);
    }
}
```



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

                Student s[] = new Student[3];
                for(int i=0; i<3; i++){

                        s[i]= new student();
                }
                for(int j=0, j<3; j++){

                        System.out.println
                                ("Enter details
                                  of " + " " +
                                   (j+1) +
                                   " student");
                        s[j].details();
                }

                for(int k=0; k<3 ; k++){
                        System.out.println
                                ("Details of " +
                                   " " + (k+1) +
                                   " student ");
                        s[k].display();
                }

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

                        System.out.println
                                ("SGPA of " + " " +
                                  (i+1) +
                                  " students ");
                s[i].sgpacalc();
}  }  }
```



```
O/P:
Enter details of 1 student :
Enter usn : 1
Enter name : a
Enter marks :
98
100
99
97
96
93
92
95
Enter credits for 8 subjects
4
4
3
3
3
1
1
1
Details of 1 student :
usn : 1
Name : a
Marks :
Mark 1 : 98
Mark 2 : 100
Mark 3 : 99
Mark 4 : 97
Mark 5 : 96
Mark 6 : 93
Mark 7 : 92
Mark 8 : 95
SGPA of 1 student :
```

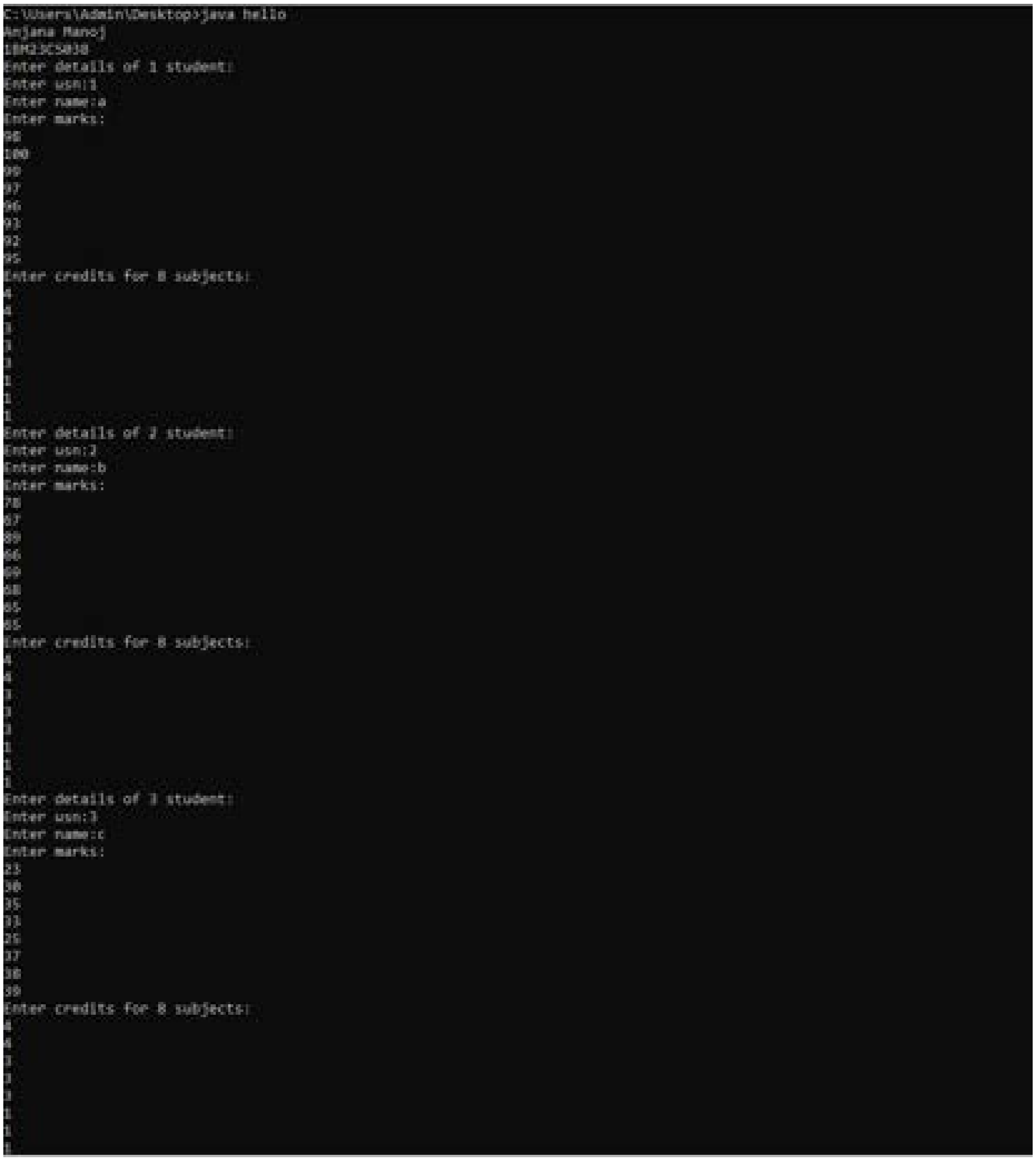
11-12-24

<u>Code:</u>

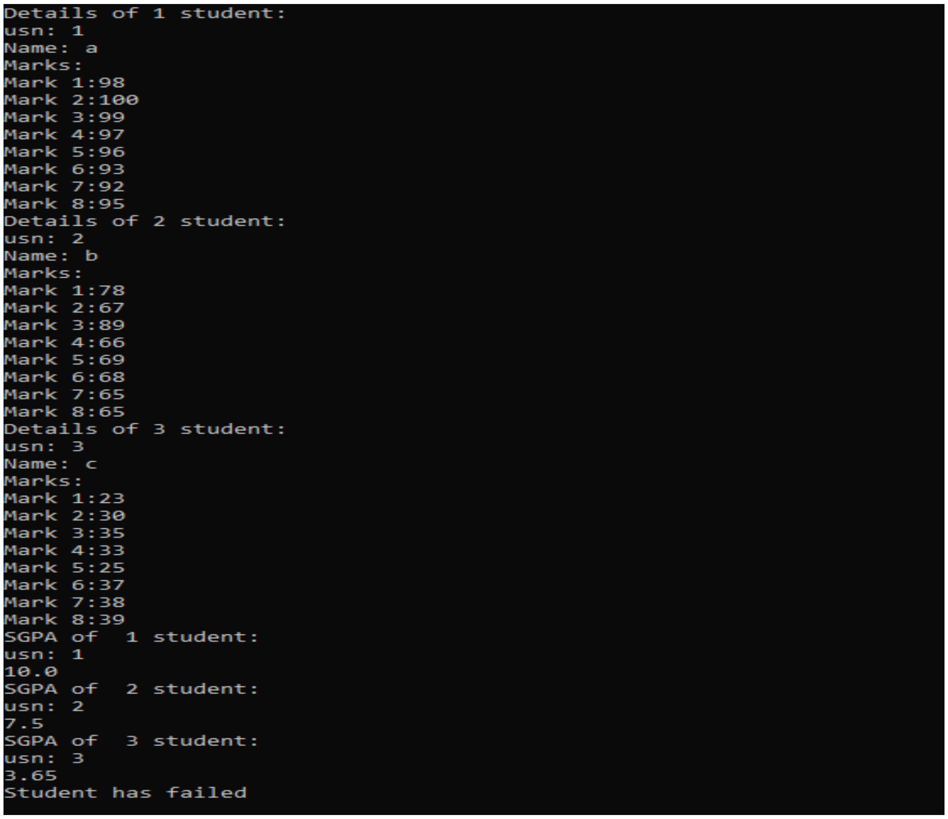
```
import java.util.Scanner;
class Student{
String usn,name;
int m[]=new int[8];
int c[]=new int[8];
int g,eS,tC;
double sgpa;
Scanner s=new Scanner(System.in);
void details(){
```

```
System.out.print("Enter usn:");
usn=s.next();
System.out.print("Enter name:");
name=s.next();
System.out.println("Enter marks:");
for(int i=0;i<8;i++){
m[i]=s.nextInt();
}
System.out.println("Enter credits for 8 subjects:");
for(int i=0;i<8;i++){
c[i]=s.nextInt();
}
}
void display(){
System.out.println("usn:" +" "+usn);
System.out.println("Name:" +" "+name);
System.out.println("Marks:");
for(int i=0;i<8;i++){
System.out.println("Mark " + (i+1) +":" +m[i]);

}

}
void sgpaCalc(){
System.out.println("usn:" +" "+usn);
for(int i=0;i<8;i++){
if(m[i]==100){
g=10;
}
else{
g=(m[i]/10)+1;
}
eS+=g*c[i];
tC+=c[i];
}
sgpa=(double)eS/(double)tC;
System.out.println(sgpa);
if(sgpa<=4.0)
{
System.out.println("Student has failed");
}
}
}
class hello{
public static void main(String [] args){
```

```
System.out.println("Anjana Manoj");
System.out.println("1BM23CS038");
Student s[]=new Student[3];
for(int i=0;i<3;i++){
s[i]=new Student();
}
for(int j=0;j<3;j++){
System.out.println("Enter details of "+""+(j+1)+" student:");
s[j].details();
}
for(int k=0;k<3;k++){
System.out.println("Details of "+""+(k+1)+" student:");
s[k].display();
}
for(int i=0;i<3;i++){
System.out.println("SGPA of "+" "+(i+1)+" student:");
s[i].sgpaCalc();
}

}
}
```



```
C:\Users\Admin\Desktop>java hello
Anjana Manoj
1BM23CS030
Enter details of 1 student:
Enter usn:1
Enter name:a
Enter marks:
98
100
99
97
96
93
92
95
Enter credits for 8 subjects:
4
4
3
3
3
1
1
1
Enter details of 2 student:
Enter usn:2
Enter name:b
Enter marks:
78
67
89
66
69
68
65
65
Enter credits for 8 subjects:
4
4
3
3
3
1
1
1
Enter details of 3 student:
Enter usn:3
Enter name:c
Enter marks:
23
30
35
33
25
37
38
39
Enter credits for 8 subjects:
4
4
3
3
3
1
1
1
```

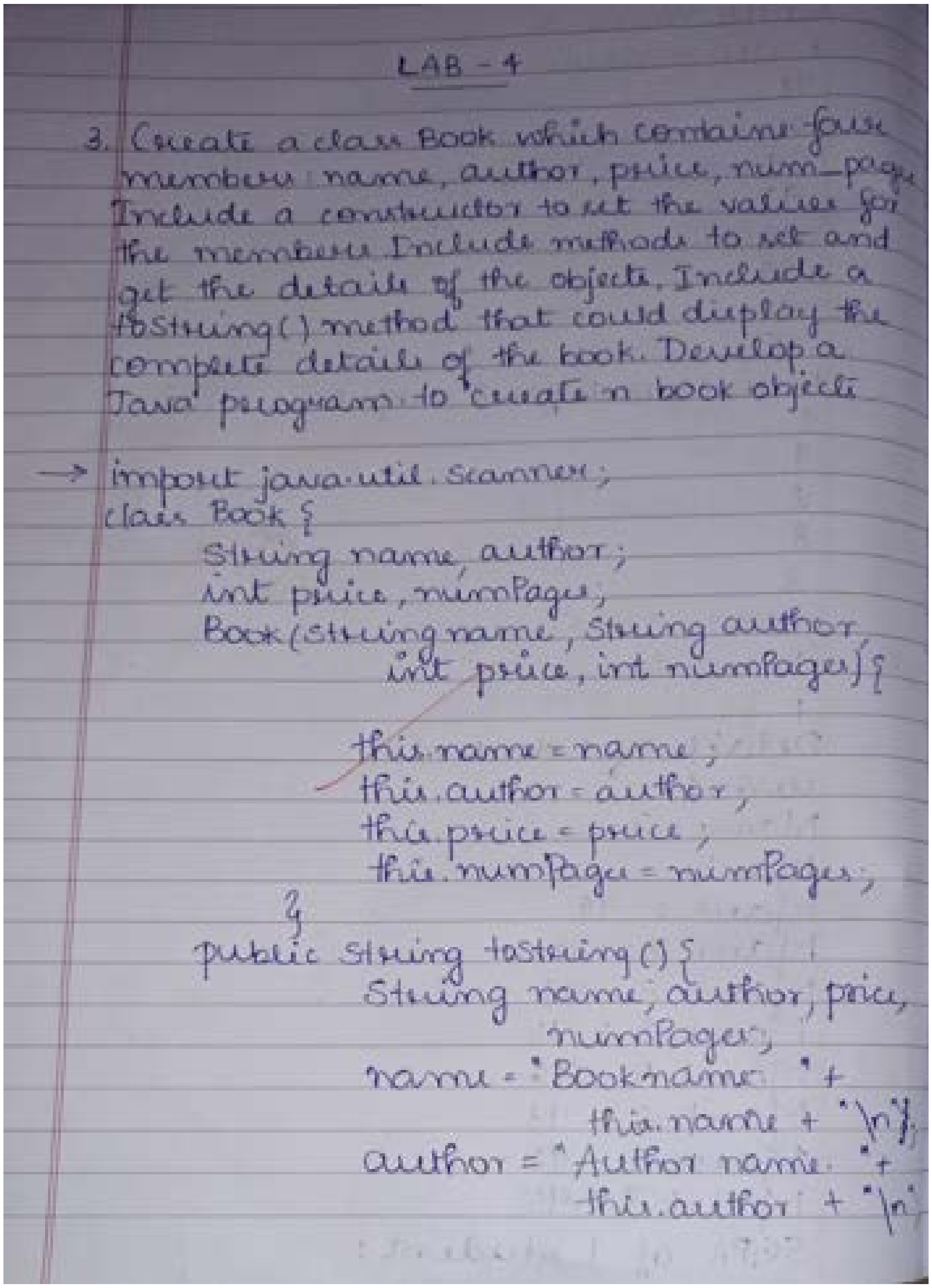


```
Details of 1 student:
usn: 1
Name: a
Marks:
Mark 1:98
Mark 2:100
Mark 3:99
Mark 4:97
Mark 5:96
Mark 6:93
Mark 7:92
Mark 8:95
Details of 2 student:
usn: 2
Name: b
Marks:
Mark 1:78
Mark 2:67
Mark 3:89
Mark 4:66
Mark 5:69
Mark 6:68
Mark 7:65
Mark 8:65
Details of 3 student:
usn: 3
Name: c
Marks:
Mark 1:23
Mark 2:30
Mark 3:35
Mark 4:33
Mark 5:25
Mark 6:37
Mark 7:38
Mark 8:39
SGPA of  1 student:
usn: 1
10.0
SGPA of  2 student:
usn: 2
7.5
SGPA of  3 student:
usn: 3
3.65
Student has failed
```

Program 3

Demonstration of array of objects of each book type

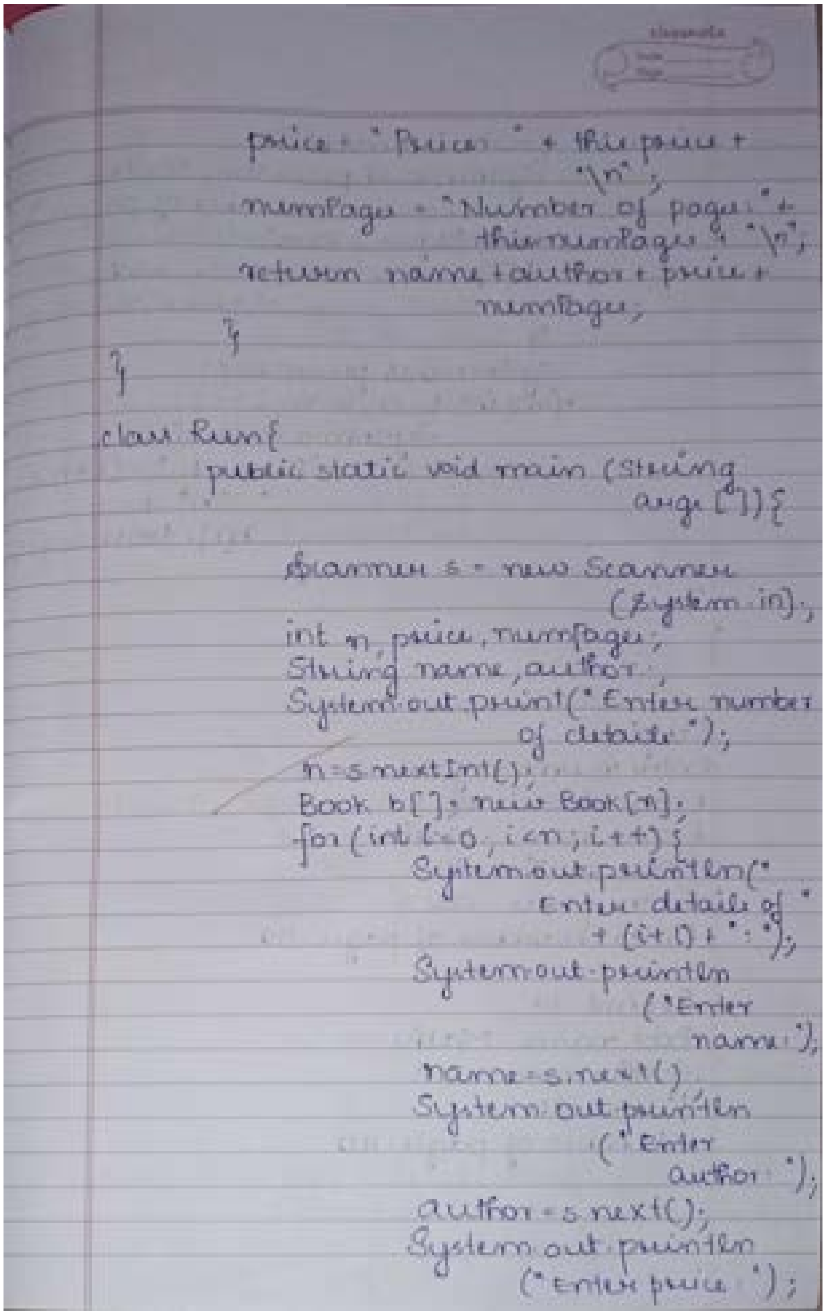
Algorithm:



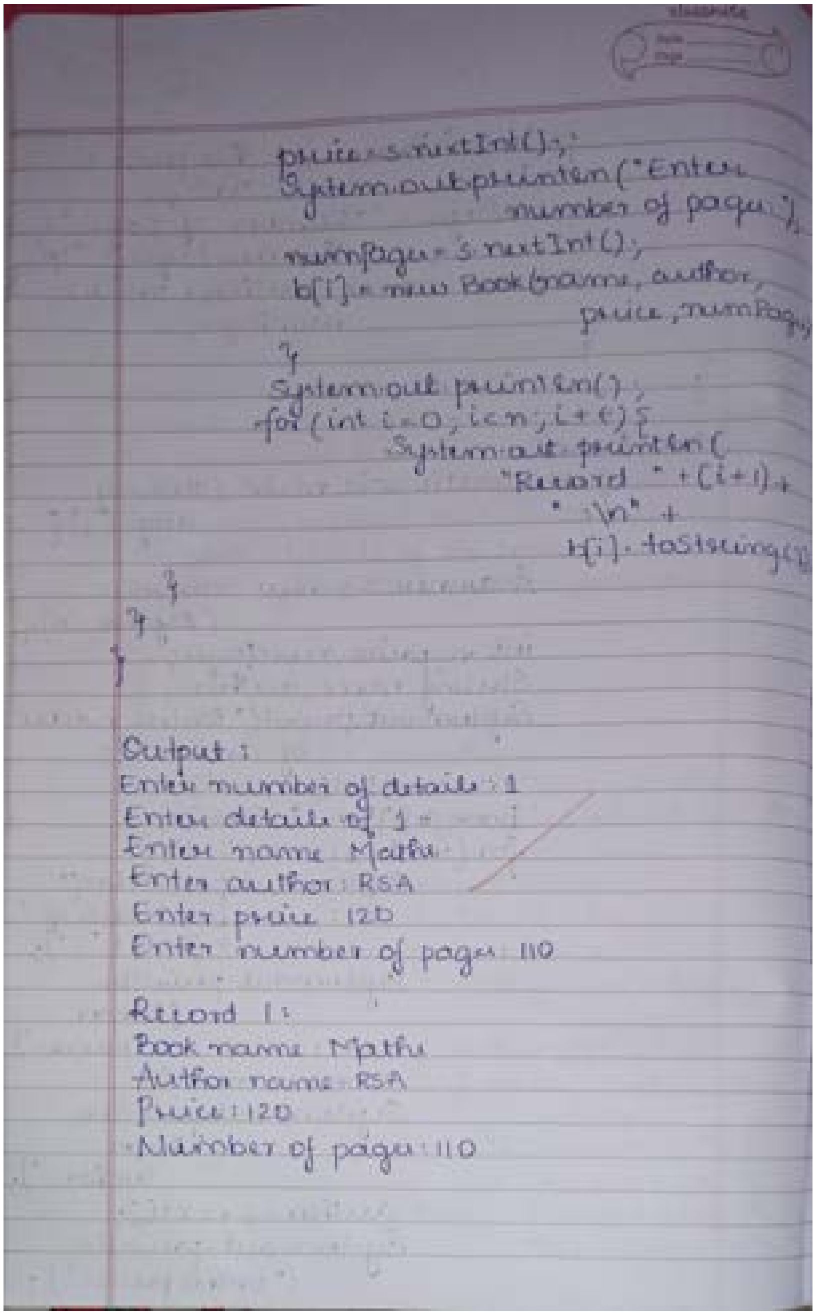
LAB - 4

3. 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

→
```
import java.util.Scanner;
class Book {
        String name, author;
        int price, numPages;
        Book(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 = "Bookname "+
                            this.name + "\n";
                author = "Author name "+
                           this.author + "\n";
```



```
        price = "Price: " + this.price +
                              "\n";
        numPage = "Number of page: " +
                        this.numPage + "\n";
        return name+author+ price +
                        numPage;
    }
}

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

        Scanner s = new Scanner
                        (System.in);
        int n, price, numPage;
        String name, author;
        System.out.print(" Enter number
                        of details ");
         n=s.nextInt();
         Book b[]= new Book[n];
         for(int i=0; i<n; i++){
               System.out.println("
                      Enter details of "
                        + (i+1) + ": ");
               System.out.println
                        (" Enter
                            name: ");
                name=s.next()
               System.out.println
                       (" Enter
                          author: ");
               author=s.next();
               System.out.println
                     (" Enter price: ");
```



