VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Akul J Shingetagere (1BM23CS023)

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 **Akul J Shingetagere (1BM23CS023),** 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.

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

Index

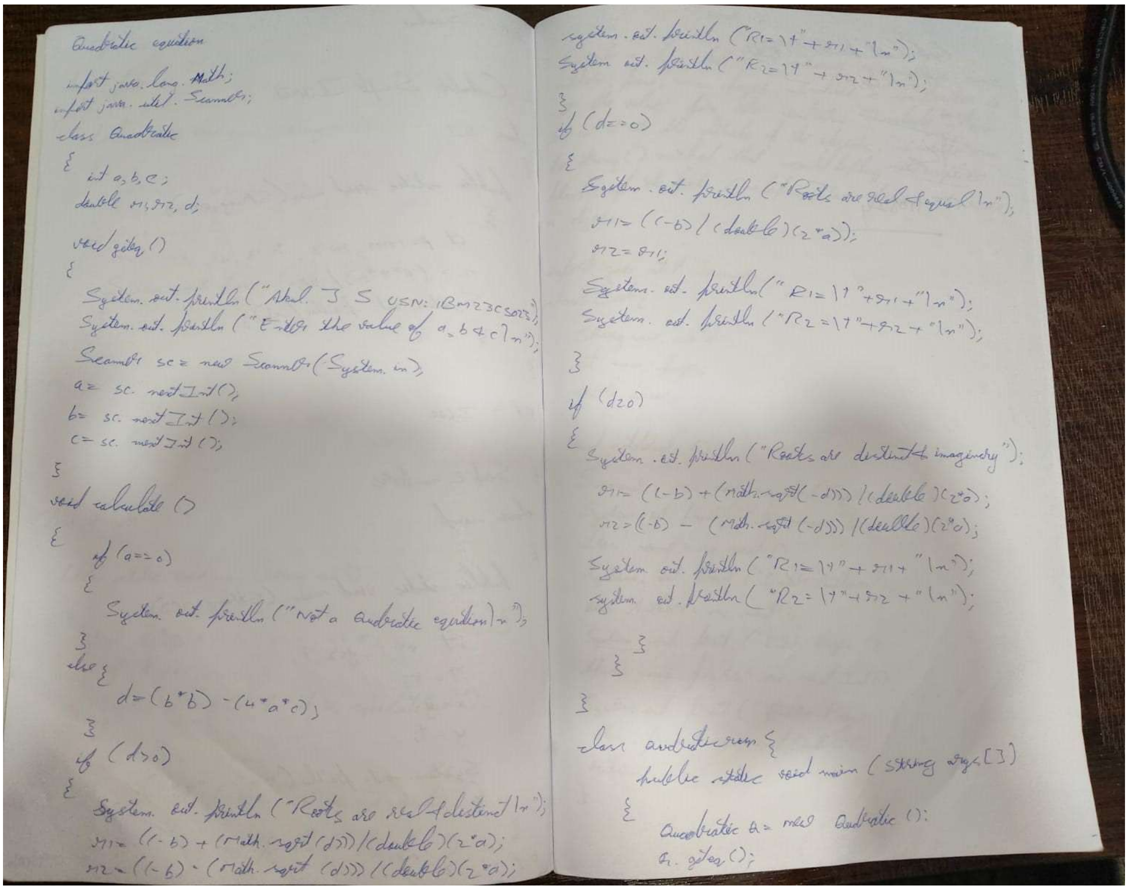
Sl. No.	Date	Experiment Title	Page No.
1	9/10/24	Quadratic equation	4
2	16/10/24	SGPA calculator	9
3	23/10/24	Book program	16
4	23/10/24	Abstract class shape program	20
5	13/11/24	Bank program	24
6	13/11/24	Packages	32
7	20/11/24	Exception handling	39
8	27/11/24	Multithreading	44
9	27/11/24	Integer division with user interface	47
10	27/11/24	Inter process communication and deadlock	50

Github Link: https://github.com/AkulJ023/akul-java-lab

Program 1

Implement Quadratic Equation

Algorithm



Quadratic equation

```
import java.lang.Math;
import java.util.Scanner;
class Quadratic
{
  int a,b,c;
 double r1,r2,d;

 void getEq()
 {
  System.out.println("Akul J S USN: 1BM23CS023");
  System.out.println("Enter the value of a,b&c\n");
  Scanner sc= new Scanner(System.in);
  a= sc.nextInt();
  b= sc.nextInt();
  c= sc.nextInt();
}
void calculate()
{
   if(a==0)
   {
     System.out.println("Not a Quadratic equation\n");
   }
  else{
     d=(b*b)-(4*a*c);
   }
  if(d>0)
 {
   System.out.println("Roots are real&distinct\n");
   r1= ((-b)+(Math.sqrt(d)))/(double)(2*a);
   r2=((-b)-(Math.sqrt(d)))/(double)(2*a);
system.out.println("R1=\t"+r1+"\n");
System.out.println("R2=\t"+r2+"\n");

}
if(d==0)
{
 System.out.println("Roots are real&equal\n");
  r1=((-b)/(double)(2*a));
  r2=r1;
 System.out.println("R1=\t"+r1+"\n");
 System.out.println("R2=\t"+r2+"\n");

}
if(d<0)
{
 System.out.println("Roots are distinct&imaginary");
  r1=((-b)+(Math.sqrt(-d)))/(double)(2*a);
  r2=((-b)-(Math.sqrt(-d)))/(double)(2*a);
 System.out.println("R1=\t"+r1+"\n");
 system.out.println("R2=\t"+r2+"\n");

    }
  }

}
class Quadraticmain{
    public static void main(String args[])
   {
     Quadratic Q= new Quadratic();
     Q.getEq();
```