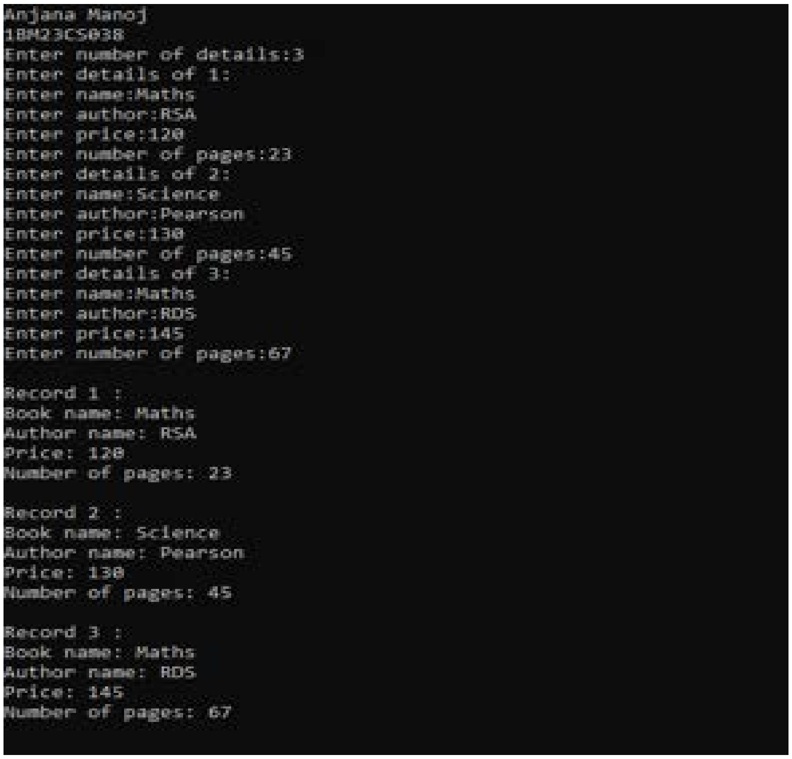
```
        price=s.nextInt();
        System.out.println("Enter
                      number of pages:");
        numPages=s.nextInt();
        b[i]=new Book(name,author,
                          price,numPages);
        }
        System.out.println();
       for(int i=0;i<n;i++){
               System.out.println(
                  "Record "+(i+1)+
                    ":\n"+
                  b[i].toString());
    }
  }
}
```

Output:

```
Enter number of details: 1
Enter details of 1:
Enter name: Maths
Enter author: RSA
Enter price: 120
Enter number of pages: 110

Record 1:
Book name: Maths
Author name: RSA
Price: 120
Number of pages: 110
```

Code:

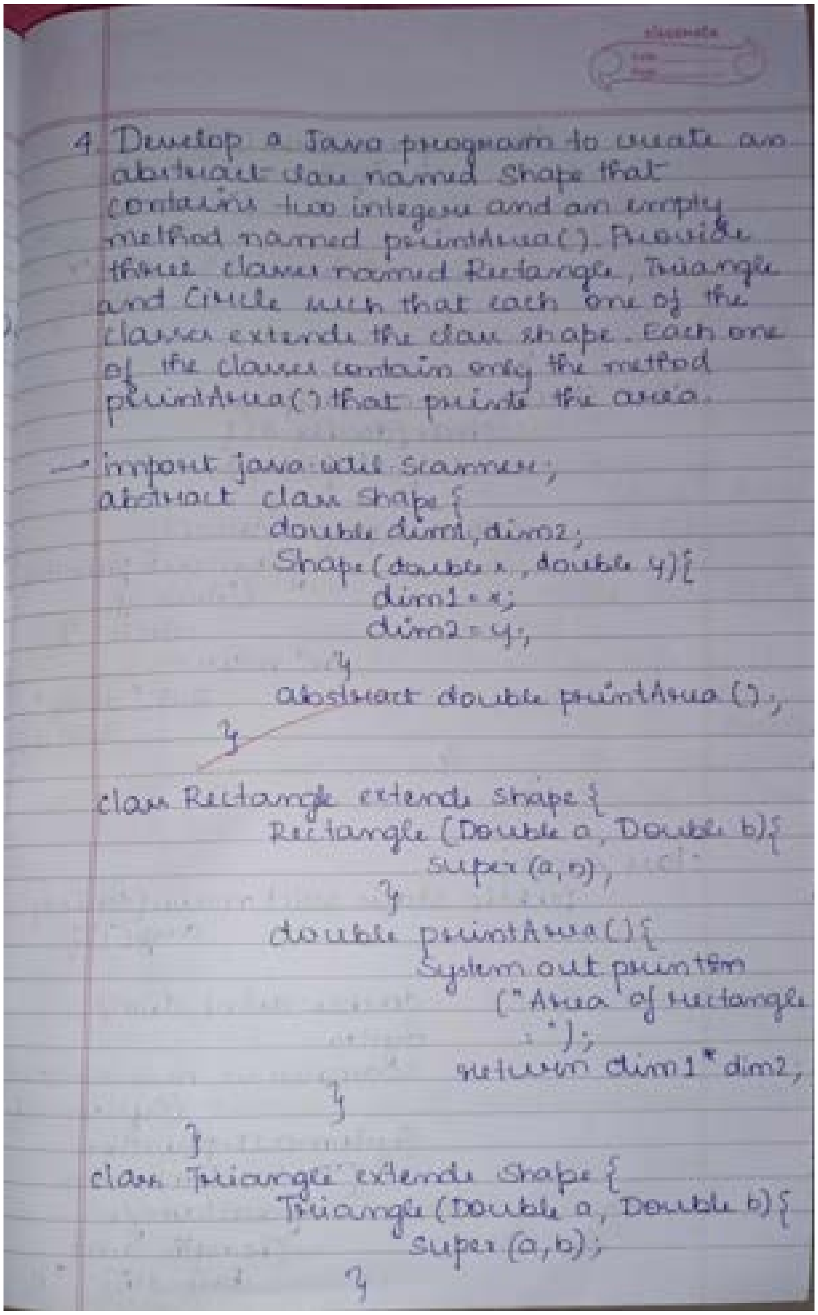
```
import java.util.Scanner;
class Book{
String name,author;
int price,numPages;
Book(String name,String author,int price,int numPages){
this.name=name;
this.author=author;
this.price=price;
this.numPages=numPages;
}
public String toString(){
String name,author,price,numPages;
name = "Book name: " + 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 Run{
public static void main(String args[]){
Scanner s=new Scanner(System.in);
int n,price,numPages;
String name,author;
System.out.println("Anjana Manoj");
System.out.println("1BM23CS038");
System.out.print("Enter number of details:");
n=s.nextInt();
Book b[]=new Book[n];
for(int i=0;i<n;i++){
System.out.println("Enter details of " +(i+1)+":");
System.out.print("Enter name:");
name=s.next();
System.out.print("Enter author:");
author=s.next();
System.out.print("Enter price:");
price=s.nextInt();
System.out.print("Enter number of pages:");
numPages=s.nextInt();
b[i]=new Book(name,author,price,numPages);
}
System.out.println();
for(int i=0;i<n;i++){
System.out.println("Record "+(i+1)+" :\n"+b[i].toString());
}

}
}
```



```
Anjana Manoj
1BM23CS038
Enter number of details:3
Enter details of 1:
Enter name:Maths
Enter author:RSA
Enter price:120
Enter number of pages:23
Enter details of 2:
Enter name:Science
Enter author:Pearson
Enter price:130
Enter number of pages:45
Enter details of 3:
Enter name:Maths
Enter author:RDS
Enter price:145
Enter number of pages:67

Record 1 :
Book name: Maths
Author name: RSA
Price: 120
Number of pages: 23

Record 2 :
Book name: Science
Author name: Pearson
Price: 130
Number of pages: 45

Record 3 :
Book name: Maths
Author name: RDS
Price: 145
Number of pages: 67
```

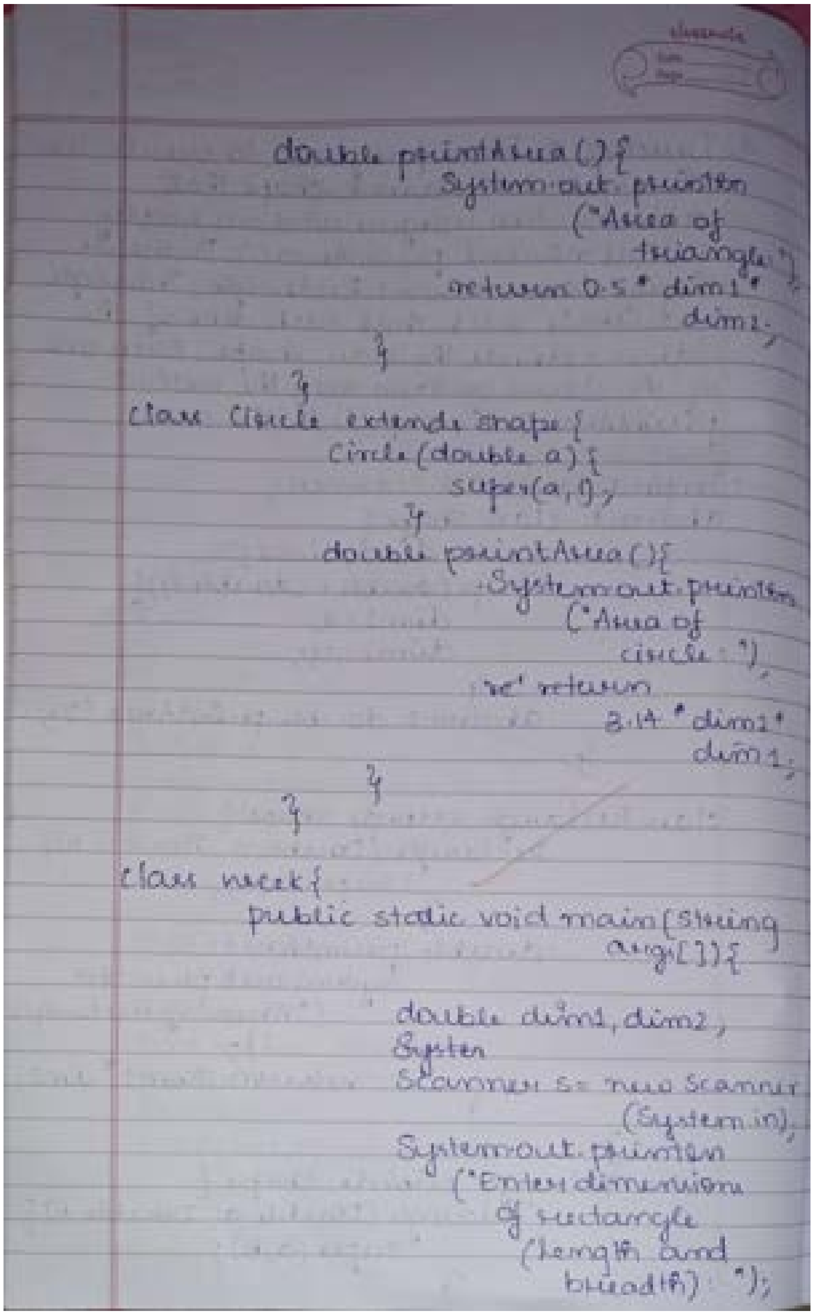
Program 4
Demonstration of Abstract class

Algorithm:

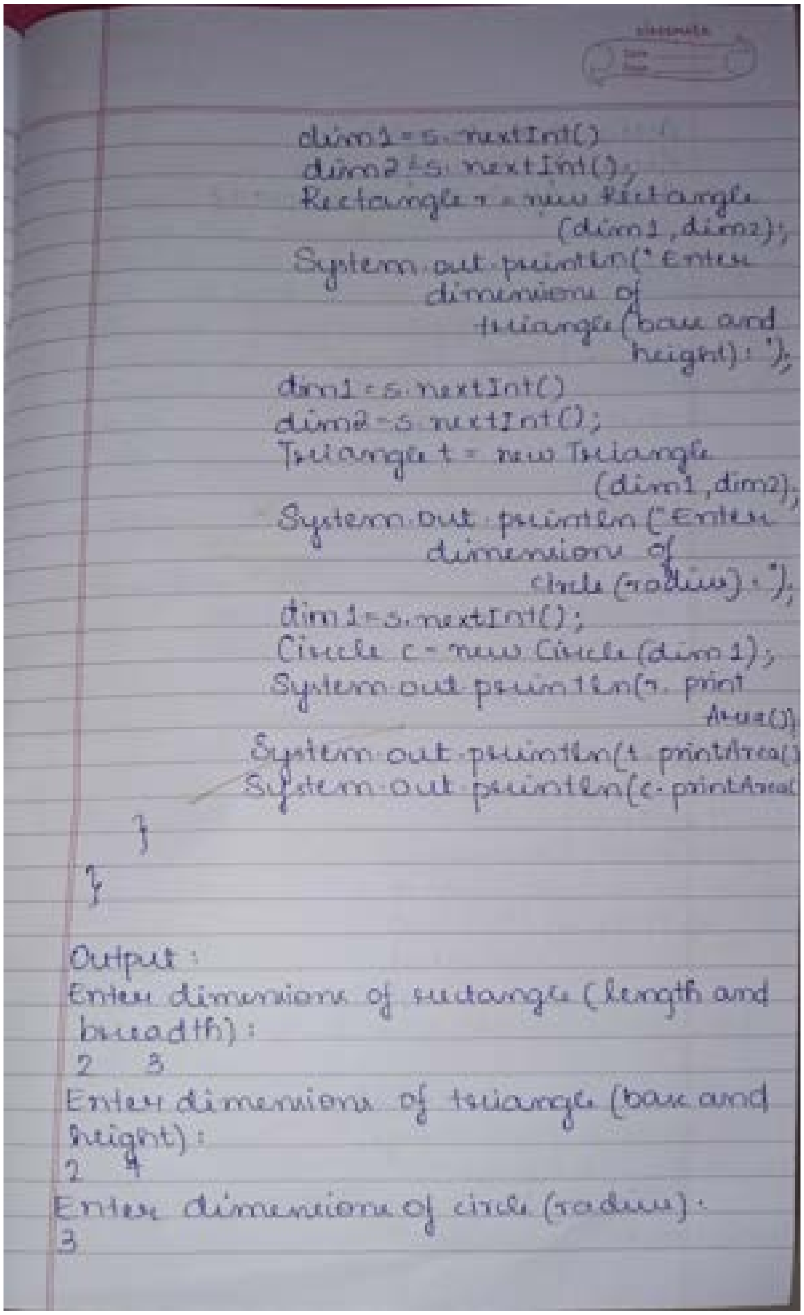


4. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea()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

```
→ import java.util.Scanner;
abstract class Shape{
            double dim1, dim2;
            Shape(double x, double y){
                    dim1=x;
                    dim2=y;
            }
            abstract double printArea();
      }

class Rectangle extends Shape{
            Rectangle(Double a, Double b){
                    super(a,b);
            }
            double printArea(){
                    System.out.println
                          ("Area of Rectangle
                           : ");
                      return dim1*dim2;
            }
      }
class Triangle extends Shape{
            Triangle(Double a, Double b){
                    super(a,b);
            }
```



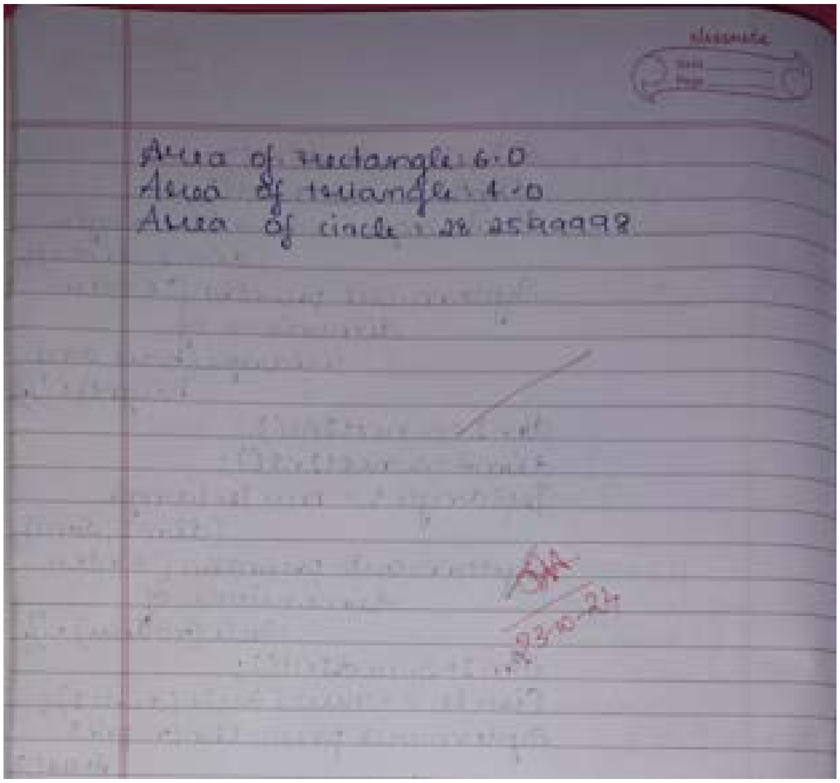
```
        double printArea(){
                    System.out.println
                              ("Area of
                                  triangle:");
                    return 0.5 * dim1 *
                                   dim2;
              }
        }
class Circle extends shape{
            Circle(double a){
                    super(a,1);
              }
            double printArea(){
                    System.out.println
                              ("Area of
                                  circle:");
                    re' return
                              3.14 * dim1 *
                                   dim1;
              }
        }

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

                double dim1, dim2;
                System
                Scanner s= new Scanner
                              (System.in);
                System.out.println
                    ("Enter dimensions
                      of rectangle
                        (length and
                          breadth) :");
```



```
        dim1=s.nextInt()
        dim2=s.nextInt();
        Rectangle r = new Rectangle
                          (dim1,dim2);
        System.out.println("Enter
                  dimensions of
                     triangle(base and
                          height): ");
        dim1=s.nextInt()
        dim2=s.nextInt();
        Triangle t = new Triangle
                          (dim1,dim2);
        System.out.println("Enter
                  dimensions of
                       Circle(radius): ");
        dim1=s.nextInt();
        Circle c = new Circle(dim1);
        System.out.println(r.print
                               Area());
        System.out.println(t.printArea()
        System.out.println(c.printArea(
    }
}
```

Output:

```
Enter dimensions of rectangle (length and
breadth):
2   3
Enter dimensions of triangle (base and
height):
2   4
Enter dimensions of circle (radius):
3
```

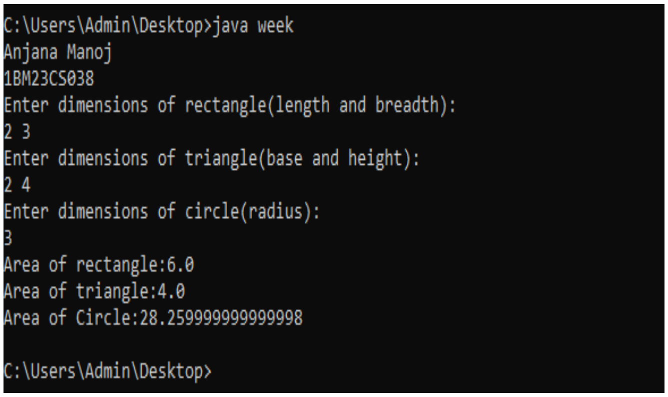


```
Area of rectangle: 6.0
Area of triangle: 4.0
Area of circle: 28.2599998
```

23-10-24

Code:

```
import java.util.Scanner;
abstract class Shape{
double dim1,dim2;
Shape(double x,double y){
dim1=x;
dim2=y;
}
abstract double printArea();
}
class Rectangle extends Shape{
Rectangle(double a,double b){
super(a,b);
}
double printArea(){
System.out.print("Area of rectangle:");
return dim1*dim2;
}
}
class Triangle extends Shape{
Triangle(double a,double b){
super(a,b);
}
double printArea(){
System.out.print("Area of triangle:");
return 0.5*dim1*dim2;
}
}
class Circle extends Shape{
Circle(double a){
super(a,1);
}
double printArea(){
System.out.print("Area of Circle:");
return 3.14*dim1*dim1;
}
}

class week{
public static void main(String args[]){
double dim1,dim2;
System.out.println("Anjana Manoj");
System.out.println("1BM23CS038");
Scanner s=new Scanner(System.in);
System.out.println("Enter dimensions of rectangle(length and breadth):");
dim1=s.nextInt();
dim2=s.nextInt();
Rectangle r=new Rectangle(dim1,dim2);
System.out.println("Enter dimensions of triangle(base and height):");
dim1=s.nextInt();
dim2=s.nextInt();
Triangle t=new Triangle(dim1,dim2);
System.out.println("Enter dimensions of circle(radius):");
dim1=s.nextInt();
Circle c=new Circle(dim1);
System.out.println(r.printArea());
System.out.println(t.printArea());
System.out.println(c.printArea());


}
}
```

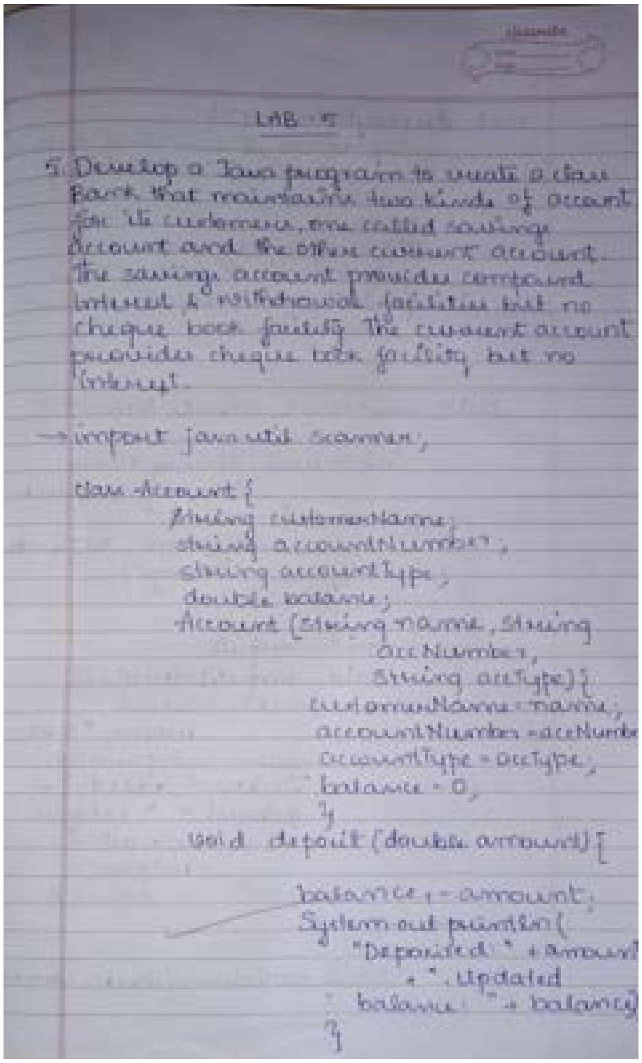


```
C:\Users\Admin\Desktop>java week
Anjana Manoj
1BM23CS038
Enter dimensions of rectangle(length and breadth):
2 3
Enter dimensions of triangle(base and height):
2 4
Enter dimensions of circle(radius):
3
Area of rectangle:6.0
Area of triangle:4.0
Area of Circle:28.259999999999998

C:\Users\Admin\Desktop>
```

Program 5

Demonstration of Bank

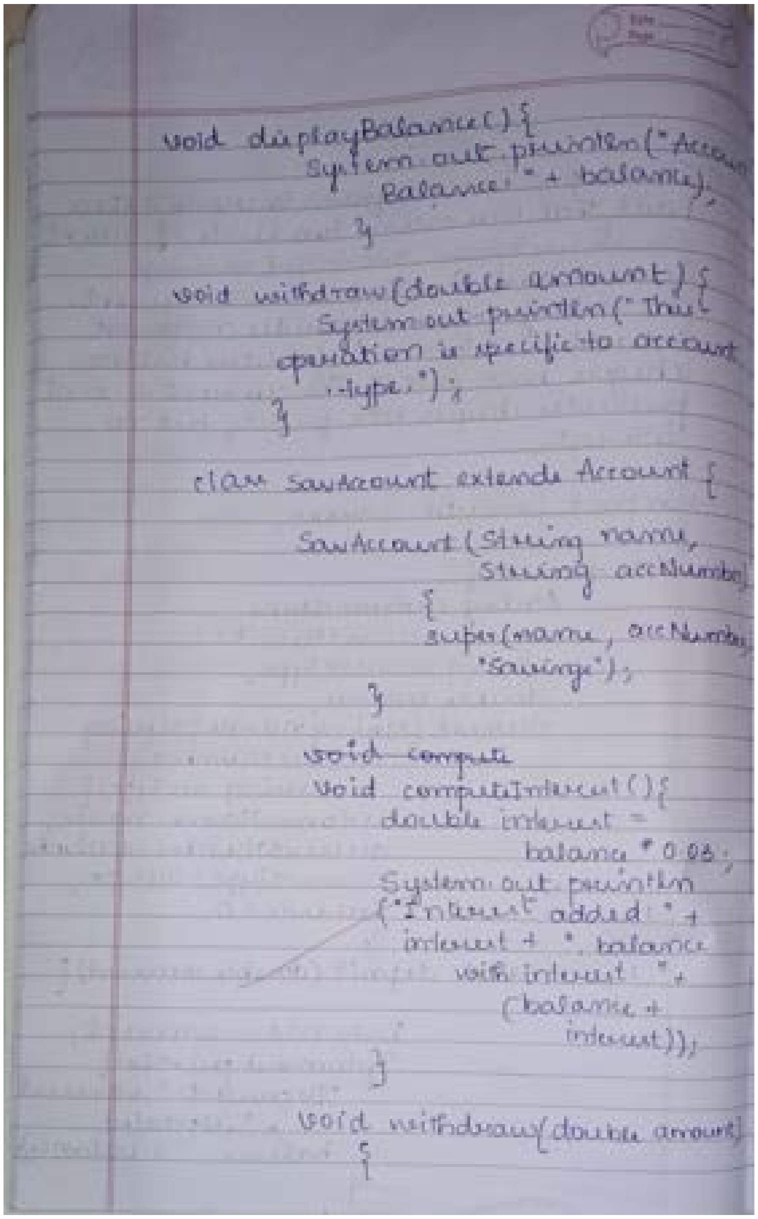
Algorithm:



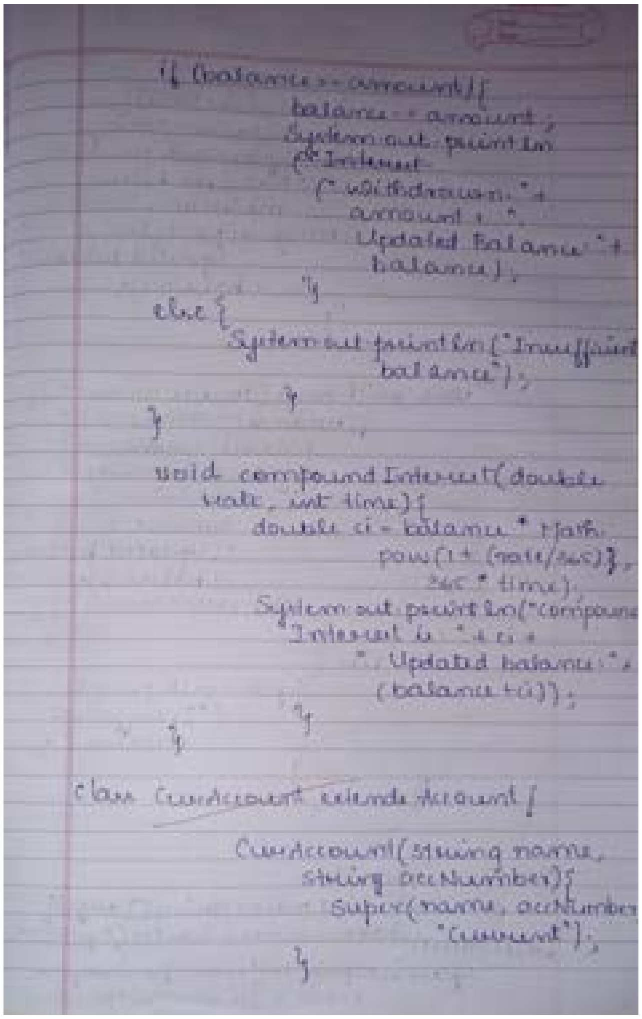
LAB - 5

5. 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 & withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest.

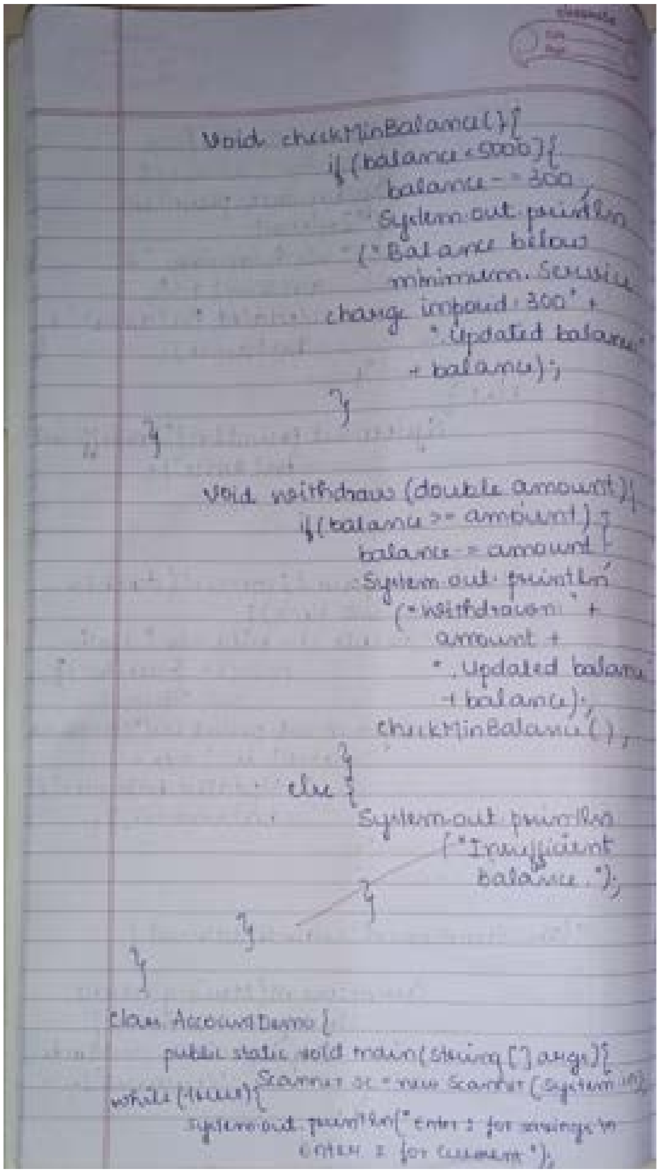
```
→ import java.util.Scanner;

class Account {
        String customerName;
        string accountNumber;
        String accountType;
        double balance;
        Account (String name, String
                        accNumber,
                        String accType){
                customerName = name;
                accountNumber = accNumb
                accountType = accType;
                balance = 0;
                }
        void deposit (double amount){

                balance += amount;
                System.out.println(
                    "Deposited " + amoun
                       + ". Updated
                    balance: " + balance)
                }
```



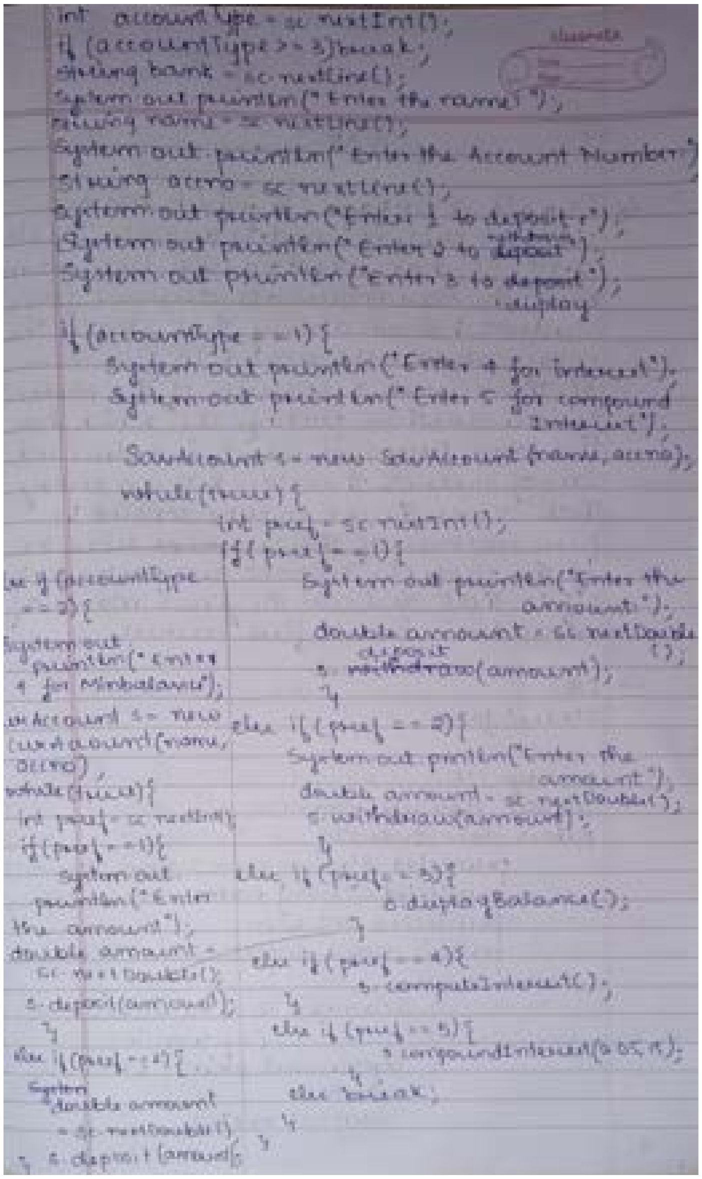
```
void displayBalance(){
        System.out.println("Account
            Balance: " + balance);
    }

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

class SavAccount extends Account {

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

    ~~void compute~~
    void computeInterest(){
        double interest =
                balance * 0.03;
        System.out.println
        ("Interest added: " +
         interest + ", balance
            with interest: " +
             (balance +
                 interest));
        }

    void withdraw(double amount)
        {
```



```
if (balance >= amount){
            balance -= amount;
            System.out.println
            ~~("Interest~~
              ("Withdrawn: "+
                amount + ",
                Updated Balance: "+
                 balance);
           }
else{
      System.out.println("Insufficient
                 balance");
         }
}

void compoundInterest(double
    rate, int time){
      double ci = balance * Math.
                 pow(1+ (rate/365)),
                   365 * time);
      System.out.println("Compound
        Interest is: "+ ci +
             ", Updated balance: "+
              (balance+ci));
         }
  }

class CurrAccount extends Account{

        CurrAccount(String name,
            String accNumber){
              super(name, accNumber,
                    "current");
         }
```

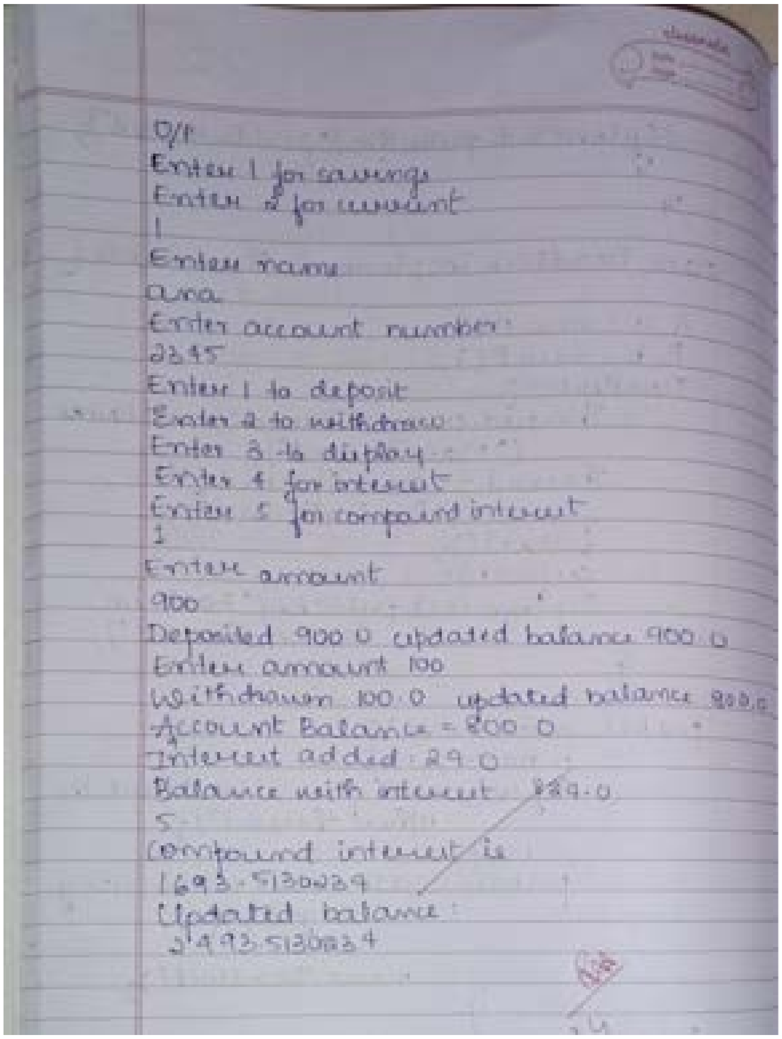


```
Void checkMinBalance(){
        if(balance<5000){
            balance -= 300;
            System.out.println
            ("Balance below
              minimum. Service
        charge impoud: 300" +
              ". Updated balance"
              + balance);
        }
    }

    Void withdraw (double amount){
        if(balance >= amount) {
            balance -= amount;
            System.out.println
             ("Withdrawn: " +
              amount +
              ". Updated balance"
              + balance);
            checkMinBalance();
        }
        else {
            System.out.println
                ("Insufficient
                 balance.");
        }
    }
}
class AccountDemo {
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
while (true){
        System.out.println("Enter 1 for savings \n
                Enter 2 for Current");
```



```
int accountType = sc.nextInt();
if (accountType >= 3)break;
String bank = sc.nextLine();
System.out.println("Enter the name ");
String name = sc.nextLine();
System.out.println("Enter the Account Number")
String accno = sc.nextLine();
System.out.println("Enter 1 to deposit ");
System.out.println("Enter 2 to ~~deposit~~ withdraw");
System.out.println("Enter 3 to ~~deposit~~ display");

if (accountType == 1) {
    System.out.println("Enter 4 for interest").
    System.out.println("Enter 5 for compound
                                   Interest").
    SavAccount s = new SavAccount (name, accno);
    while (true) {
            int pref = sc.nextInt();
            if ( pref == 1) {
                System.out.println("Enter the
                                     amount");
                double amount = sc.nextDouble
                                         ();
                s.~~withdraw~~ deposit(amount);
                }
            else if ( pref == 2) {
                System.out.println("Enter the
                                     amount");
                double amount = sc.nextDouble();
                s.withdraw(amount);
                }
            else if (pref == 3) {
                        s.displayBalance();
                }
            else if ( pref == 4) {
                    s.computeInterest();
            }
            else if (pref == 5) {
                    s.compoundInterest(0.05, 1);
            }
            else break;
        }
    }
```

```
else if (accountType
  == 2) {
System.out
  .println("Enter
  4 for MinBalance");
CurAccount s = new
 CurAccount(name,
 accno)
while (true) {
 int pref = sc.nextInt();
  if (pref == 1) {
     System.out
   .println("Enter
 the amount");
 double amount =
   sc.nextDouble();
  s.deposit(amount);
   }
 else if (pref == 2) {
   System
   double amount
     = sc.nextDouble()
 } s.deposit(amount);
```

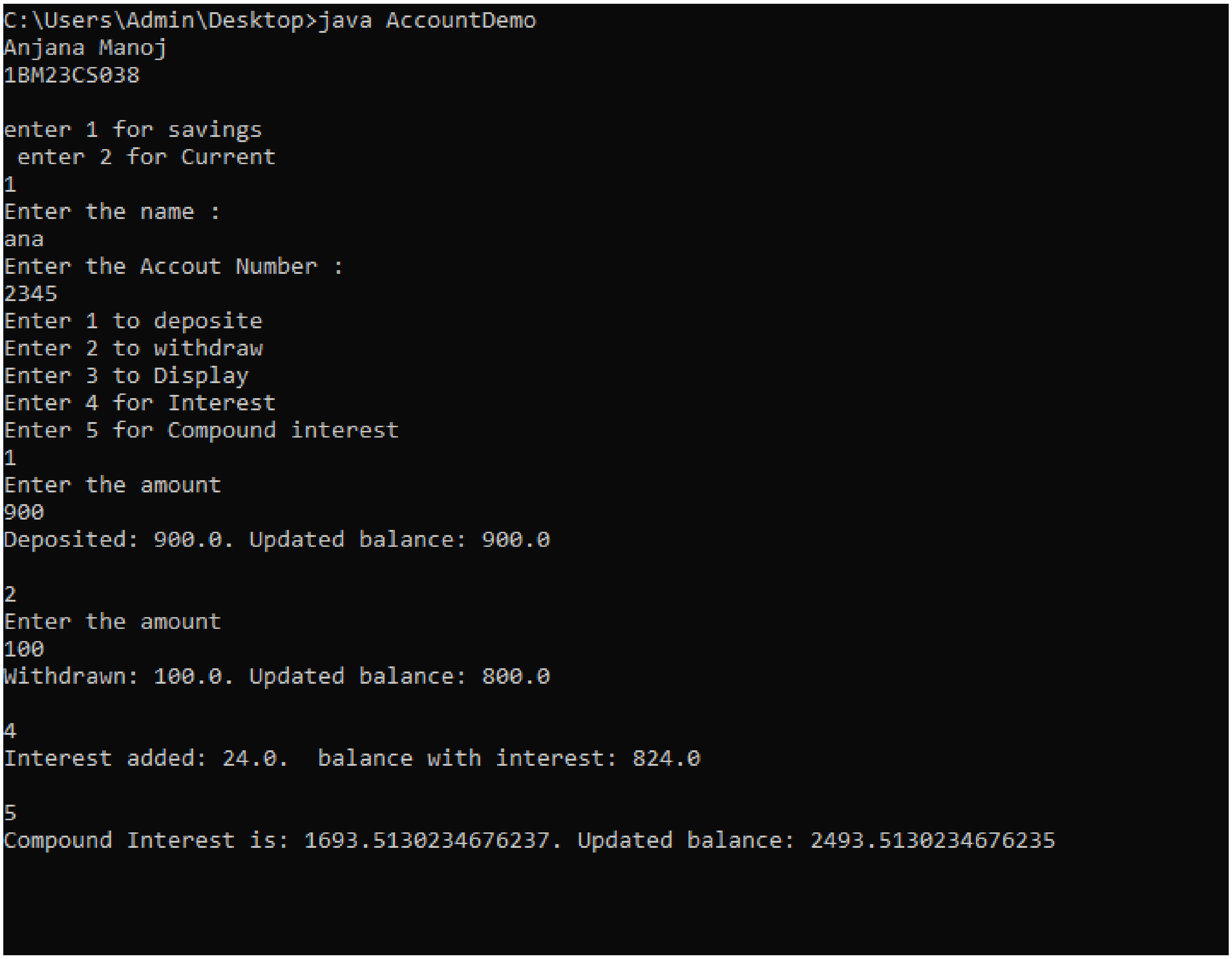


```
O/P
Enter 1 for savings
Enter 2 for current
1
Enter name
ana
Enter account number
2345
Enter 1 to deposit
Enter 2 to withdraw
Enter 3 to display
Enter 4 for interest
Enter 5 for compound interest
1
Enter amount
900
Deposited 900.0 updated balance 900.0
Enter amount 100
Withdrawn 100.0 updated balance 800.0
Account Balance = 800.0
Interest added 29.0
Balance with interest 829.0
5
compound interest is
1693.5130239
Updated balance:
2493.5130234
```