~~System.out.println("Enter~~

```
a.calculate();
}
```

07/10/24

3. C

auth

set

to

to

declar

n o

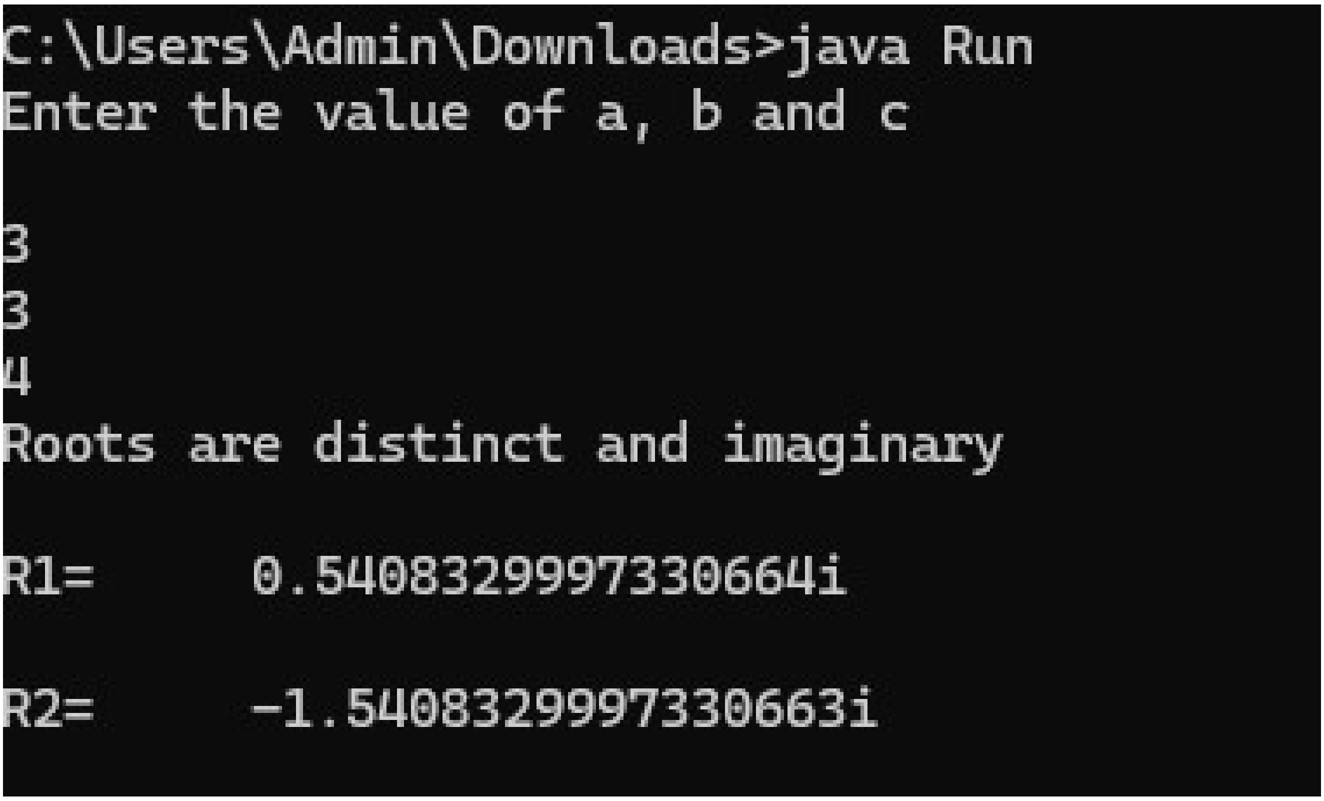
impo

cla

Code:

```
import java.lang.Math;
import java.util.Scanner;
class Quadratic
{
   int a,b,c;
   double r1,r2,d;

   void calculate()
   {
      if(a==0)
      {
         System.out.println("Not a quadratic equation \n");
      }
      else{
         d=(b*b)-(4*a*c);
      }
      if(d>0)
      {
         System.out.println("Roots are real and distinct \n");
         r1=((-b) + (Math.sqrt(d)))/(double)(2*a);
         r2=((-b) - (Math.sqrt(d)))/(double)(2*a);
         System.out.println("R1= \t" + r1 +"\n");
         System.out.println("R2= \t" + r2 +"\n");
      }
      if(d==0)
      {
         System.out.println("Roots are real and equal \n");
         r1=((-b)/(double)(2*a));
         r2=r1;
         System.out.println("R1= \t" + r1 +"\n");
         System.out.println("R2= \t" + r2 +"\n");
      }
      if(d<0)
      {
         System.out.println("Roots are distinct and imaginary \n");
         r1=((-b) + (Math.sqrt(-d)))/(double)(2*a);
         r2=((-b) - (Math.sqrt(-d)))/(double)(2*a);
         System.out.println("R1= \t" + r1 +"i\n");
         System.out.println("R2= \t" + r2 +"i\n");
      }

   }


}


class Run{
   public static void main(String args[]){
        Quadratic Q= new Quadratic();
        System.out.println("Enter the value of a, b and c \n");
        Scanner sc = new Scanner(System.in);
        Q.a=sc.nextInt();
        Q.b=sc.nextInt();
        Q.c=sc.nextInt();
        Q.calculate();
    }
}
```

Output:

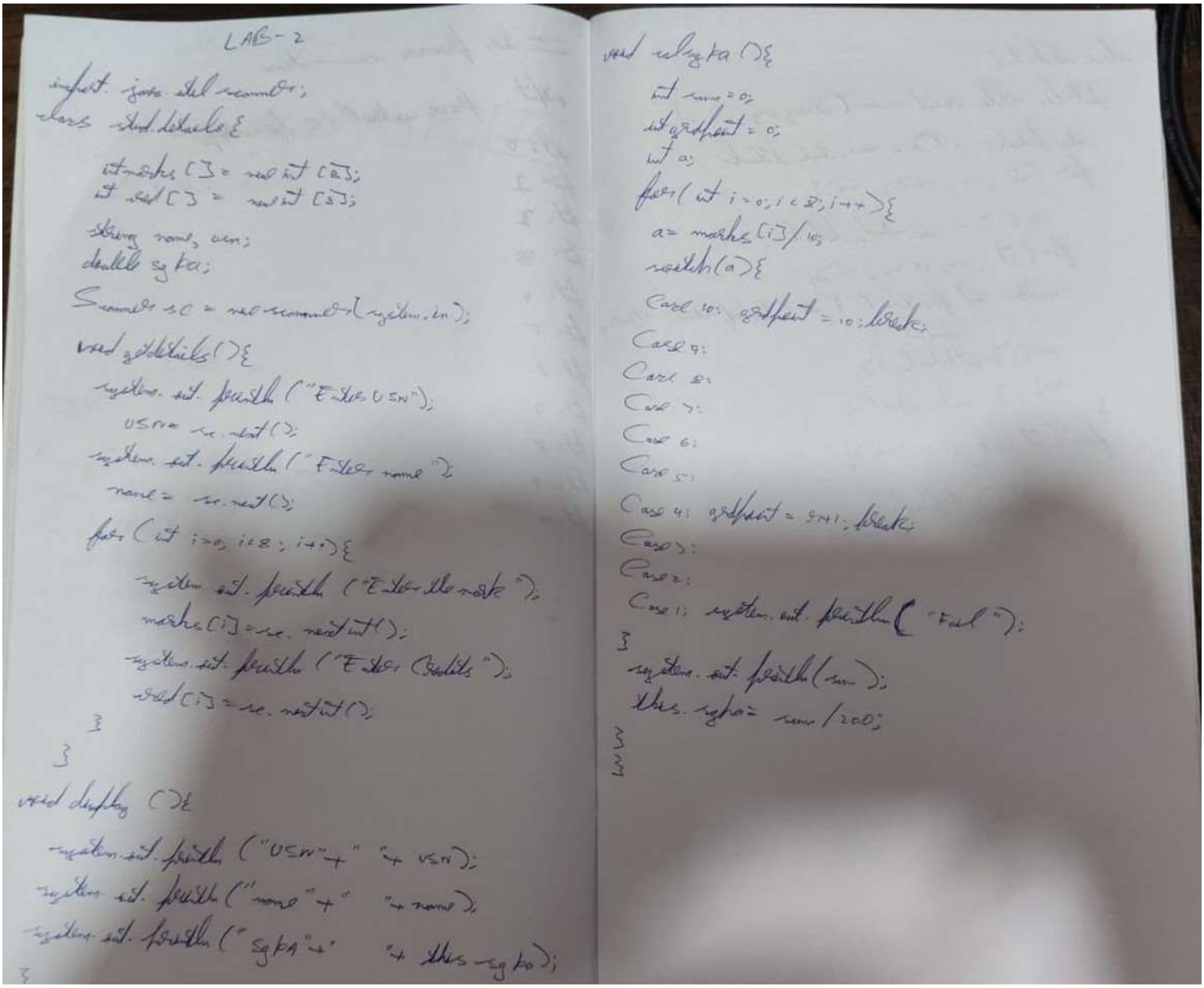


```
C:\Users\Admin\Downloads>java Run
Enter the value of a, b and c

3
3
4
Roots are distinct and imaginary

R1=     0.5408329997330664i

R2=     -1.5408329997330663i
```

Program 2

SGPA Calculator

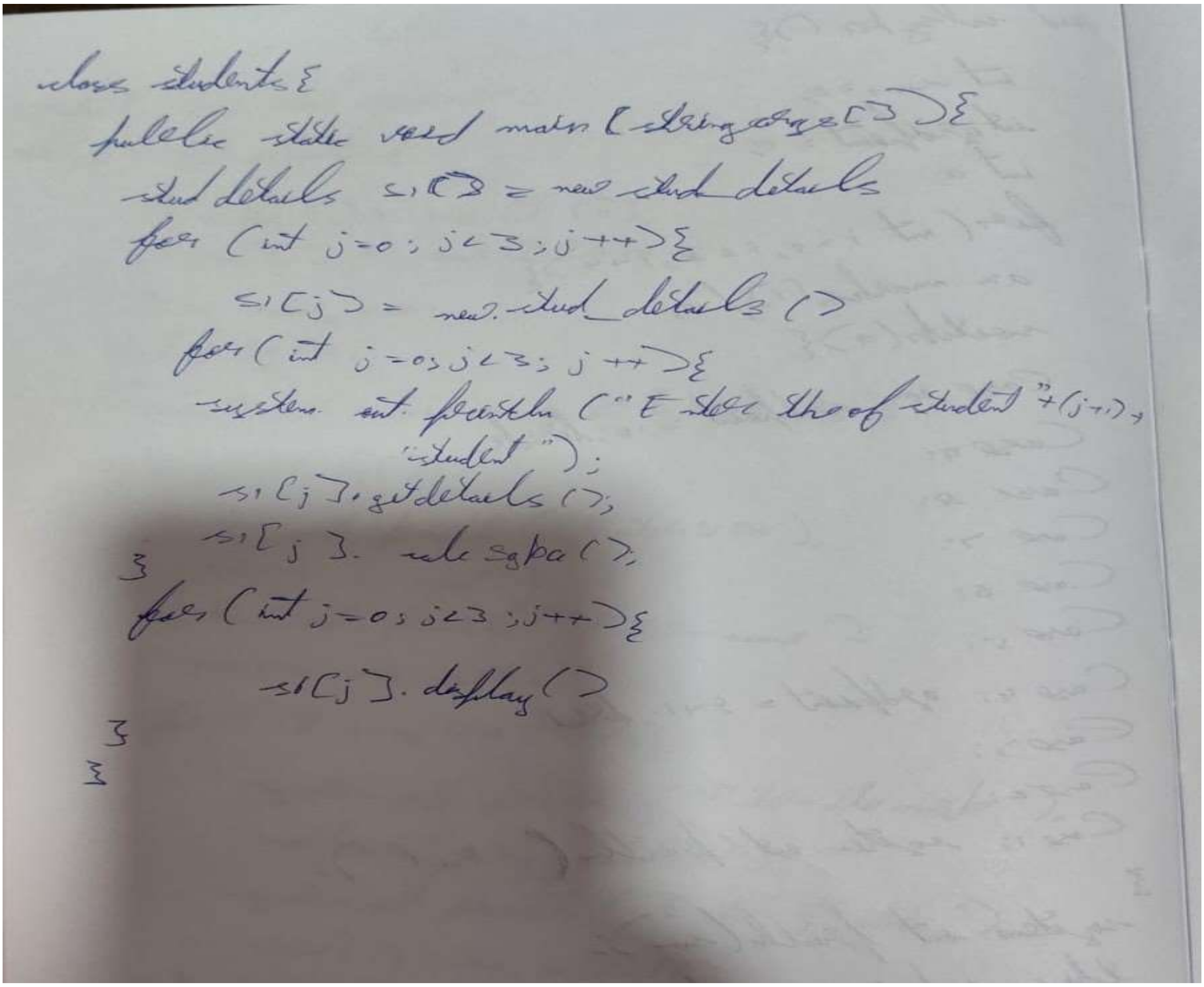
Algorithm:



LAB-2

```
import. java util scanner;
class stud details{

  int marks [] = new int [8];
  int cred [] = new int [8];
  String name, usn;
  double sgpa;
  Scanner sc = new scanner(system.in);
  void getdetails(){
   system. out. println ("Enter USN");
     USN= sc.next();
   system. out. println ("Enter name");
     name = sc.next();
   for (int i=0; i<8 ; i++){
      system. out. println ("Enter the mark");
      marks[i]=sc. nextint();
     system. out. println ("Enter Credits");
      cred[i] = sc. nextint();
   }
  }
void display (){
  system.out. println ("USN"+"  "+ usn);
 system. out. println ("name"+"   "+ name);
system. out. println ("sgpa"+"    "+ this-sgpa);
}
```

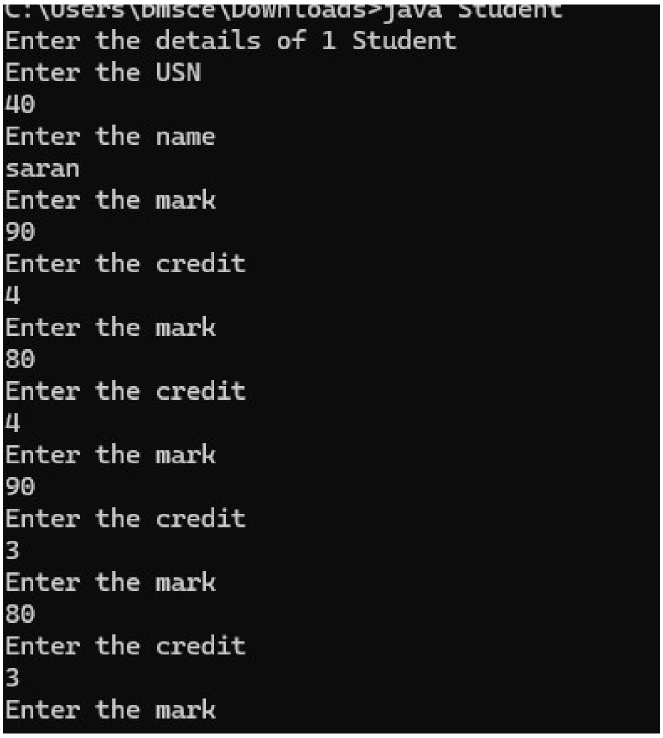
```
void calsgpa (){
   int sum = 0;
   int grdpoint = 0;
   int a;
  for( int i=0; i<8; i++){
   a= marks [i]/10;
   switch(a){
   Case 10: grdpoint = 10; break;
  Case 9:
  Case 8:
  Case 7:
  Case 6:
  Case 5:
  Case 4: grdpoint = a+1; break;
  Case 3:
  Case 2:
  Case 1: system. out. println( "Fail");
 }
 system. out. println(sum);
 this. sgpa= sum/200;
}
}
}
```



```
class students{
  public static void main(String args[]){
   stud details s1[] = new stud details
   for (int j=0; j<3; j++){
        s1[j] = new stud_details()
     for(int j=0; j<3; j++){
     system.out.println("Enter the of student"+(j+1)+
                    "student");
       s1[j].getdetails();
       s1[j]. calcsgpa();
  }
  for(int j=0; j<3; j++){
          s1[j]. display()
  }
}
```