Code:

```
import java.util.Scanner;

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

    Account(String name, String accNumber, String accType) {
        customerName = name;
        accountNumber = accNumber;
        accountType = accType;
        balance = 0;
    }
    void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount + ". Updated balance: " + balance);
    }

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

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

class SavAccount extends Account {

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

    void computeInterest() {
        double interest = balance * 0.03;
        System.out.println("Interest added: " + interest + ".  balance with interest: " +
(balance+interest));
    }

    void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawn: " + amount + ". Updated balance: " + balance);
        } else {
            System.out.println("Insufficient balance.");
        }
    }
     void compoundInterest(double rate, int time) {
      // Formula for compound interest
        double ci = balance * Math.pow(1 + (rate / 365), 365 * time);
        System.out.println("Compound Interest is: " + ci + ". Updated balance: " + (balance+ci));
    }
}

class CurAccount extends Account {

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

    void checkMinBalance() {
        if (balance < 5000) {
            balance -= 300;
            System.out.println("Balance below minimum. Service charge imposed: 300"  + ". Updated
balance: " + balance);
        }
    }

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

class AccountDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
while(true){
System.out.println("enter 1 for savings \n enter 2 for Current ");
int accountType = sc.nextInt();
if(accountType>=3) break;
String bank = sc.nextLine();
System.out.println("Enter the name : ");
String name = sc.nextLine();
System.out.println("Enter the Accout Number : ");
String accno = sc.nextLine();
System.out.println("Enter 1 to deposit");
System.out.println("Enter 2 to withdraw");
System.out.println("Enter 3 to Display");
if(accountType==1){
System.out.println("Enter 4 for Interest");
System.out.println("Enter 5 for Compound interest");
SavAccount s = new SavAccount(name,accno);
while(true){
int pref = sc.nextInt();
if(pref==1){
System.out.println("Enter the amount ");
double amount = sc.nextDouble();
s.deposit(amount);
}
else if(pref==2){
System.out.println("Enter the amount ");
double amount = sc.nextDouble();
s.withdraw(amount);
}
else if(pref==3){
s.displayBalance();
}
else if(pref==4){
s.computeInterest();
}
else if(pref==5){
s.compoundInterest(0.05,15);
}
else break;
}
}
else if(accountType==2){
System.out.println("Enter 4 for checkMinBalance");
CurAccount s = new CurAccount(name,accno);
while(true){
int pref = sc.nextInt();
if(pref==1){
System.out.println("Enter the amount ");
double amount = sc.nextDouble();
s.deposit(amount);
}
else if(pref==2){
System.out.println("Enter the amount ");
double amount = sc.nextDouble();
s.withdraw(amount);
}
else if(pref==3){
s.displayBalance();
}
else if(pref==4){
s.checkMinBalance();
}
else break;
}
}
else break;
```

```
    }
}
}
```

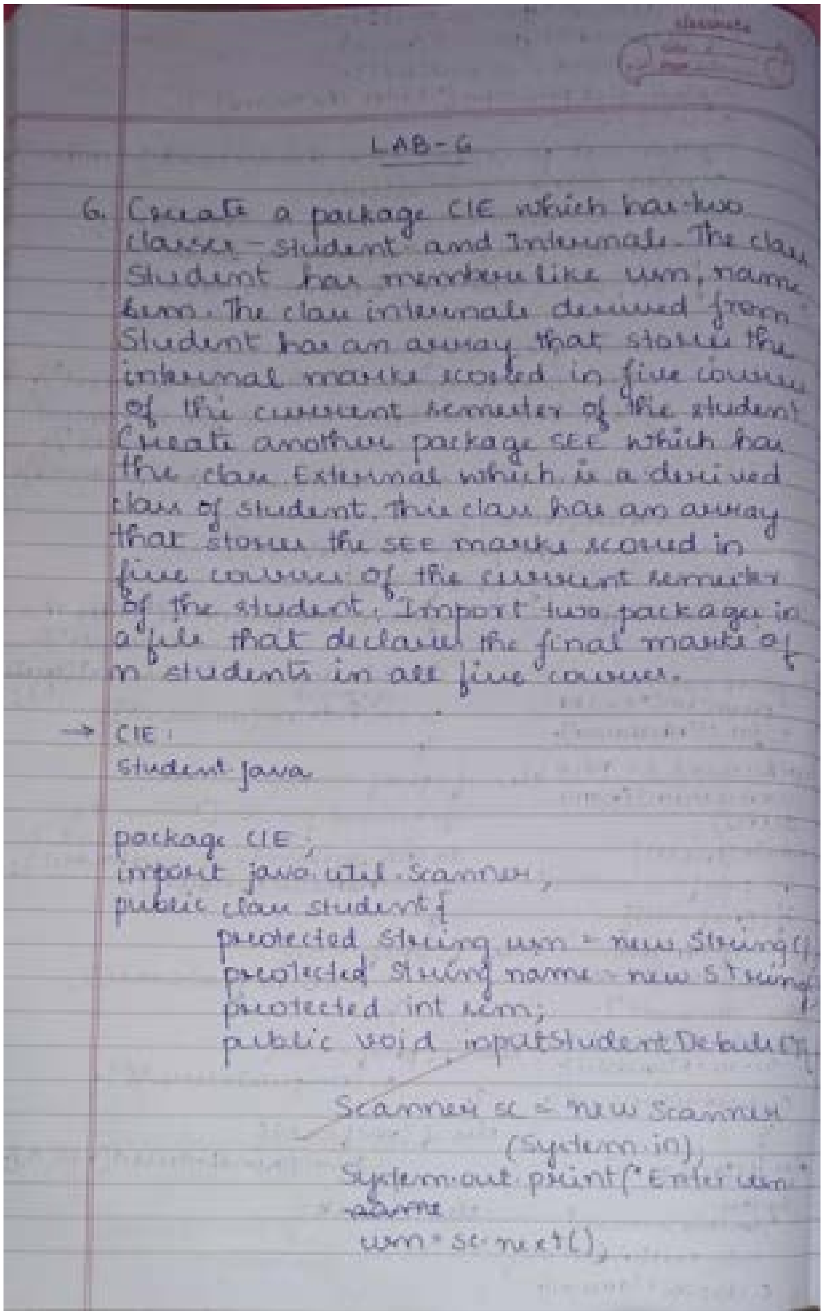


```
C:\Users\Admin\Desktop>java AccountDemo
Anjana Manoj
1BM23CS038

enter 1 for savings
 enter 2 for Current
1
Enter the name :
ana
Enter the Accout Number :
2345
Enter 1 to deposite
Enter 2 to withdraw
Enter 3 to Display
Enter 4 for Interest
Enter 5 for Compound interest
1
Enter the amount
900
Deposited: 900.0. Updated balance: 900.0

2
Enter the amount
100
Withdrawn: 100.0. Updated balance: 800.0

4
Interest added: 24.0.  balance with interest: 824.0

5
Compound Interest is: 1693.5130234676237. Updated balance: 2493.5130234676235
```

<u>Program 6</u>

Demonstration of Package

Algorithm:



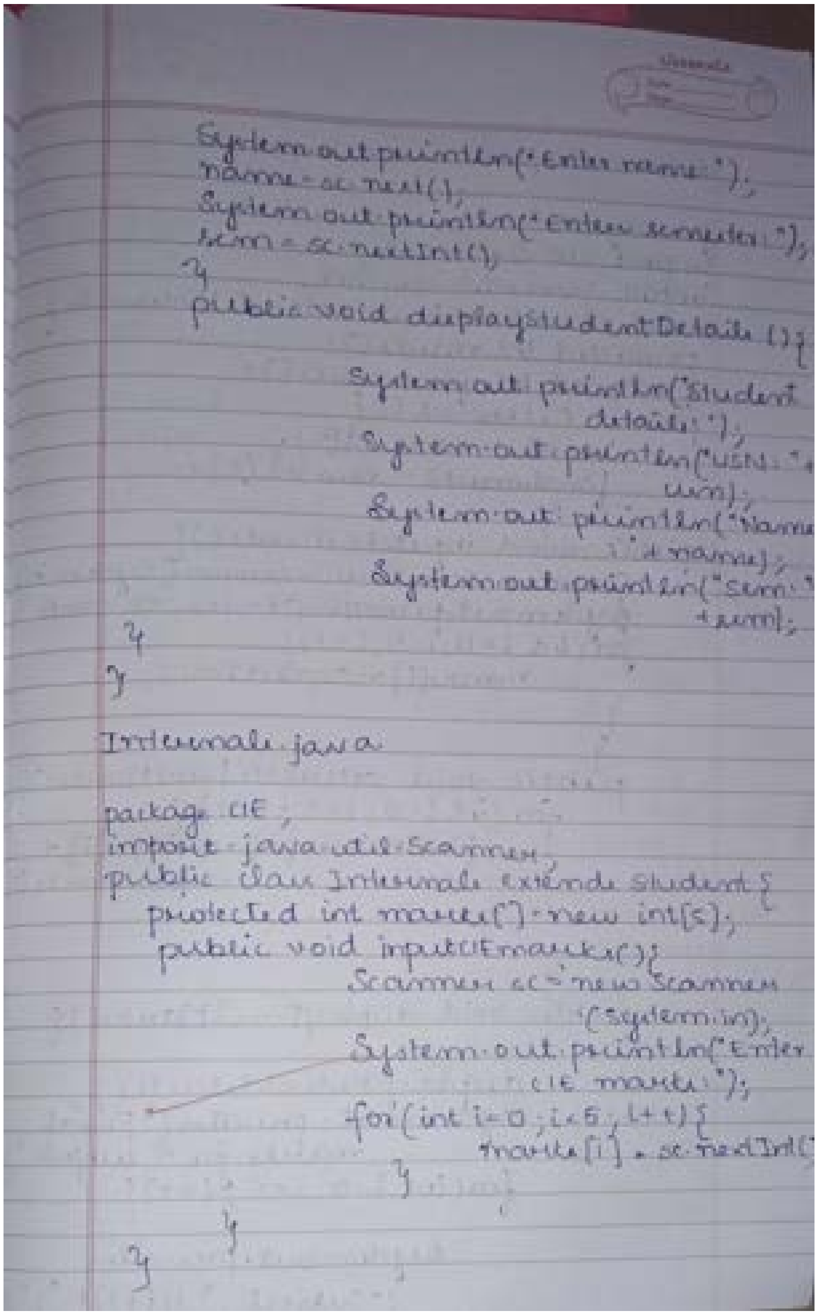
LAB-6

6. 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 two packages in a file that declares the final marks of n students in all five courses.

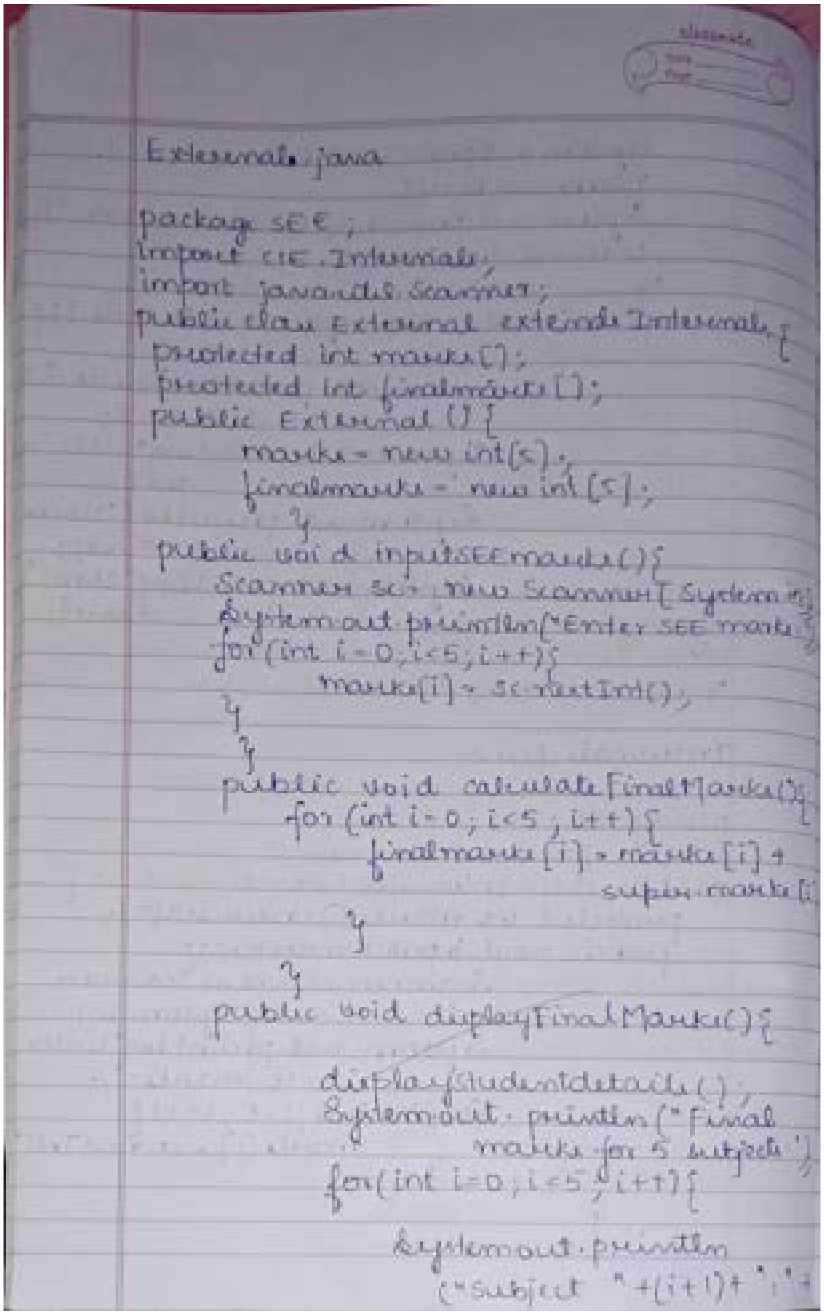
→ CIE :

Student.java

```
package CIE;
import java.util.Scanner;
public class student{
        protected String usn = new String(
        protected String name = new String
        protected int sem;
        public void inputStudentDetails()

                Scanner sc = new Scanner
                          (System.in);
                System.out.print("Enter usn"
                ~~name~~
                usn = sc.next();
```

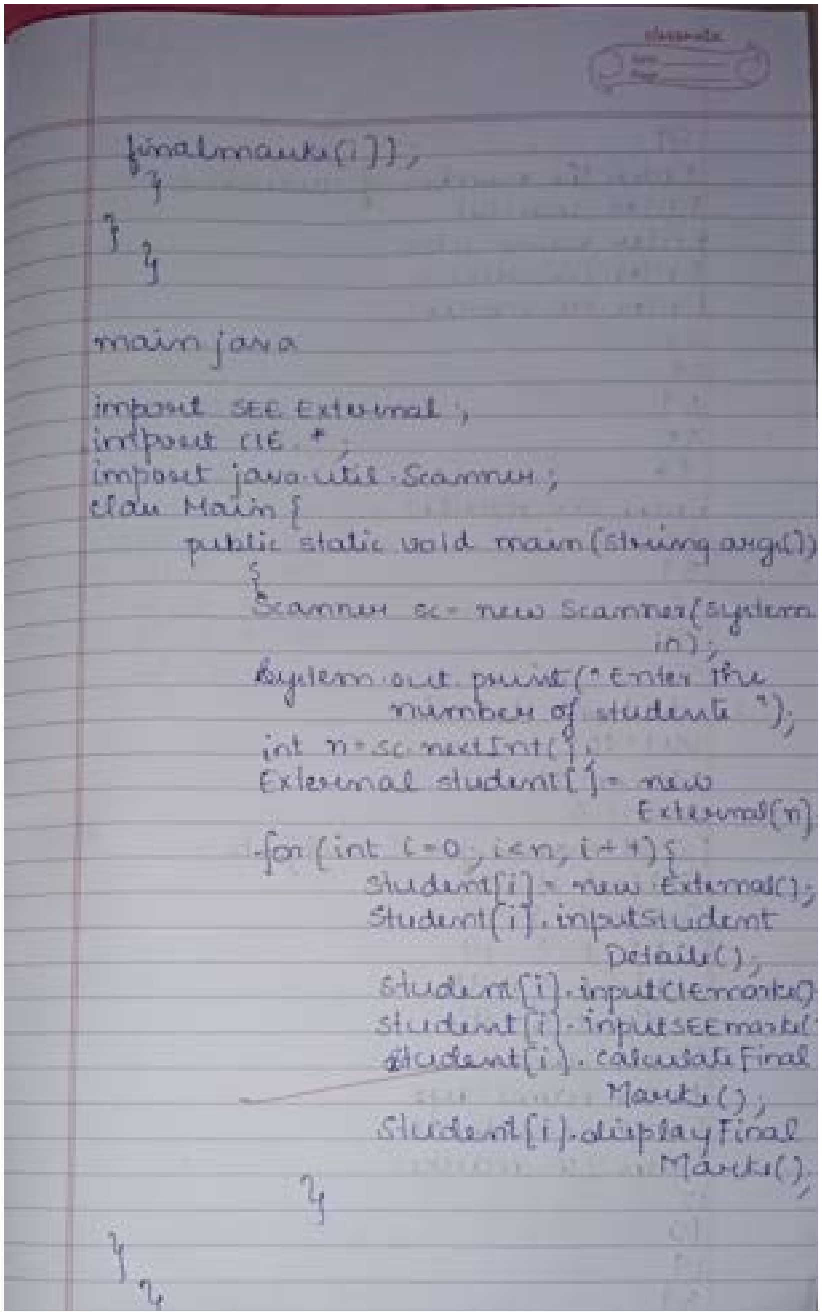


```
        System.out.println("Enter name:");
        name=sc.next();
        System.out.println("Enter semester:");
        sem=sc.nextInt();
      }
      public void displayStudentDetails(){
                    System.out.println("Student
                                  details:");
                    System.out.println("USN: "+
                                      usn);
                    System.out.println("Name
                                  "+name);
                    System.out.println("Sem: "
                                      +sem);
  }
}

Internals.java

package CIE;
import java.util.Scanner;
public class Internals extends Student{
    protected int marks[]=new int[5];
     public void inputCIEmarks(){
                    Scanner sc=new Scanner
                              (System.in);
                    System.out.println("Enter
                              CIE marks:");
                    for(int i=0;i<5;i++){
                          marks[i]=sc.nextInt(
                       }
          }
  }
```