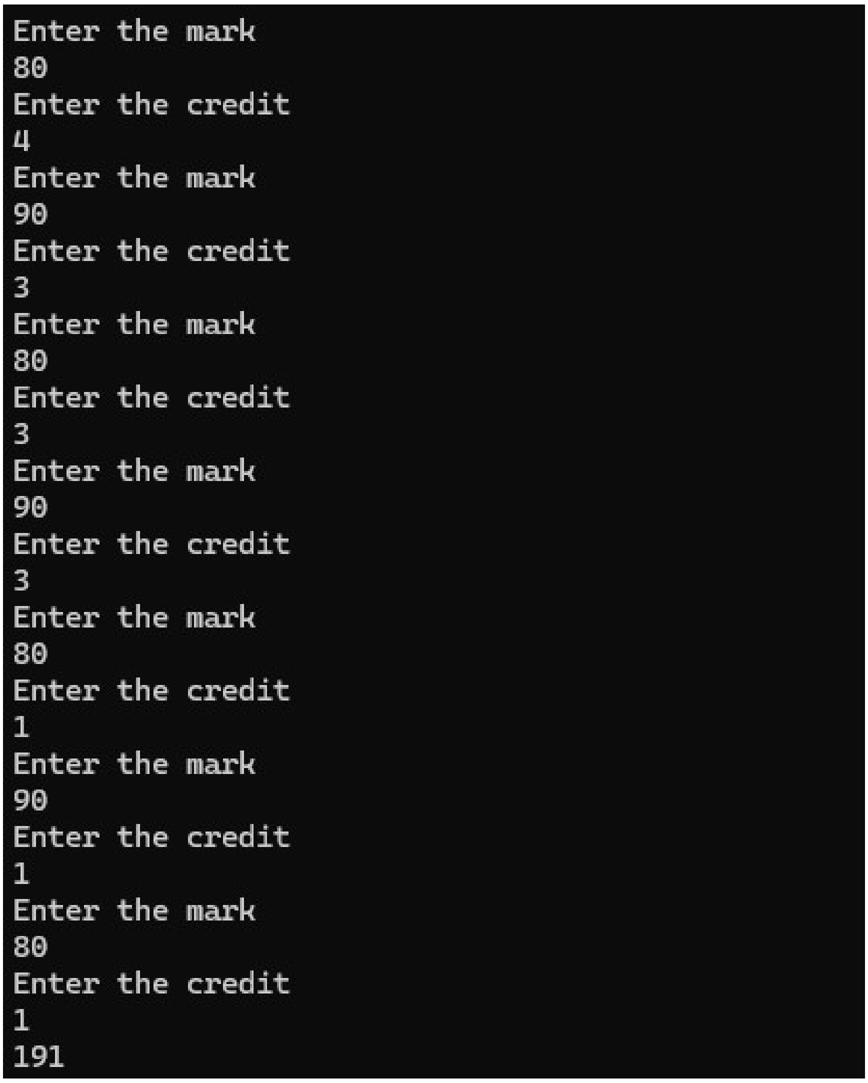
Code:

```
import java.util.Scanner;
class Stud_details{

    int marks[]=new int[8];
    int cred[]= new int[8];
    String name,usn;
    double sgpa;
    Scanner sc =new Scanner(System.in);


    void getdetails(){
        System.out.println(" Enter the USN ");
        usn=sc.next();
        System.out.println(" Enter the name ");
        name=sc.next();
        for(int i=0;i<8;i++){
            System.out.println(" Enter the mark ");
            marks[i]=sc.nextInt();
            System.out.println(" Enter the credit ");
            cred[i]=sc.nextInt();
}
}

void display(){



    System.out.println("usn"+" "+usn);
    System.out.println("name"+" "+name);


    System.out.println("SGPA is"+this.sgpa);


}

void calcSGPA(){
    int sum=0;
    int grdpoint=0;

    int x;
    for(int i=0;i<8;i++){
        x=marks[i]/10;
        switch(x){

            case 10: grdpoint=10;break;
            case 9:
            case 8:
            case 7:
            case 6:
            case 5:
            case 4: grdpoint=x+1;break;
            case 3:
            case 2:
            case 1:System.out.println("Fail");break;
            }
        sum=sum+ (grdpoint*cred[i]);
}
    System.out.println(sum);
    this.sgpa=sum/20.0;

}
}


class Student{
    public static void main(String args[]){
        Stud_details s1[]=new Stud_details[3];
        for(int j=0;j<3;j++){
            s1[j]=new Stud_details();       }
        for(int j=0;j<3;j++){
            System.out.println("Enter the details of "+(j+1)+" Student");
            s1[j].getdetails();
            s1[j].calcSGPA();
}
for(int j=0;j<3;j++){
    s1[j].display();
}

}

}
```

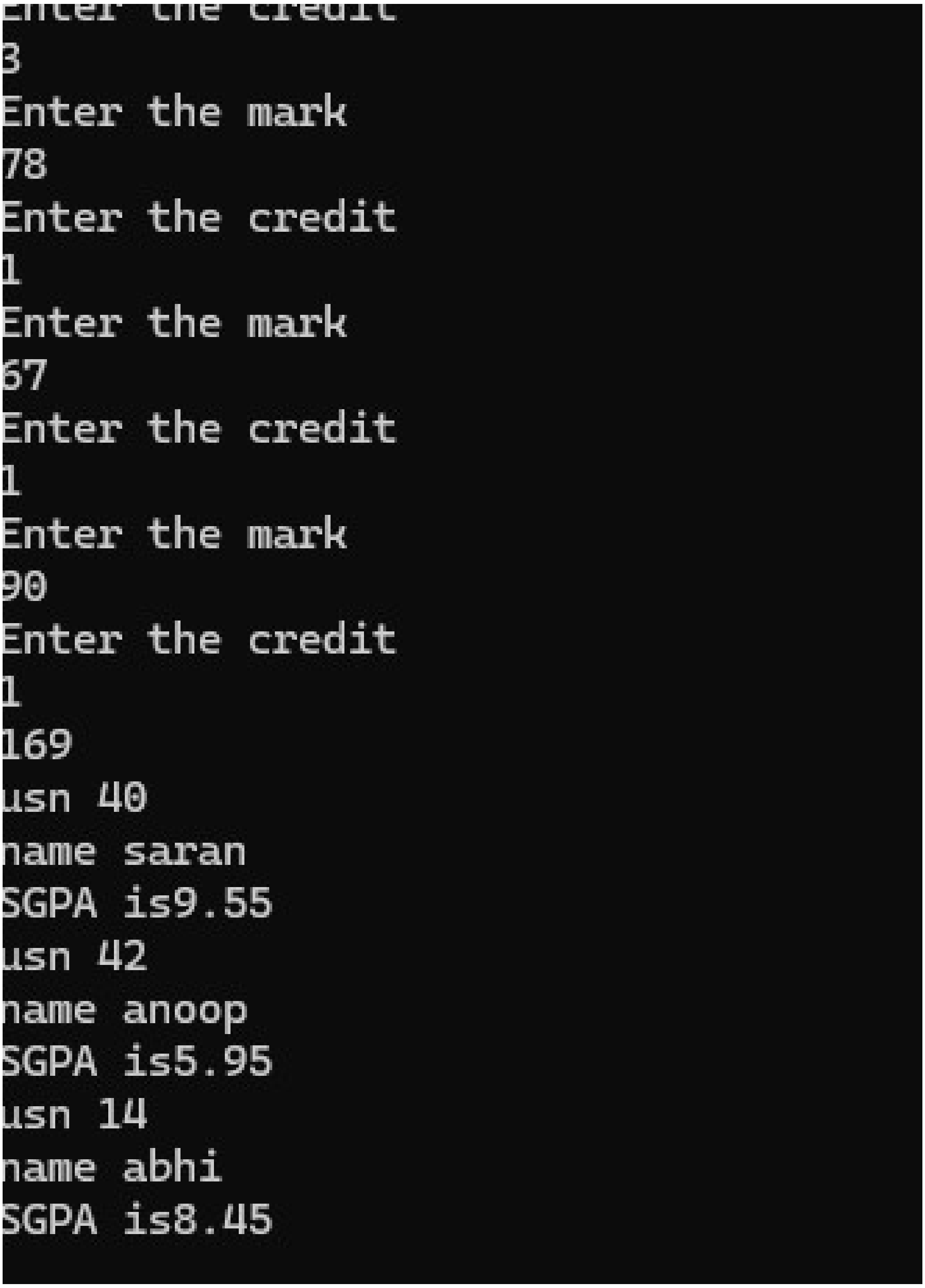


```
C:\Users\Bmsce\Downloads>java Student
Enter the details of 1 Student
Enter the USN
40
Enter the name
saran
Enter the mark
90
Enter the credit
4
Enter the mark
80
Enter the credit
4
Enter the mark
90
Enter the credit
3
Enter the mark
80
Enter the credit
3
Enter the mark
```

Gfy7ghu8hj90oklo



```
Enter the mark
80
Enter the credit
4
Enter the mark
90
Enter the credit
3
Enter the mark
80
Enter the credit
3
Enter the mark
90
Enter the credit
3
Enter the mark
80
Enter the credit
1
Enter the mark
90
Enter the credit
1
Enter the mark
80
Enter the credit
1
191
```

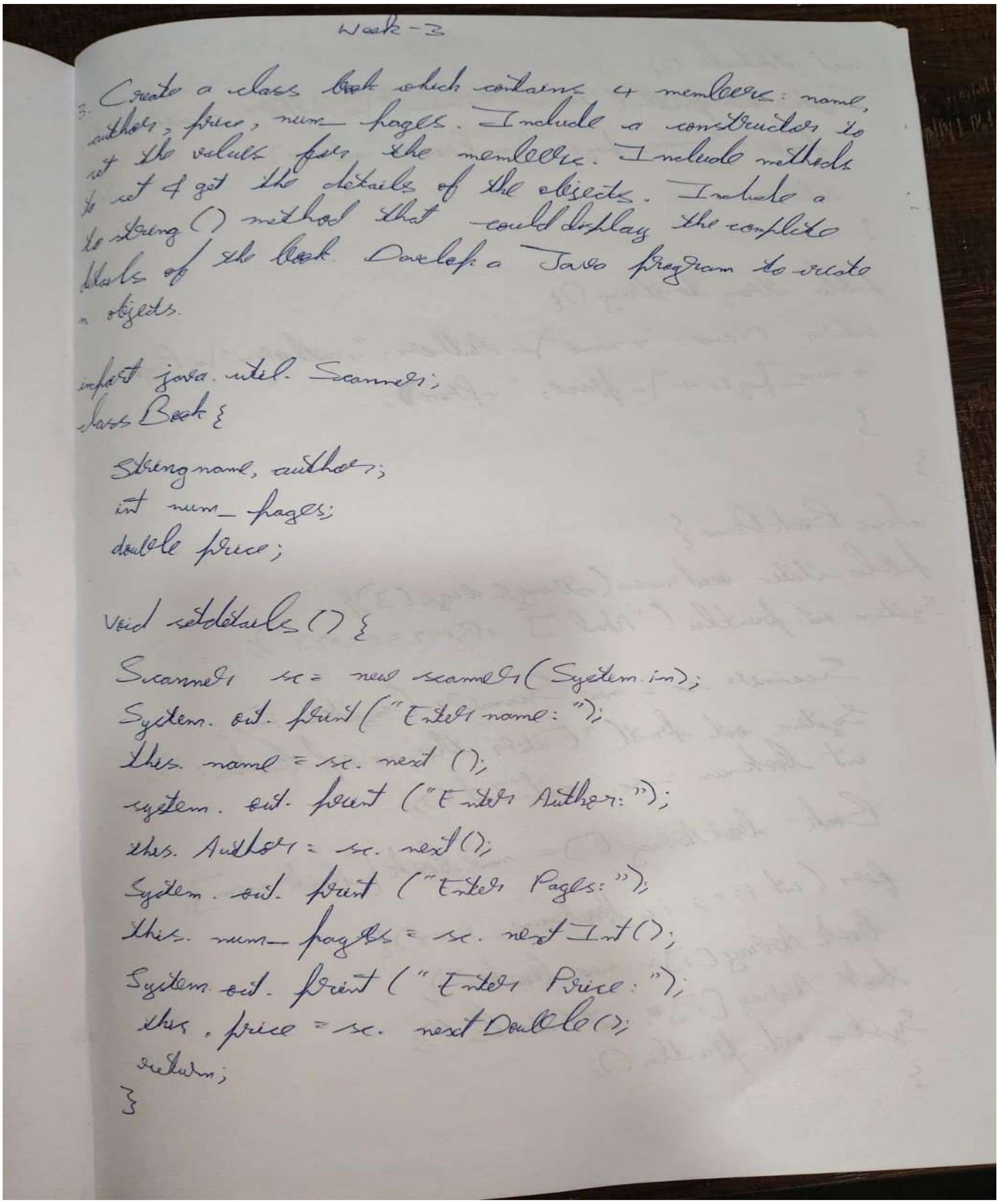


```
Enter the credit
3
Enter the mark
78
Enter the credit
1
Enter the mark
67
Enter the credit
1
Enter the mark
90
Enter the credit
1
169
usn 40
name saran
SGPA is9.55
usn 42
name anoop
SGPA is5.95
usn 14
name abhi
SGPA is8.45
```