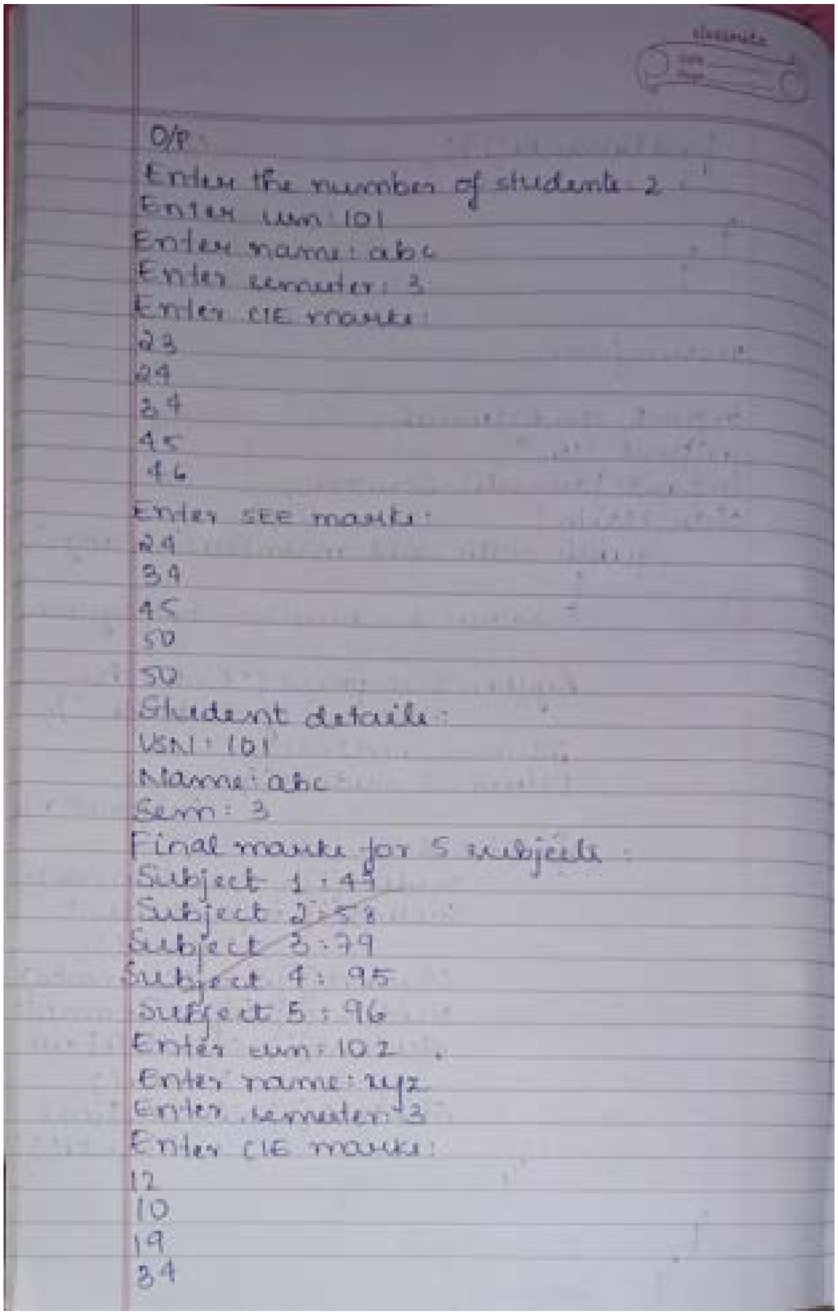


External. java

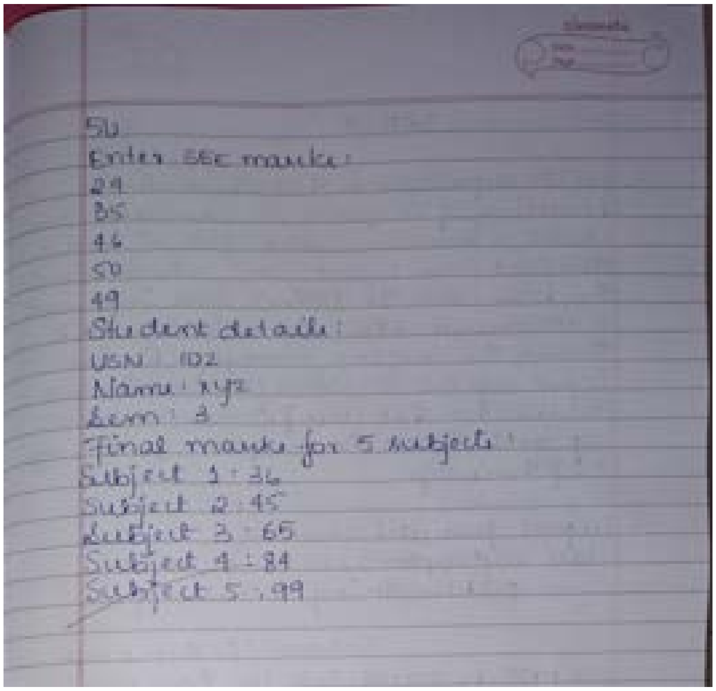
```
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class External extends Internals{
 protected int marks[];
 protected int finalmarks[];
 public External(){
        marks = new int[5];
        finalmarks = new int[5];
        }
 public void inputSEEmarks(){
    Scanner sc = new Scanner[System.in];
    System.out.println("Enter SEE marks
    for(int i=0;i<5;i++){
             marks[i] = sc.nextInt();
    }
   }
   public void calculateFinalMarks(){
       for (int i=0; i<5 , i++){
              finalmarks[i] = marks[i] +
                              super.marks[i
            }
       }
   public void displayFinalMarks(){

              displaystudentdetails();
              System.out.println("Final
                         marks for 5 subjects");
              for(int i=0;i<5;i++){

                  System.out.println
                    ("Subject "+(i+1)+":"+
```



```
finalmarks[i]};
   }
}
  }

main java

import SEE.External;
import CIE.*;
import java.util.Scanner;
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();
           External student[]= new
                               External(n)
           for(int i=0; i<n; i++){
                  Student[i]= new External();
                  Student[i].inputStudent
                              Details();
                  Student[i].inputCIEmarks()
                  student[i].inputSEEmarks(
                  Student[i].calculateFinal
                              Marks();
                  Student[i].displayFinal
                              Marks();
           }
}
  }
```



```
O/P:
Enter the number of students: 2
Enter usn: 101
Enter name: abc
Enter semester: 3
Enter CIE marks:
23
24
34
45
46
Enter SEE marks:
24
34
45
50
50
Student details:
USN: 101
Name: abc
Sem: 3
Final marks for 5 subjects:
Subject 1: 49
Subject 2: 58
Subject 3: 79
Subject 4: 95
Subject 5: 96
Enter usn: 102
Enter name: xyz
Enter semester: 3
Enter CIE marks:
12
10
19
34
```



```
50
Enter SEE marks:
29
35
46
50
49
Student details:
USN: 102
Name: xyz
Sem: 3
Final marks for 5 subjects:
Subject 1 : 36
Subject 2 : 45
Subject 3 : 65
Subject 4 : 84
Subject 5 : 99
```

Code:

1. CIE:
    - Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
protected int marks[]=new int[5];
public void inputCIEmarks(){
Scanner sc=new Scanner(System.in);
System.out.println("Enter CIE marks:");
for(int i=0;i<5;i++){
marks[i]=sc.nextInt();
}
}}
```

    - Student.java

```
package CIE;
import java.util.Scanner;
public class Student{
protected String usn=new String();
protected String name=new String();
protected int sem;
public void inputStudentDetails(){
```

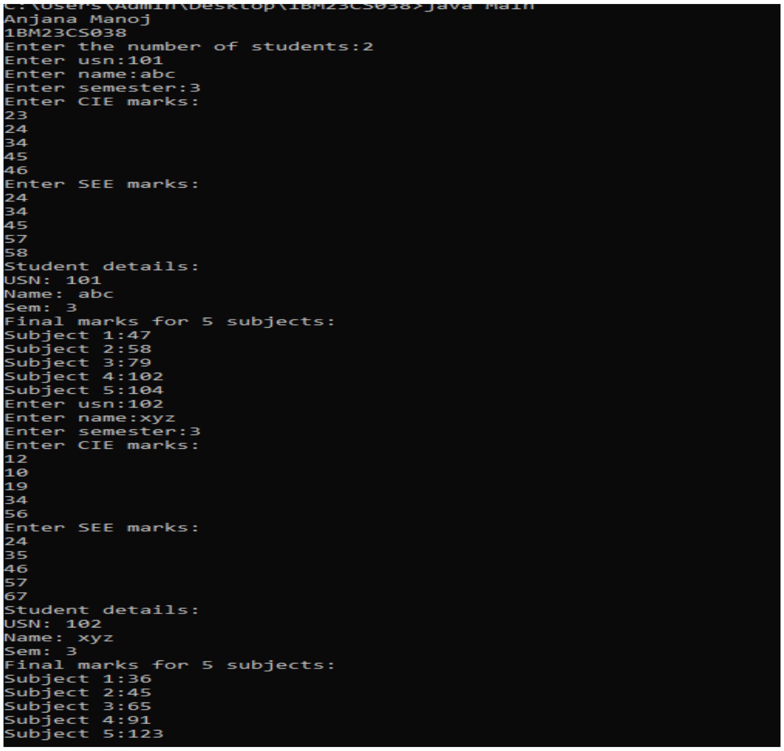
```
Scanner sc=new Scanner(System.in);
System.out.print("Enter usn:");
usn=sc.next();
System.out.print("Enter name:");
name=sc.next();
System.out.print("Enter semester:");
sem=sc.nextInt();
}
public void displayStudentDetails(){
System.out.println("Student details:");
System.out.println("USN: "+usn);
System.out.println("Name: "+name);
System.out.println("Sem: "+sem);

}

}
```

2. SEE:
    - External.java

```
package SEE;
import CIE.Internals;

import java.util.Scanner;
public class External extends Internals{
protected int marks[];
protected int finalmarks[];
public External(){
marks=new int[5];
finalmarks=new int[5];
}
public void inputSEEmarks() {

Scanner sc = new Scanner(System.in);
System.out.println("Enter SEE marks:");
for(int i=0;i<5;i++){
marks[i]=sc.nextInt();
}

}
public void calculateFinalMarks() {
for(int i=0;i<5;i++){
finalmarks[i]=marks[i]+super.marks[i];
}
}
public void displayFinalMarks() {

displayStudentDetails();
System.out.println("Final marks for 5 subjects:");
for(int i=0;i<5;i++){
System.out.println("Subject "+(i+1)+":"+finalmarks[i]);
}

}}
```

3 .Main.java

```
import SEE.External;
import CIE.*;
import java.util.Scanner;
class Main{
public static void main(String args[]){
System.out.println("Anjana Manoj\n1BM23CS038");
Scanner sc=new Scanner(System.in);
System.out.print("Enter the number of students:");
int n=sc.nextInt();
External student[]=new External[n];
for(int i=0;i<n;i++){
student[i]=new External();
student[i].inputStudentDetails();

student[i].inputCIEmarks();
student[i].inputSEEmarks();
student[i].calculateFinalMarks();
student[i].displayFinalMarks();
}

}}
```

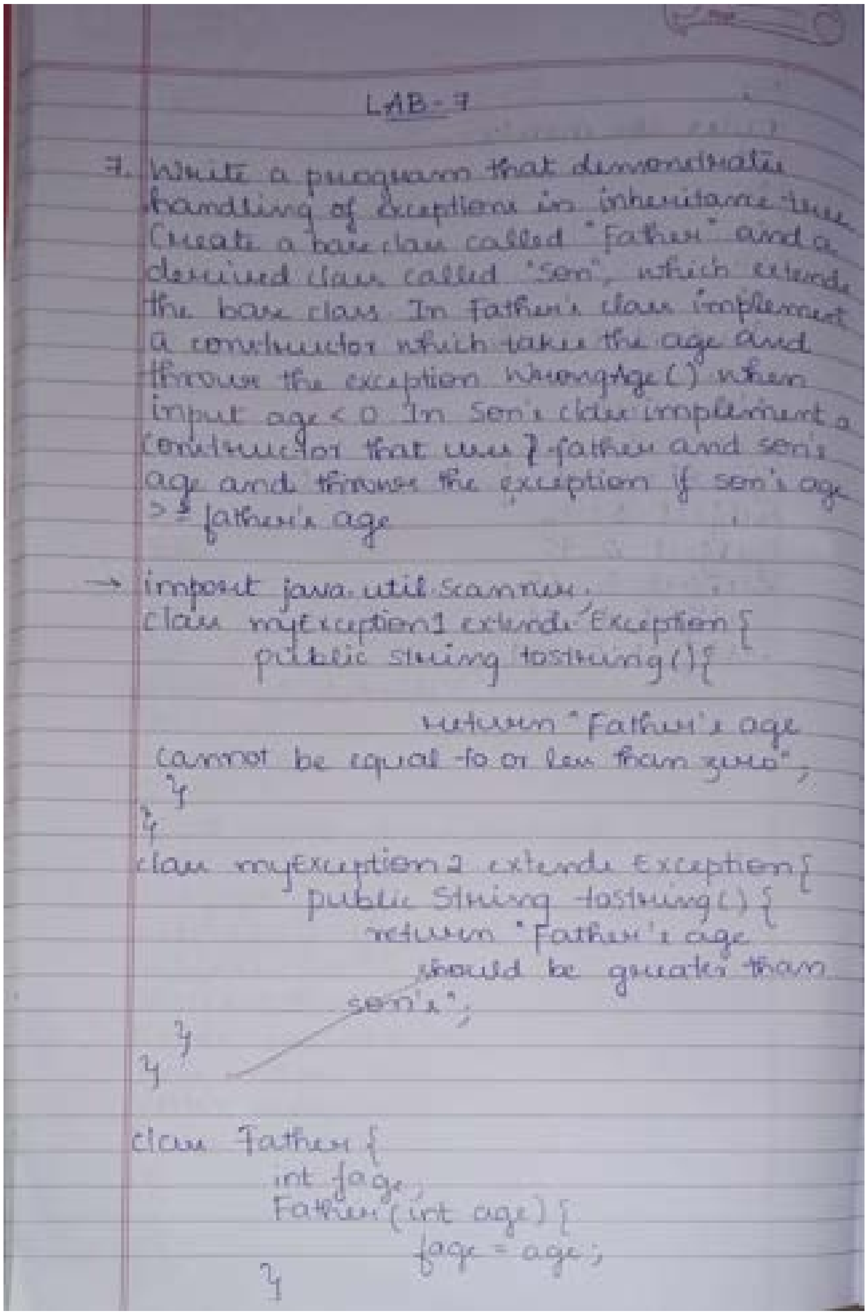


```
C:\Users\Admin\Desktop\1BM23CS038>java Main
Anjana Manoj
1BM23CS038
Enter the number of students:2
Enter usn:101
Enter name:abc
Enter semester:3
Enter CIE marks:
23
24
34
45
46
Enter SEE marks:
24
34
45
57
58
Student details:
USN: 101
Name: abc
Sem: 3
Final marks for 5 subjects:
Subject 1:47
Subject 2:58
Subject 3:79
Subject 4:102
Subject 5:104
Enter usn:102
Enter name:xyz
Enter semester:3
Enter CIE marks:
12
10
19
34
56
Enter SEE marks:
24
35
46
57
67
Student details:
USN: 102
Name: xyz
Sem: 3
Final marks for 5 subjects:
Subject 1:36
Subject 2:45
Subject 3:65
Subject 4:91
Subject 5:123
```

Program 7

Demonstration of Exception Handling

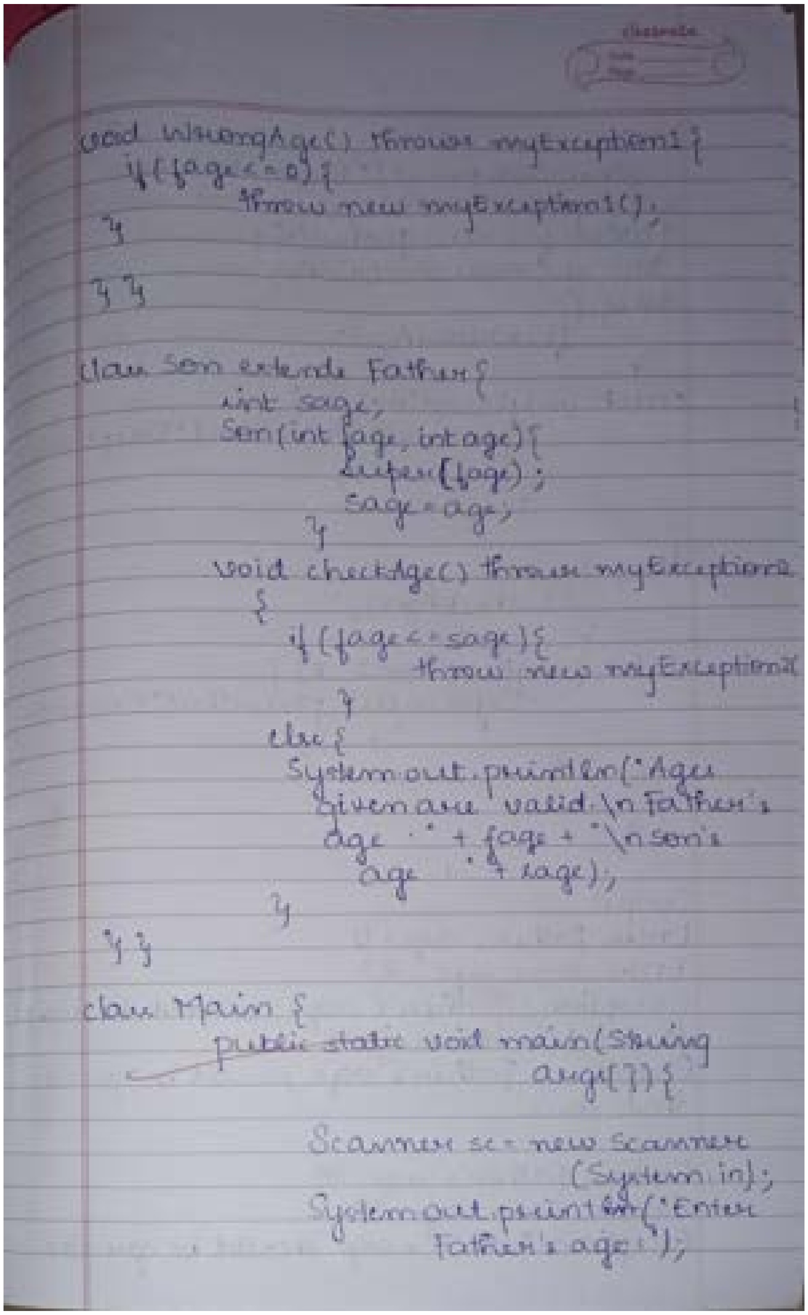
Algorithm:



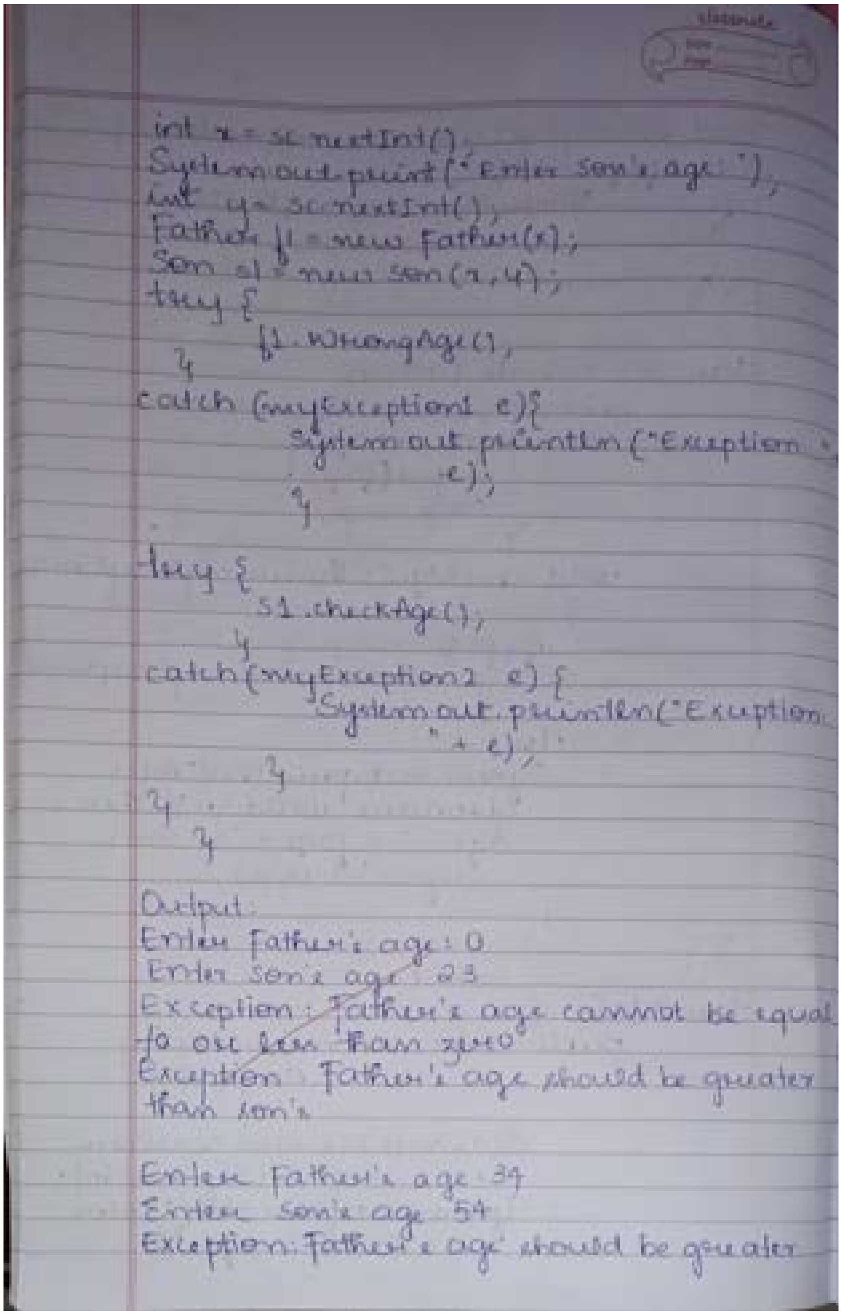
LAB-7

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

```
→ import java.util.Scanner;
class myexception1 extends Exception {
        public string tostring(){

                    return "Father's age
 cannot be equal to or less than zero";
 }
}
class myException2 extends Exception{
            public String tostring() {
                  return "Father's age
                        should be greater than
                son's";
  }
}

class Father {
        int fage;
        Father(int age) {
                fage = age;
        }
```



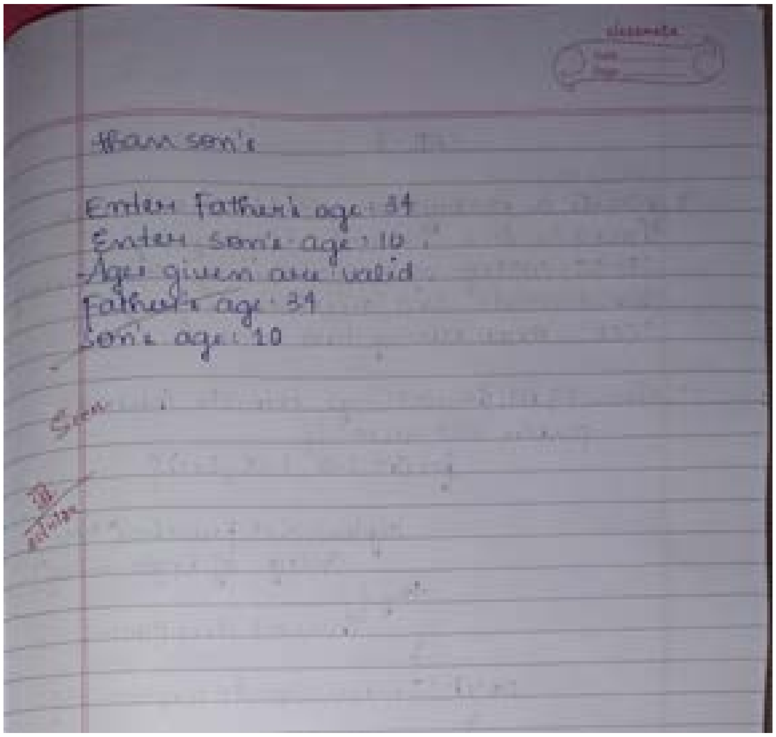
```
void wrongAge() throws myException1{
  if(fage<=0){
        throw new myException1();
  }

}}

class Son extends Father{
      int sage;
      Son(int fage, int age){
             super(fage);
             sage=age;
         }
      void checkAge() throws myException2
       {
         if(fage<=sage){
               throw new myException2(
           }
        else{
         System.out.println("Ages
          Given are valid \n Father's
           age : " + fage + "\n Son's
            age : " + sage);
         }
 }}
class Main {
      public static void main(String
                          args[]){

           Scanner sc= new Scanner
                          (System.in);
           System.out.print~~ln~~("Enter
                   Father's age:");
```



```
int x = sc.nextInt();
System.out.print("Enter Son's age ");
int y = sc.nextInt();
Father f1 = new Father(x);
Son s1 = new Son(x,y);
try {
        f1.WrongAge();
  }
catch (myException1 e){
            System.out.println("Exception "+
                        e);
        }

try {
        S1.checkAge();
    }
catch(myException2 e) {
            System.out.println("Exception
                        " + e);
        }
}
  }
```

Output:

```
Enter Father's age: 0
Enter Son's age: 23
Exception: Father's age cannot be equal
to or less than zero
Exception: Father's age should be greater
than son's

Enter Father's age 34
Enter Son's age 54
Exception: Father's age should be greater
```



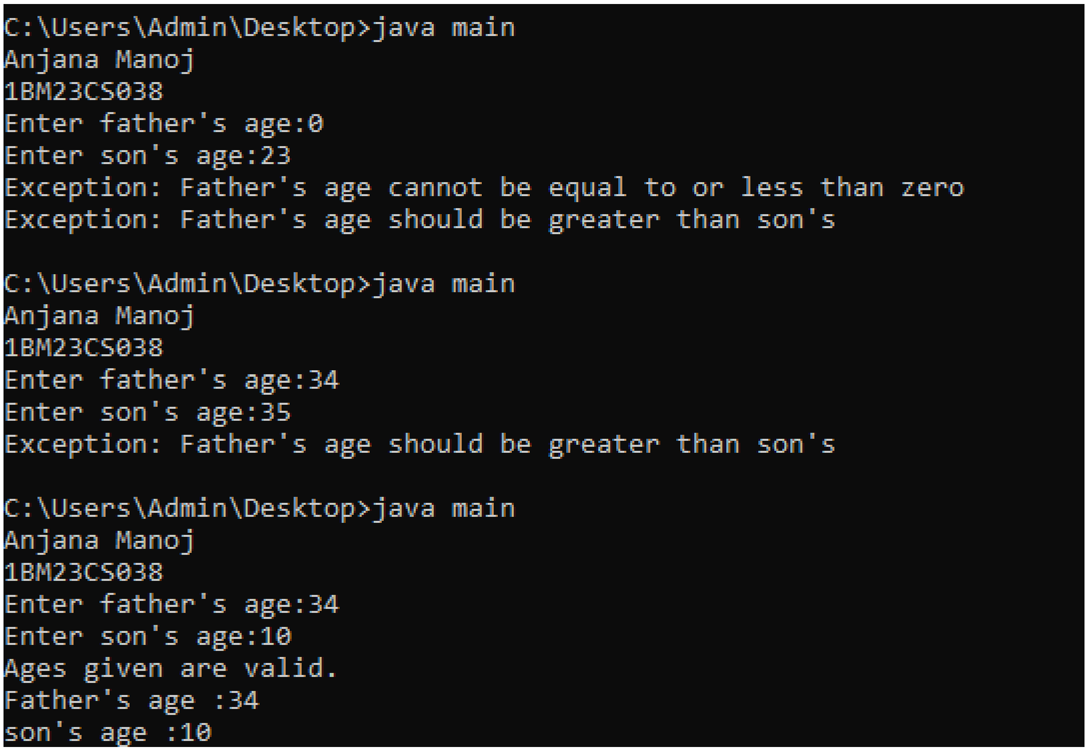
than son's

```
Enter father's age: 34
Enter son's age: 10
Ages given are valid
Father's age: 34
Son's age: 10
```

Code:

```
import java.util.Scanner;
class myException1 extends Exception{

public String toString(){
return "Father's age cannot be equal to or less than zero";
}
}
class myException2 extends Exception{
public String toString(){
return "Father's age should be greater than son's";
}
}
class Father{
int fage;
Father(int age){
fage=age;
}

void WrongAge() throws myException1 {
if(fage<=0){
throw new myException1();
}

}}
```

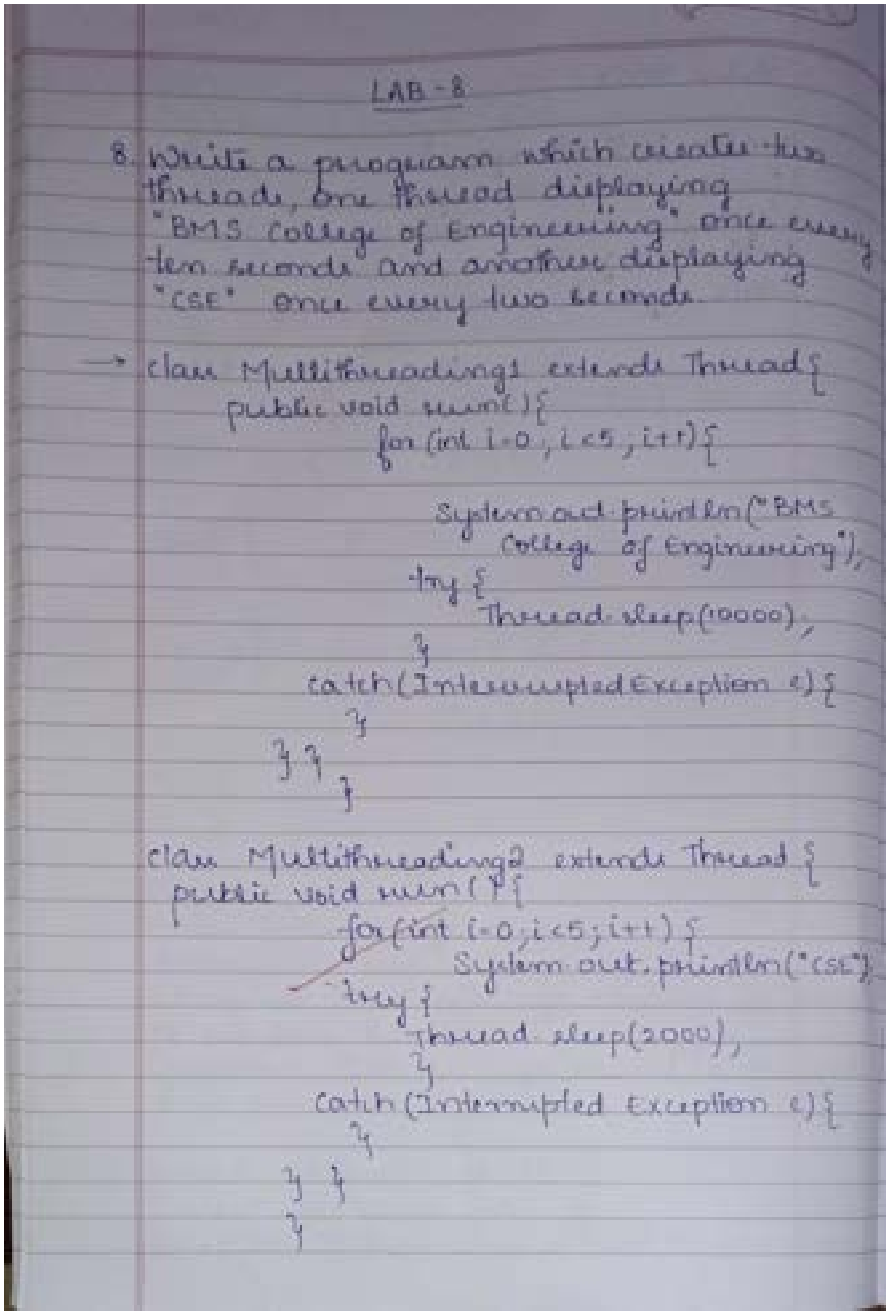
```
class Son extends Father{
int sage;
Son(int fage,int age){
super(fage);
sage=age;

}
void checkAge() throws myException2{
if(fage<=sage){
throw new myException2();
}
else{
System.out.println("Ages given are valid.\nFather's age :" +fage+"\nson's age :" +sage);
}
}}
class main{
public static void main(String args[]){
Scanner sc=new Scanner(System.in);
System.out.println("Anjana Manoj\n1BM23CS038");
System.out.print("Enter father's age:");
int x=sc.nextInt();
System.out.print("Enter son's age:");
int y=sc.nextInt();
Father f1=new Father(x);
Son s1=new Son(x,y);
try{
 f1.WrongAge();

}
catch(myException1 e){
System.out.println("Exception: "+e) ;
}
try{
 s1.checkAge();
}
catch(myException2 e){
System.out.println("Exception: "+e);
}
}}
```



```
C:\Users\Admin\Desktop>java main
Anjana Manoj
1BM23CS038
Enter father's age:0
Enter son's age:23
Exception: Father's age cannot be equal to or less than zero
Exception: Father's age should be greater than son's

C:\Users\Admin\Desktop>java main
Anjana Manoj
1BM23CS038
Enter father's age:34
Enter son's age:35
Exception: Father's age should be greater than son's

C:\Users\Admin\Desktop>java main
Anjana Manoj
1BM23CS038
Enter father's age:34
Enter son's age:10
Ages given are valid.
Father's age :34
son's age :10
```

Program 8
Demonstration of Threads

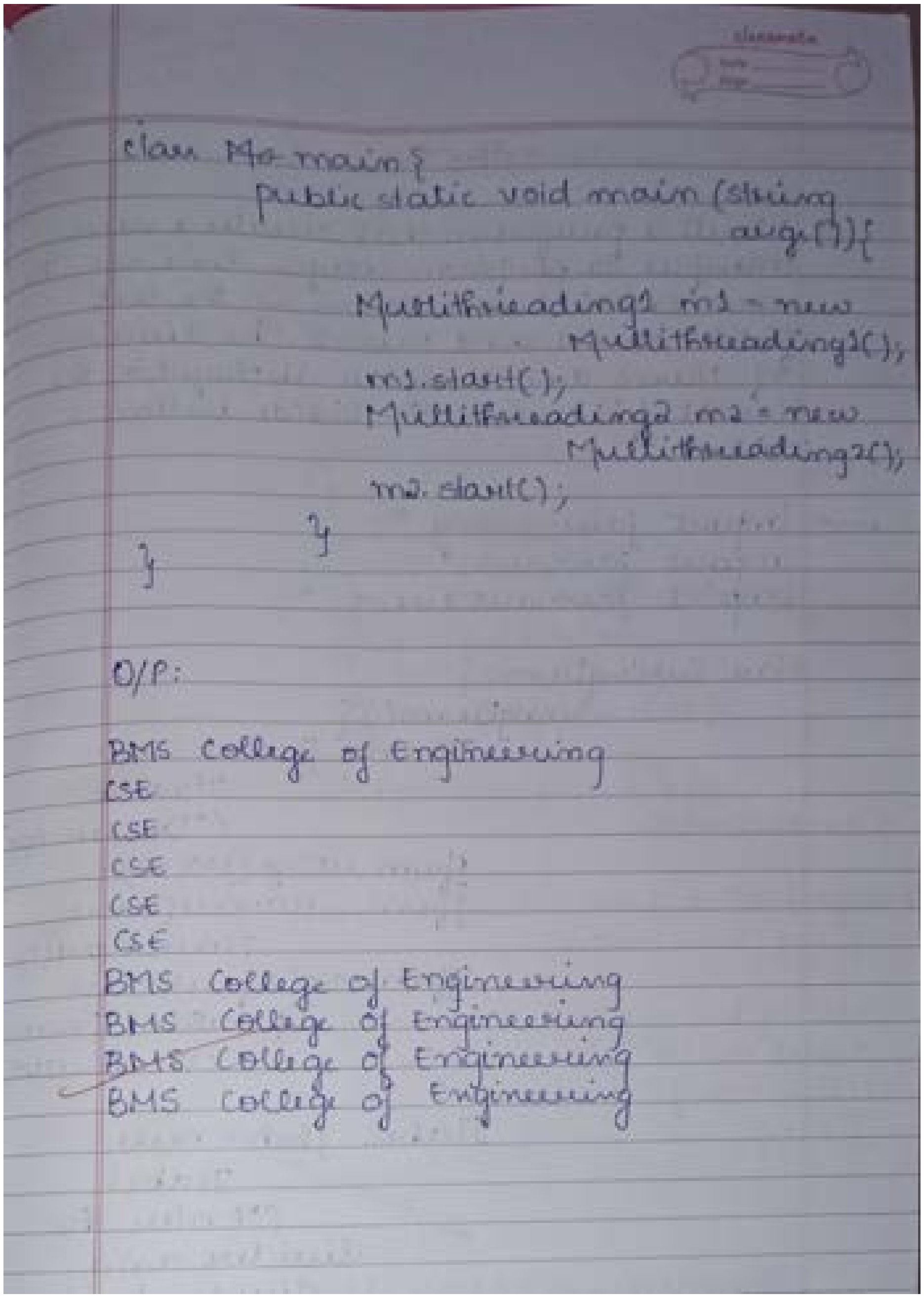
Algorithm:



LAB - 8

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

```
→ class Multithreading1 extends Thread{
      public void run(){
                for(int i=0; i<5; i++){

                      System.out.println("BMS
                           College of Engineering");
                try{
                      Thread.sleep(10000);
                }
           catch(InterruptedException e){
             }
        }}
          }

  class Multithreading2 extends Thread{
   public void run(){
             for(int i=0;i<5;i++){
                     System.out.println("CSE");
             try{
                 Thread.sleep(2000);
                 }
           catch(Interrupted Exception e){
             }
        } }
        }
```



```
class MTmain{
        public static void main(String
                                   args[]){
                Multithreading1 m1 = new
                              Multithreading1();
                m1.start();
                Multithreading2 m2 = new
                              Multithreading2();
                m2.start();
        }
}
```

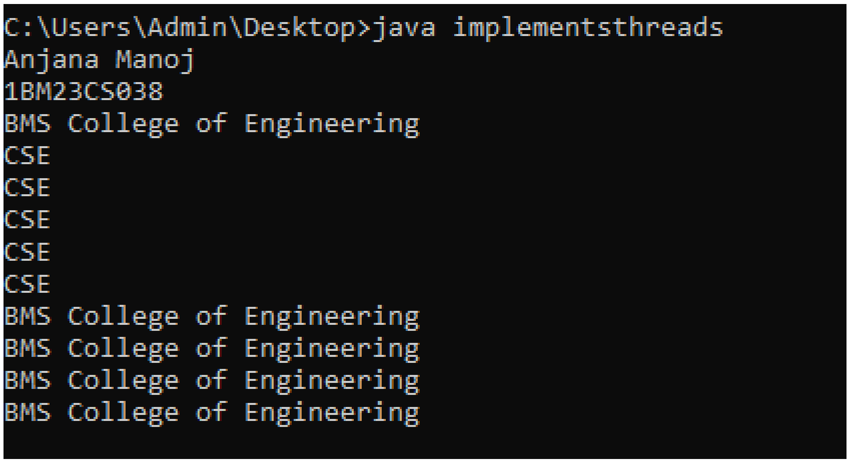
O/P:

```
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 Multithreading1 implements Runnable{
public void run(){
for(int i=0;i<5;i++){
  System.out.println("BMS College of Engineering");
  try{
  Thread.sleep(10000);
}
catch(InterruptedException e){
}
}}
}
```

```
class Multithreading2 implements Runnable{
public void run(){
for(int i=0;i<5;i++){
  System.out.println("CSE");
  try{
Thread.sleep(2000);
}
catch(InterruptedException e){
}
}}
}
class implementsthreads{
public static void main(String args[]){
System.out.println("Anjana Manoj\n1BM23CS038");
Multithreading1 m1=new Multithreading1();
Thread mt1=new Thread(m1);
mt1.start();
Multithreading2 m2=new Multithreading2();
Thread mt2=new Thread(m2);
mt2.start();
}
}
```

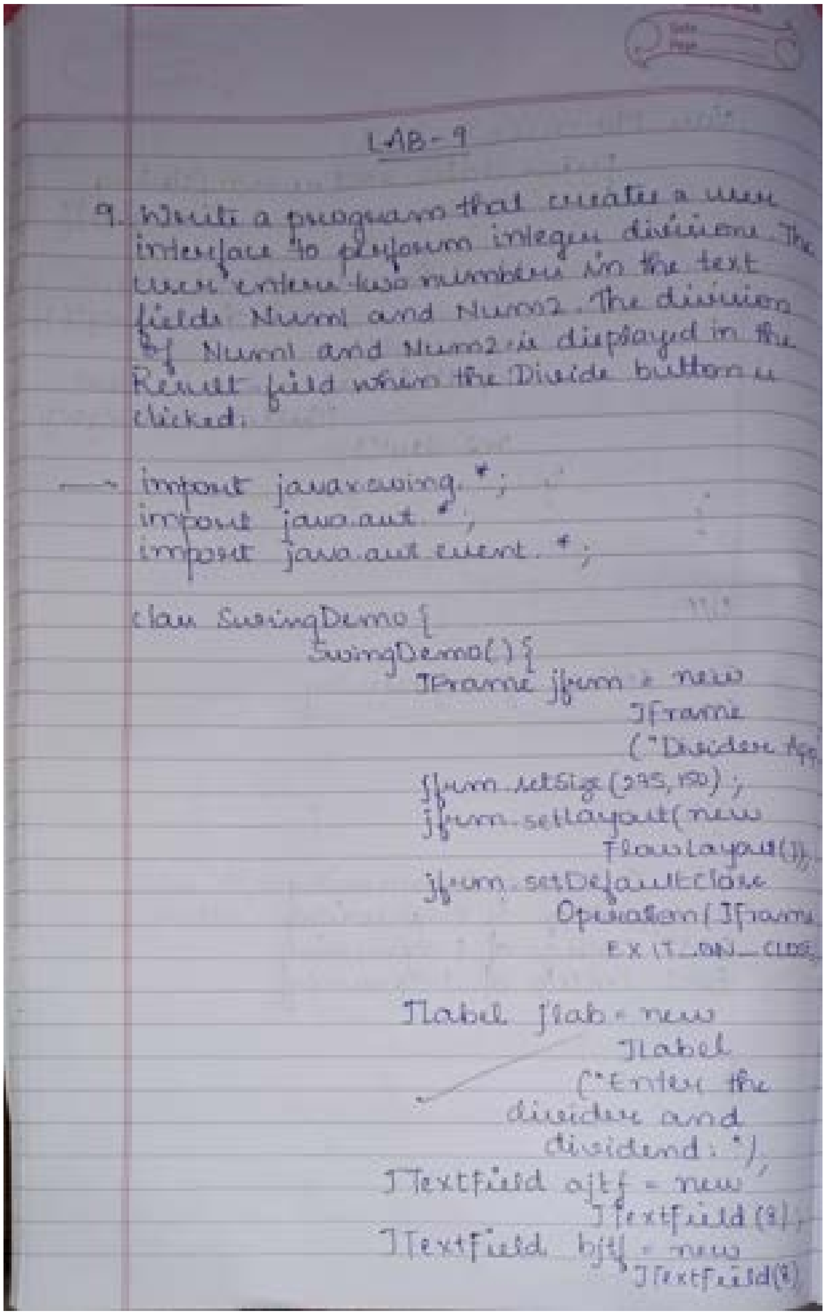


```
C:\Users\Admin\Desktop>java implementsthreads
Anjana Manoj
1BM23CS038
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
```

Program 9

User interface for division of numbers

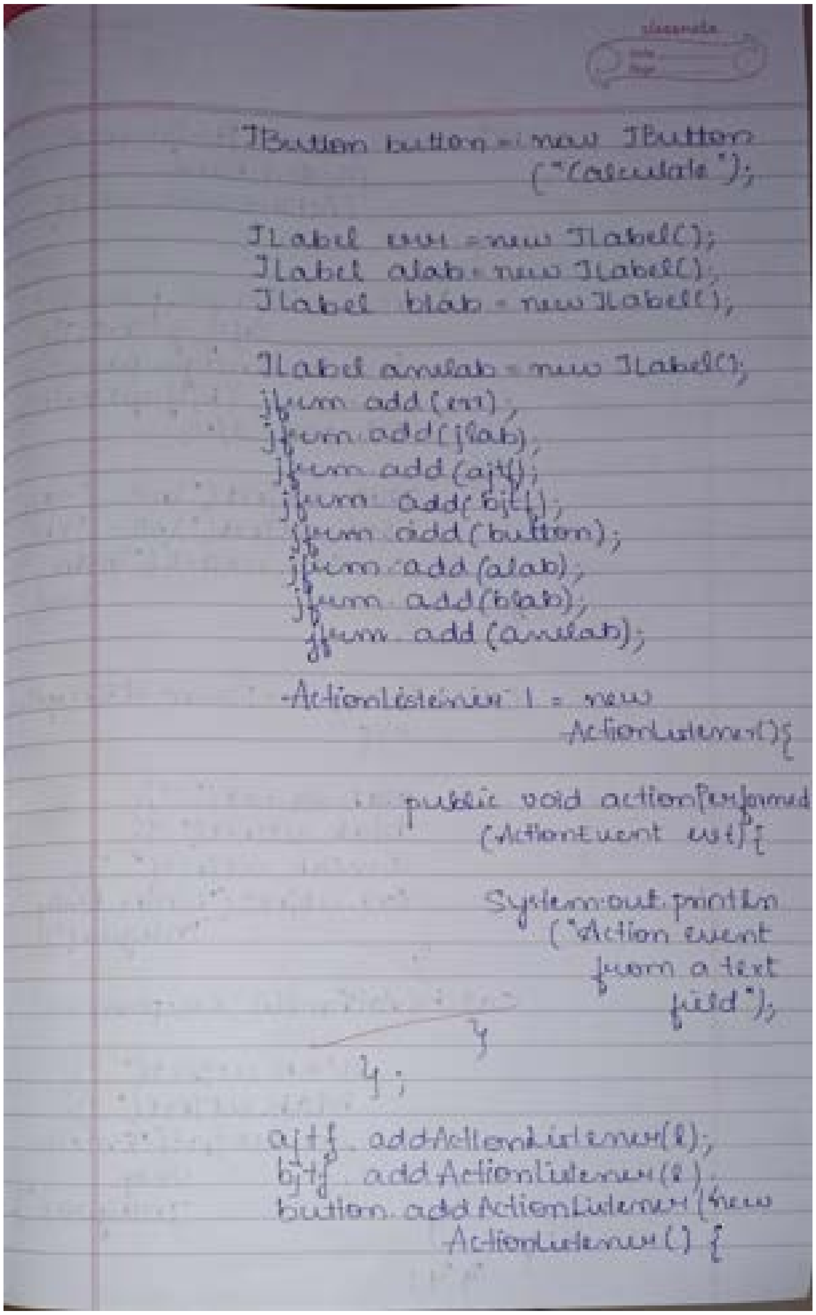
Algorithm:



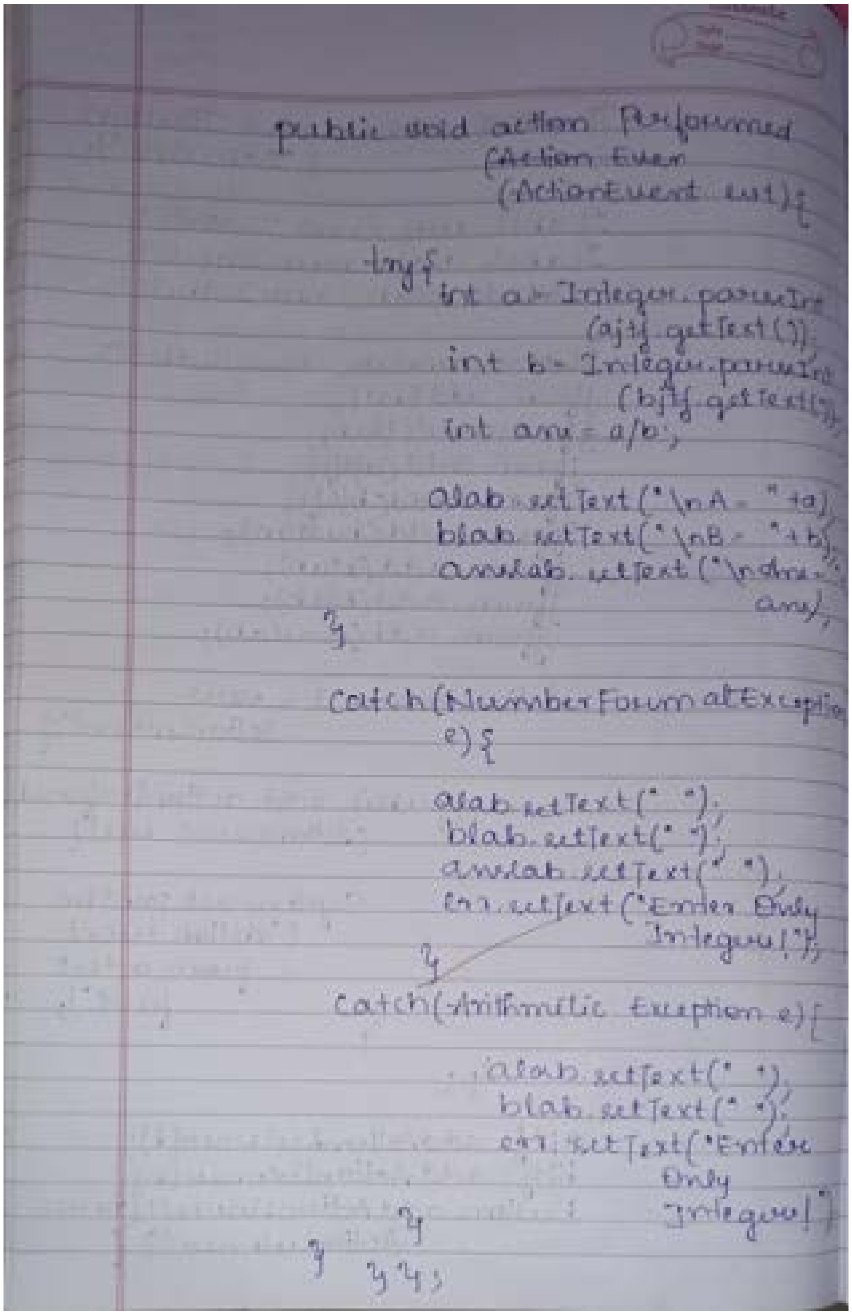
LAB-9

9. 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.

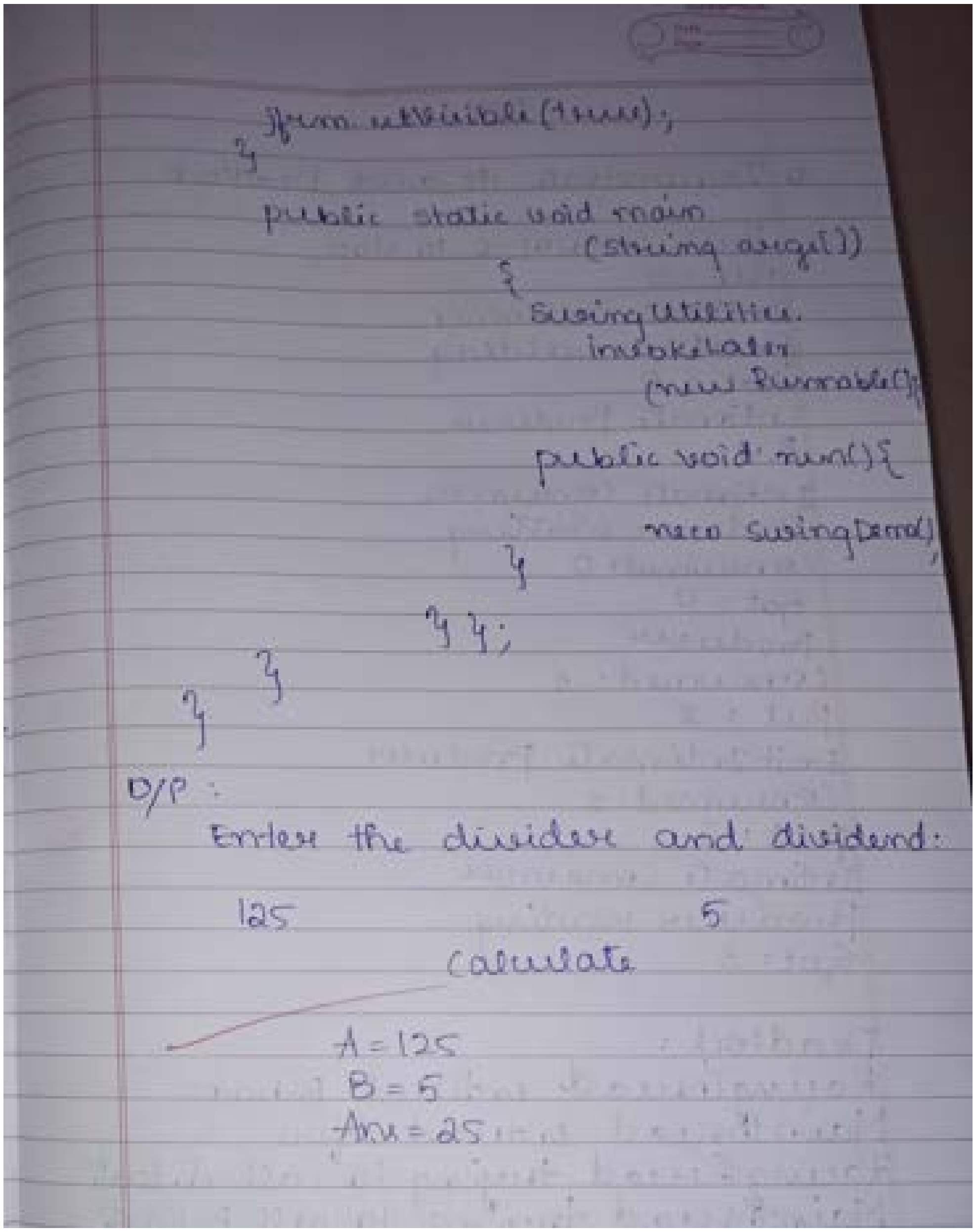
```
→ import javax.swing.*;
  import java.awt.*;
  import java.awt.event.*;

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

                 JLabel jlab = new
                              JLabel
                            ("Enter the
                      divider and
                          dividend:");
               JTextField ajtf = new
                              JTextField(8);
               JTextField bjtf = new
                              JTextField(8)
```



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

JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();

JLabel anslab = new JLabel();
jfrm.add(err);
jfrm.add(jlab);
 jfrm.add(ajtf);
 jfrm.add(bjtf);
  jfrm.add(button);
  jfrm.add(alab);
  jfrm.add(blab);
   jfrm.add(anslab);

 ActionListener l = new
              ActionListener(){

        public void actionPerformed
              (ActionEvent evt){

              System.out.println
                ("Action event
                  from a text
                     field");
          }
    };

ajtf.addActionListener(l);
bjtf.addActionListener(l),
button.addActionListener(new
          ActionListener() {
```



```
public void action Performed
          (Action Even
           (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(NumberFormatExceptio
          e){

        alab.setText(" ");
        blab.setText(" ");
        anslab.setText(" ");
        err.setText("Enter Only
                 Integer!");
      }
    catch(Arithmetic Exception e){

        alab.setText(" ");
        blab.setText(" ");
        err.setText("Enter
                 Only
      }          Integer!")
  }
   }};
```



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

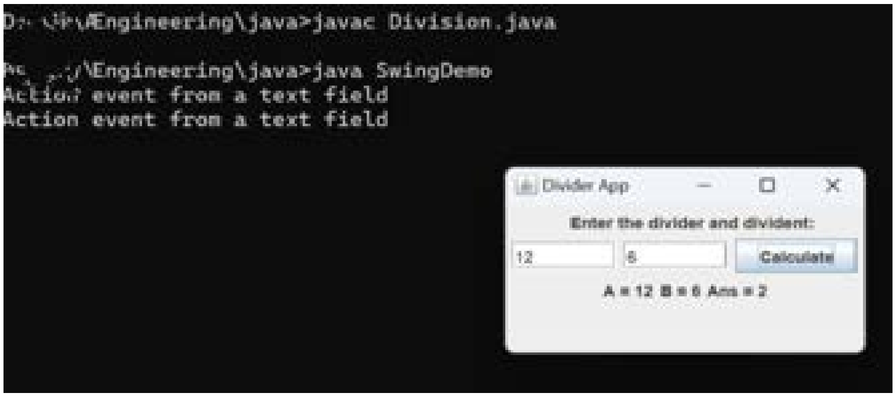
O/P :

```
Enter the divider and dividend:
125                          5
        Calculate
      A=125
      B=5
      Ans=25
```

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



```
D:\...\Engineering\java>javac Division.java

...\Engineering\java>java SwingDemo
Action event from a text field
Action event from a text field
```

Enter the divider and divident:

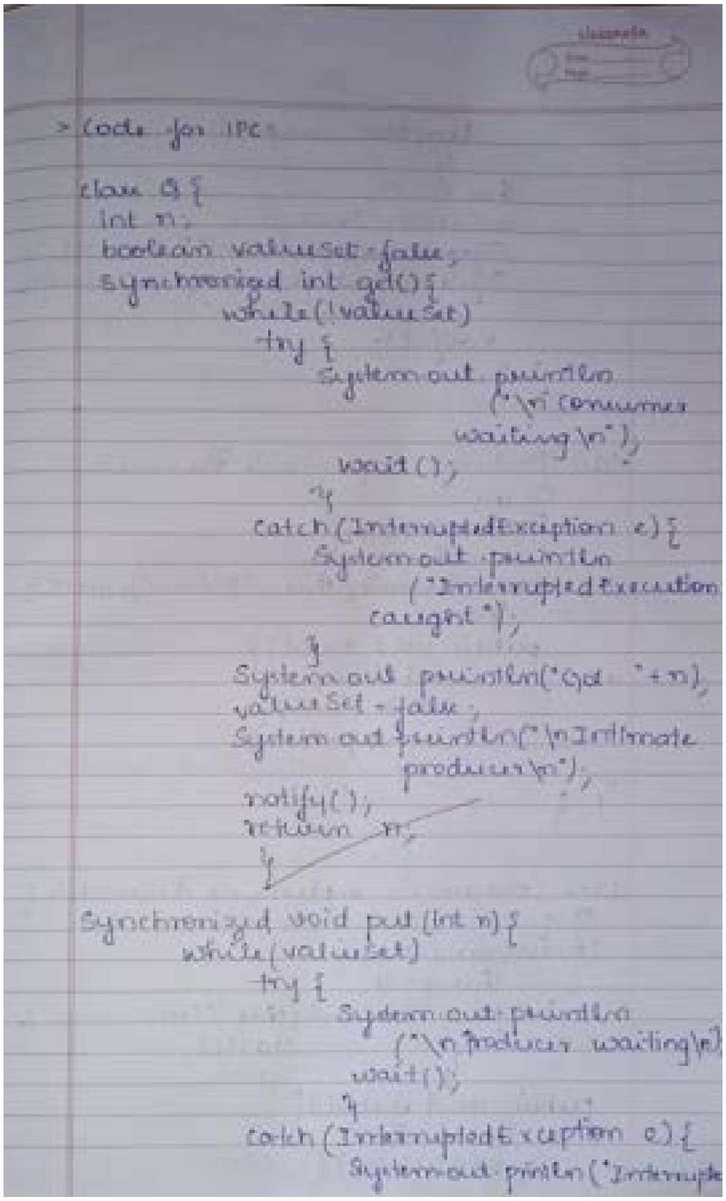
12 6

A = 12 B = 6 Ans = 2

Program 10

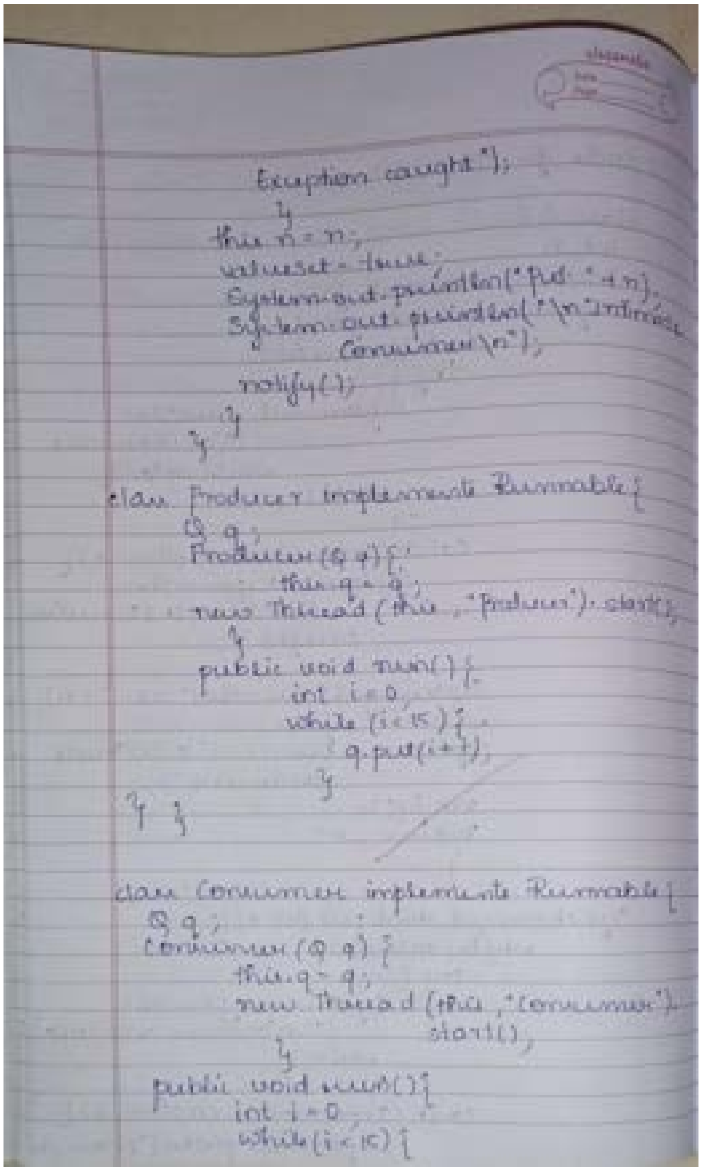
IPC and Deadlock

Algorithm:

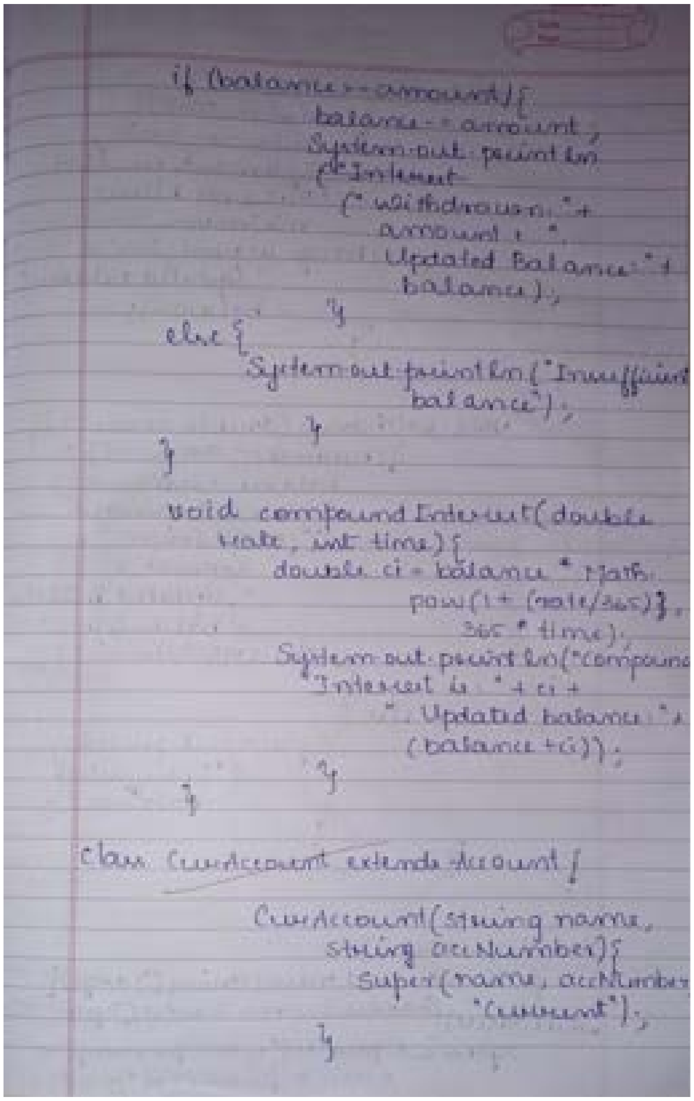


> Code for IPC:

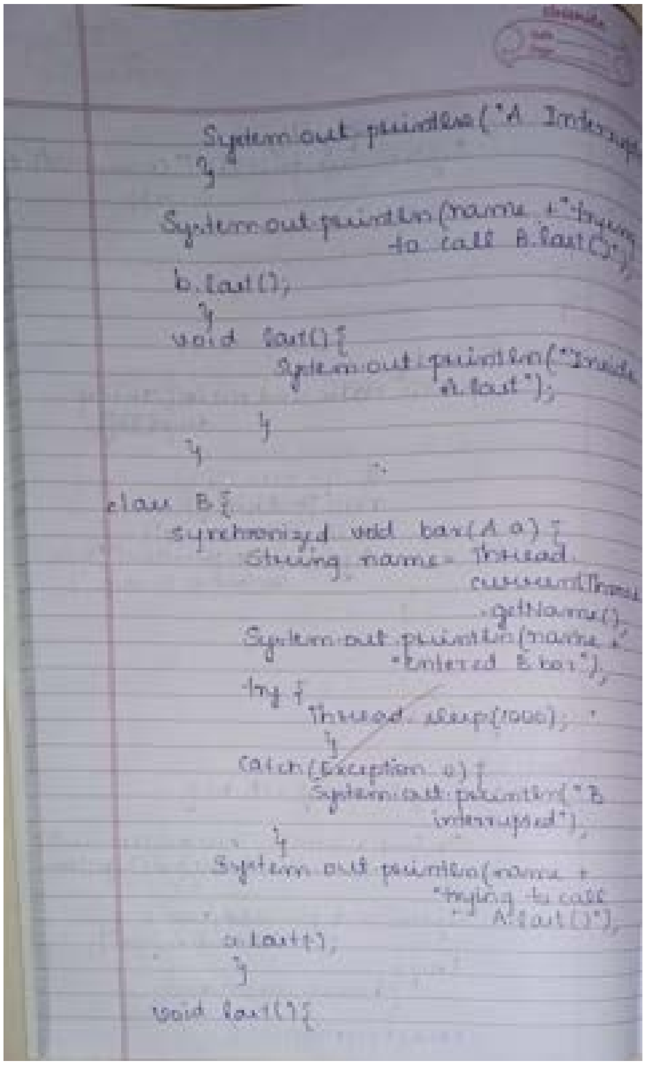
```
class Q {
 int n;
 boolean valueset=false;
 synchronized int get(){
          while(!valueSet)
            try {
               System.out.println
                        ("\n Consumer
                      waiting\n");
                 wait();
               }
            catch(InterruptedException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



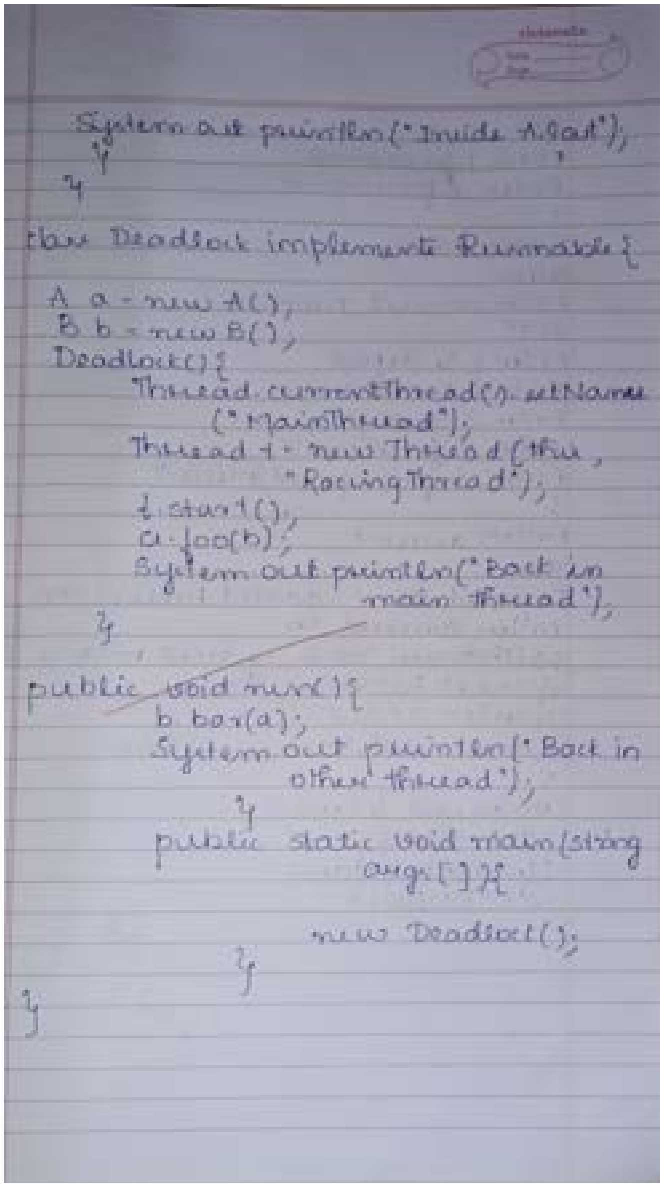
```
            Exception caught");
            }
        this.n=n;
        valueset=true;
        System.out.println("Put : "+n);
        System.out.println("\n Intimate
                Consumer\n");
        notify();
        }
    }

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){
```



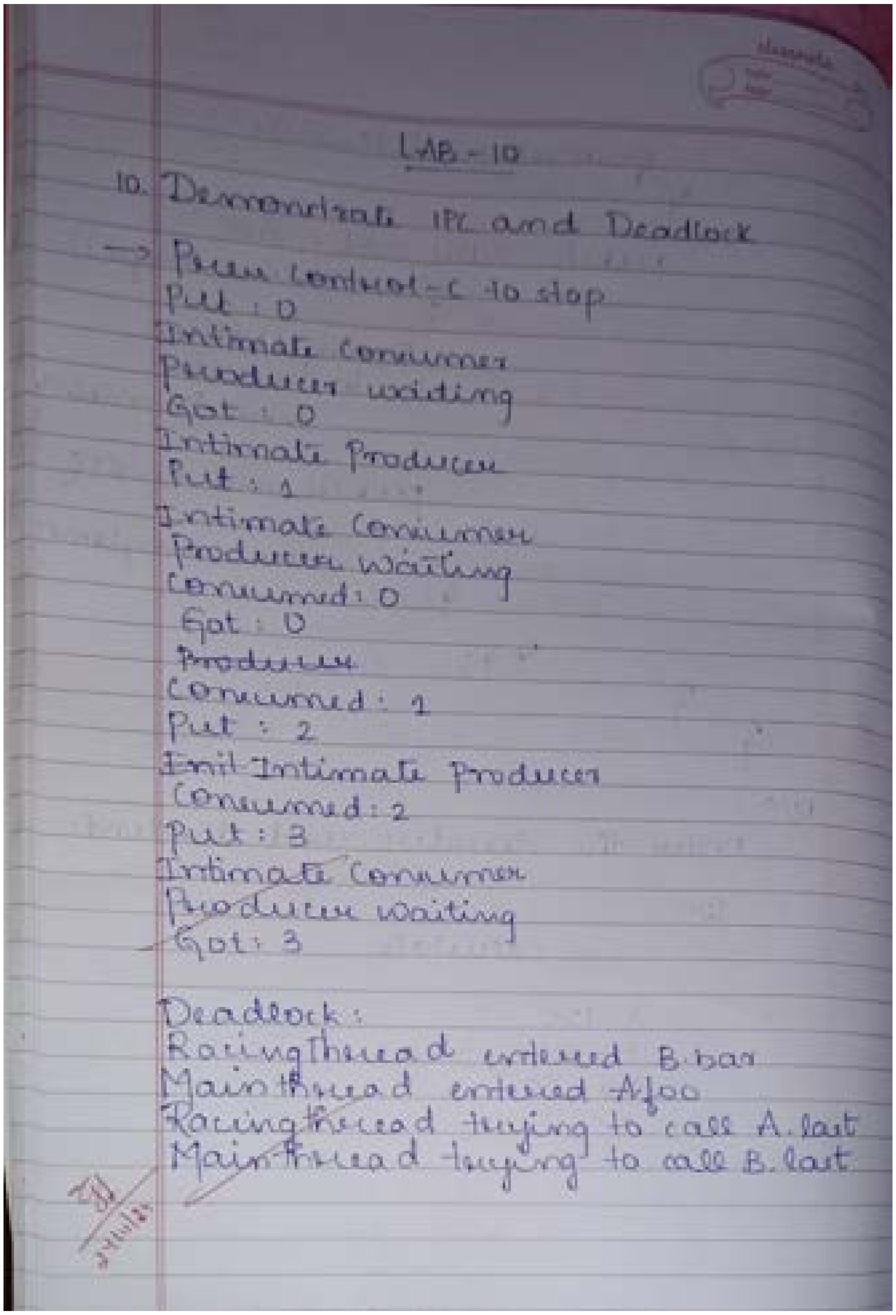
```
if (balance>=amount){
            balance -= amount;
            System.out.println
            ~~("Interest~~
              ("withdrawn: "+
               amount + ",
               Updated Balance: "+
                balance);
          }
else {
      System.out.println("Insufficient
                balance");
          }
}

void compoundInterest(double
    rate, int time){
        double ci = balance * Math.
                    pow(1+ (rate/365)},
                      365 * time);
        System.out.println("Compound
          Interest is "+ci+
               ", Updated balance: "+
                (balance+ci));
         }
 }

class CurrAccount extends Account {

        CurrAccount(String name,
            String accNumber){
              super(name, accNumber
                    "Current");
         }
```



```
        System.out.println("A Interrup
        }

      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);
        System.out.println("Back in
                         main thread");
    }

public void run(){
         b.bar(a);
         System.out.println("Back in
                   other thread");
           }
         public static void main(String
                         args[]){

                   new Deadlock();
           }
}
```



LAB - 10

10. Demonstrate IPC and Deadlock

→ Press control-C to stop
Put : 0
Intimate Consumer
Producer waiting
Got : 0
Intimate Producer
Put : 1
Intimate Consumer
Producer waiting
Consumed : 0
Got : 0
~~Producer~~
Consumed : 1
Put : 2
~~Emit~~ Intimate Producer
Consumed : 2
Put : 3
Intimate Consumer
Producer waiting
Got : 3

Deadlock :
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last
MainThread trying to call B.last

<u>Code:</u>

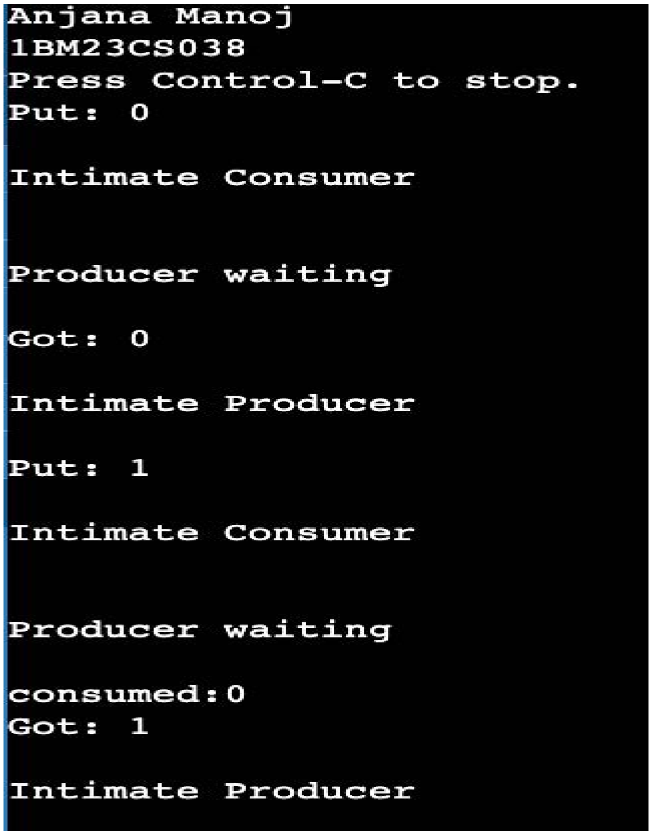
IPC:

```
class Q {

int n;

boolean valueSet = false;

synchronized int get() {
```

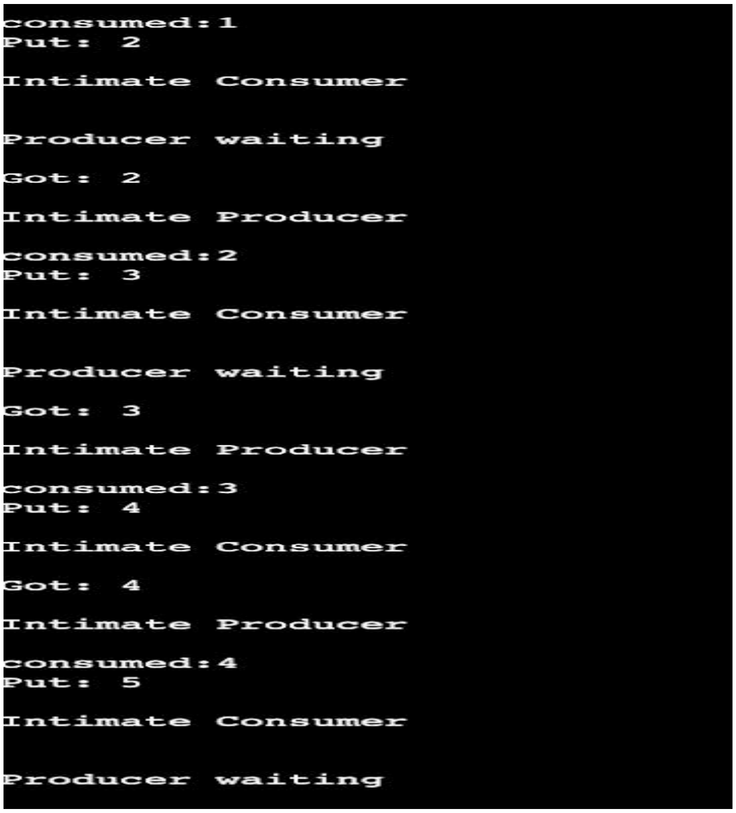
```
while(!valueSet)

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;
```

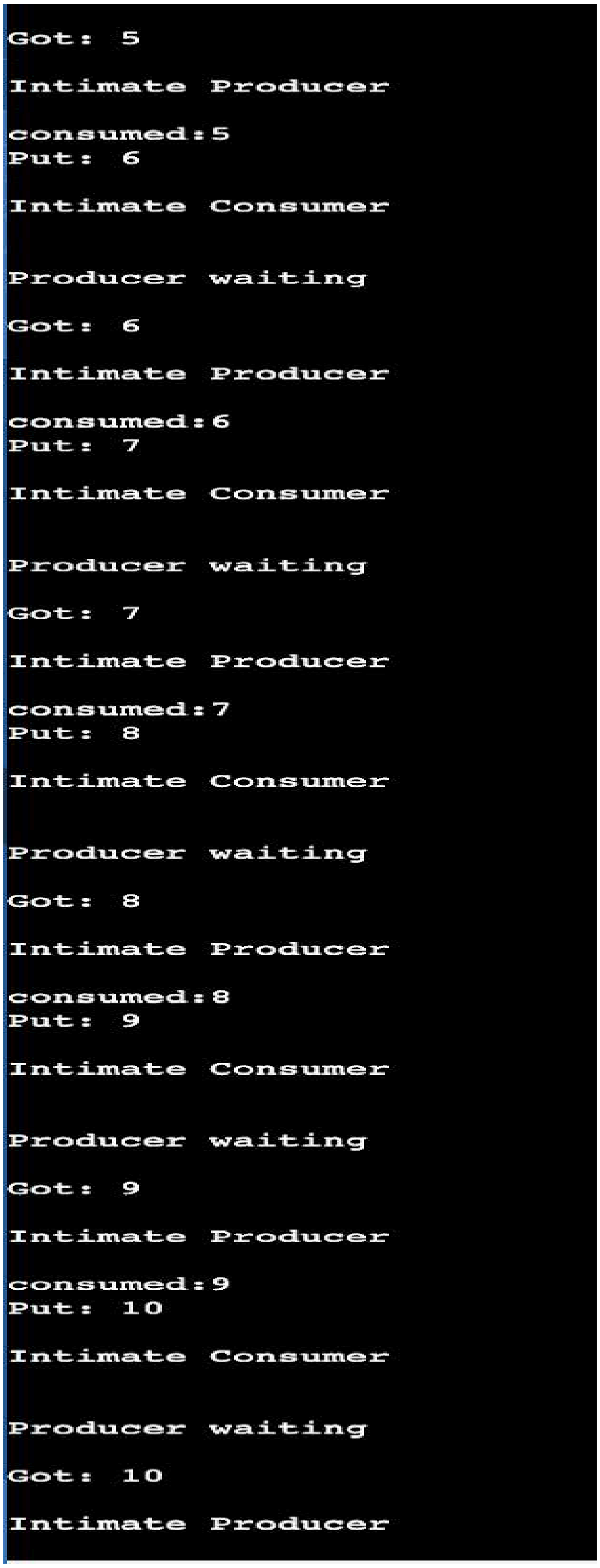
```
new Thread(this, "Consumer").start();

}

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[]) {

Q q = new Q();

new Producer(q);

new Consumer(q);

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

}

}
```



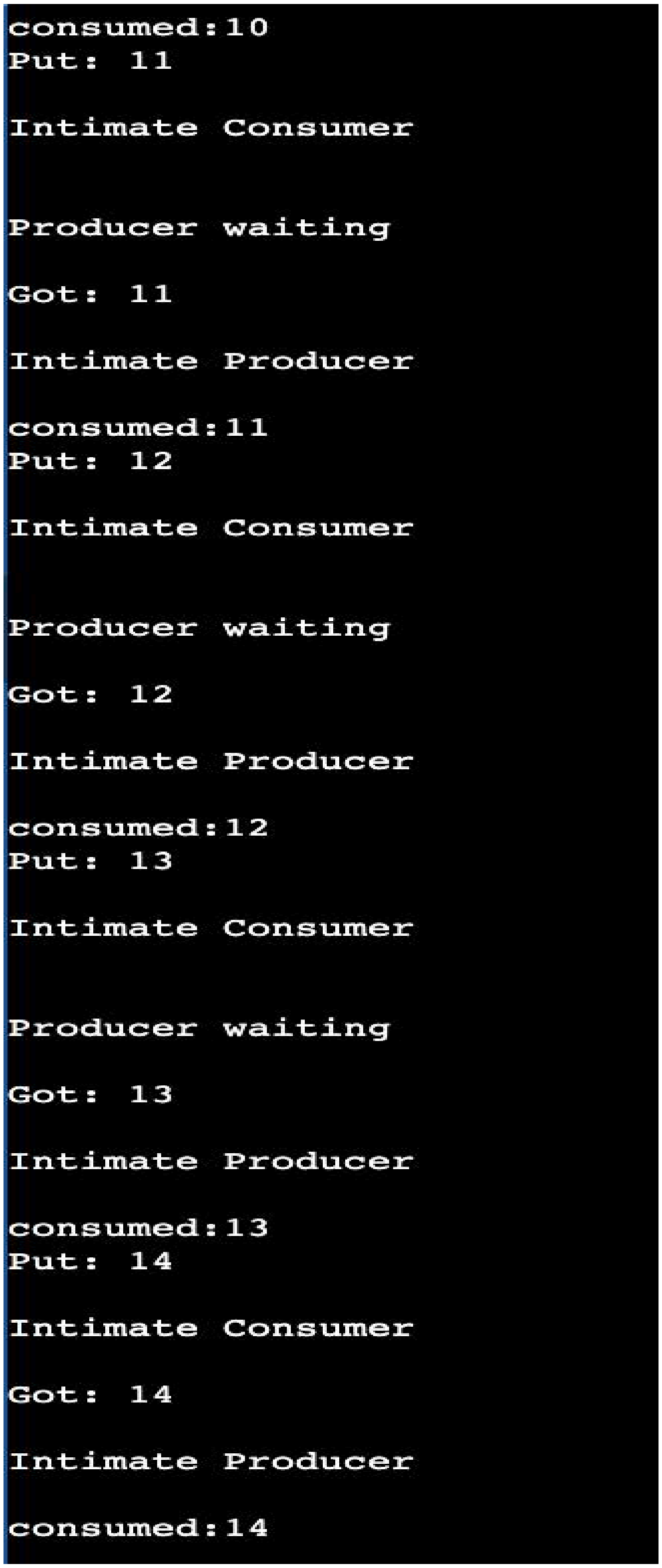
```
Anjana Manoj
1BM23CS038
Press Control-C to stop.
Put: 0

Intimate Consumer


Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

consumed:0
Got: 1

Intimate Producer
```



```
consumed:1
Put: 2

Intimate Consumer


Producer waiting

Got: 2

Intimate Producer

consumed:2
Put: 3

Intimate Consumer


Producer waiting

Got: 3

Intimate Producer

consumed:3
Put: 4

Intimate Consumer

Got: 4

Intimate Producer

consumed:4
Put: 5

Intimate Consumer


Producer waiting
```



```
Got:  5

Intimate Producer

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
```



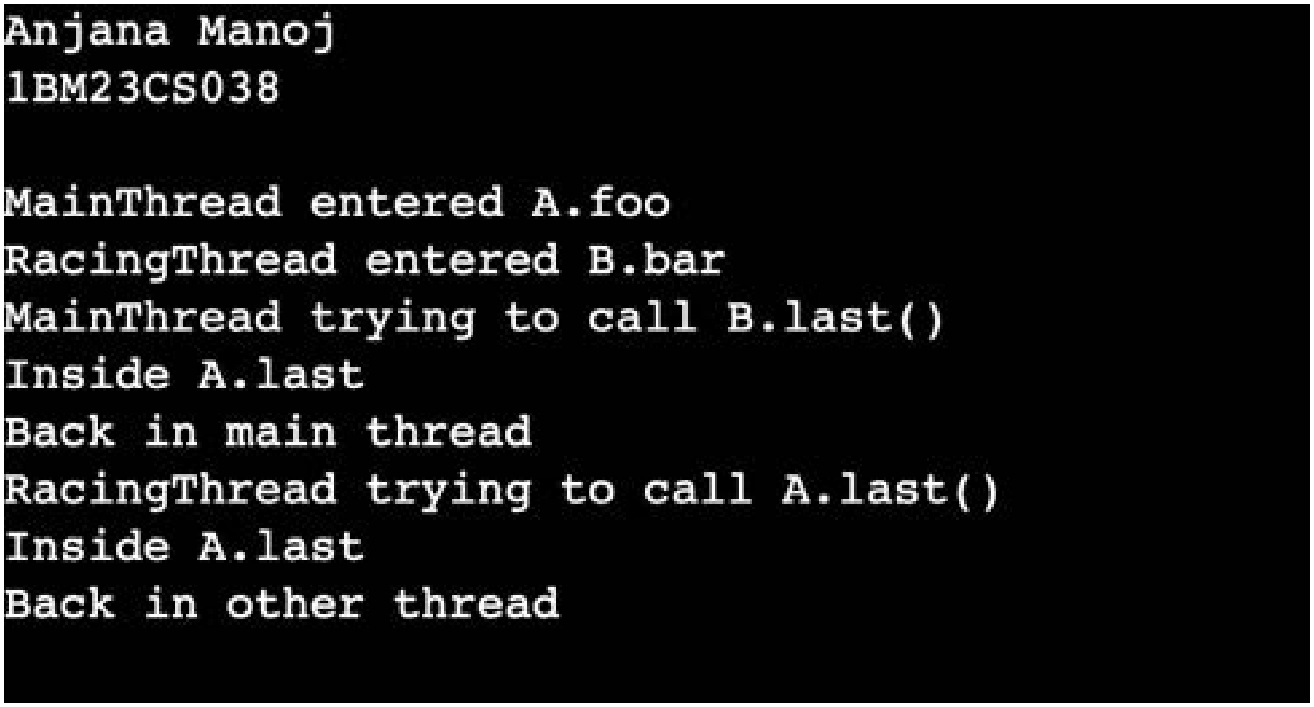
```
consumed:10
Put:  11

Intimate Consumer


Producer waiting

Got:  11

Intimate Producer

consumed:11
Put:  12

Intimate Consumer


Producer waiting

Got:  12

Intimate Producer

consumed:12
Put:  13

Intimate Consumer


Producer waiting

Got:  13

Intimate Producer

consumed:13
Put:  14

Intimate Consumer

Got:  14

Intimate Producer

consumed:14
```

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("M
ainThread");

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

t.start();

a.foo(b); // get lock on a in this
thread.

System.out.println("Back in main
thread");

}
public void run() {

b.bar(a); // get lock on b in other
thread.

System.out.println("Back in other
thread");

}
```

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

new Deadlock();

}

}
```



```
Anjana Manoj
1BM23CS038

MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
Inside A.last
Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread
```