Program 3

Book Program

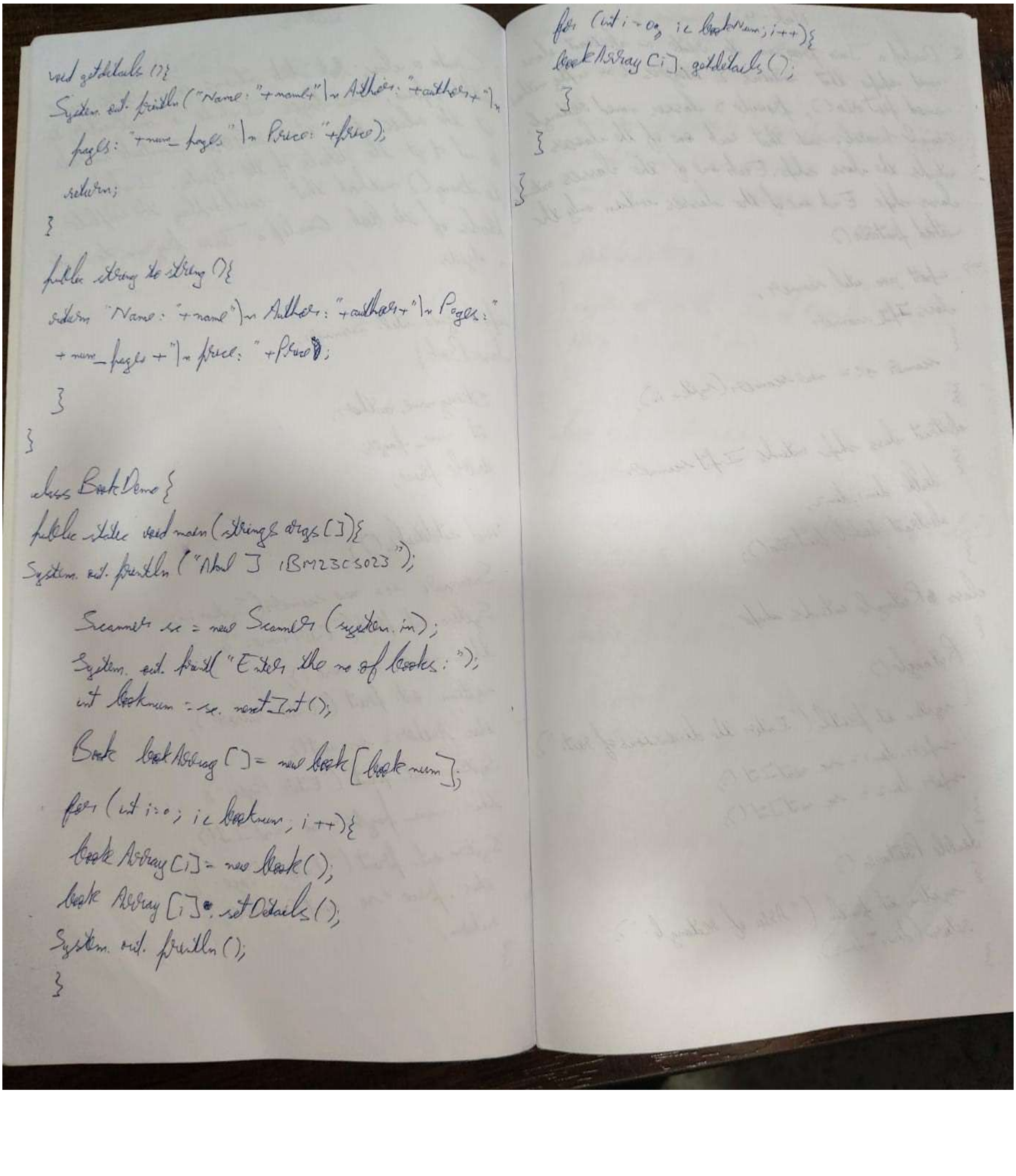
Algorithm:



Week-3

3. Create a class book which contains 4 members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set & 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 objects.

```
import java.util.Scanner;
class Book {

  String name, author;
  int num_pages;
  double price;

  void setdetails () {
   Scanner sc = new scanner(System.in);
   System.out.print("Enter name: ");
   this.name = sc.next ();
   system.out.print ("Enter Auther:");
   this.Author = sc.next();
   System.out.print ("Enter Pages:");
   this.num_pages = sc.nextInt();
   System.out.print ("Enter Price:");
   this.price = sc.nextDouble();
   return;
  }
```

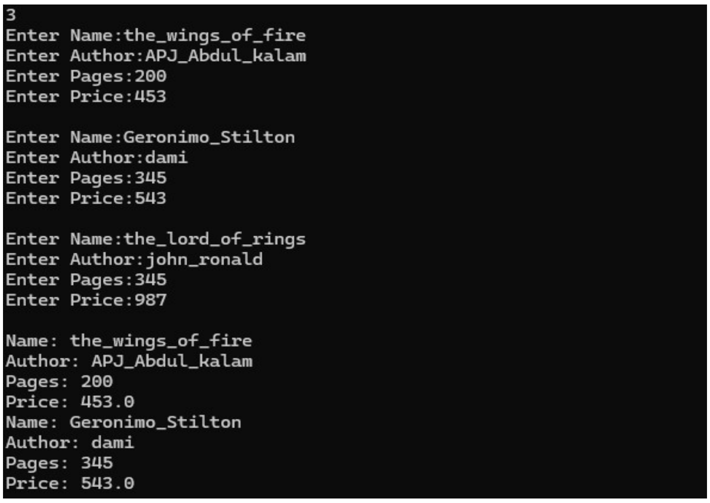


```
void getdetails (){
System. out. println ("Name: "+name+"\n Author: "+author+"\n
  pages: "+num_pages "\n Price: "+price);
  return;
}

public string to string (){
return "Name: "+name"\n Author: "+author+"\n Pages:"
 +num_pages +"\n price: "+price);

 }
}
class BookDemo {
public static void main (strings args[]){
System. out. println ("Nhal J 1BM23CS023");

    Scanner sc = new Scanner (system. in);
    System. out. print( "Enter the no of books: ");
    int booknum = sc. nextInt ();
    Book bookArray []= new book[book num];
  for (int i=0; i< booknum, i++){
  book Array [i]= new book();
  book Array [i]. setDetails ();
 System. out. println ();
 }
for (int i=0; i<bookNum; i++){
bookArray [i]. getdetails ();
 }
}
}
```

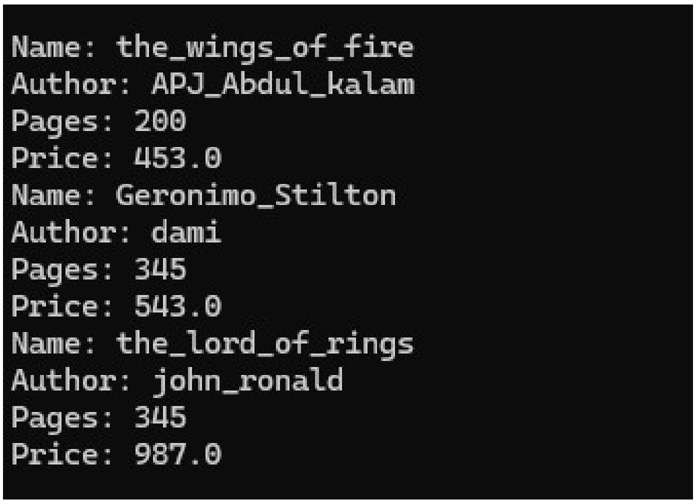
Code:

```
import java.util.Scanner;
class Book{
    String name, author;
    int num_pages;
    double price;

    void setDetails(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter Name:");
        this.name = sc.next();
        System.out.print("Enter Author:");
        this.author = sc.next();
        System.out.print("Enter Pages:");
        this.num_pages = sc.nextInt();
        System.out.print("Enter Price:");
        this.price = sc.nextDouble();
        return;
    }

    void getDetails(){
        System.out.println("Name: "+name+"\nAuthor: "+author+"\nPages: "+num_pages+"\nPrice: "+price);
        return;
    }

    public String toString(){
        return "Name: "+name+"\nAuthor: "+author+"\nPages: "+num_pages+"\nPrice: "+price;
    }
}

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

        Scanner sc = new Scanner(System.in);
        System.out.print("Akul J Shingetagere
        1bm23cs023);System.out.println("Enter the
        number of books: "); int bookNum = sc.nextInt();
```

```
          Book bookArray[] = new Book[bookNum];

          for(int i = 0; i<bookNum; i++){
          bookArray[i] = new Book();
          bookArray[i].setDetails();
          System.out.println();
          }

          for (int i = 0; i<bookNum; i++){
          bookArray[i].getDetails();
          }
        }
}
```



```
3
Enter Name:the_wings_of_fire
Enter Author:APJ_Abdul_kalam
Enter Pages:200
Enter Price:453

Enter Name:Geronimo_Stilton
Enter Author:dami
Enter Pages:345
Enter Price:543

Enter Name:the_lord_of_rings
Enter Author:john_ronald
Enter Pages:345
Enter Price:987

Name: the_wings_of_fire
Author: APJ_Abdul_kalam
Pages: 200
Price: 453.0
Name: Geronimo_Stilton
Author: dami
Pages: 345
Price: 543.0
```

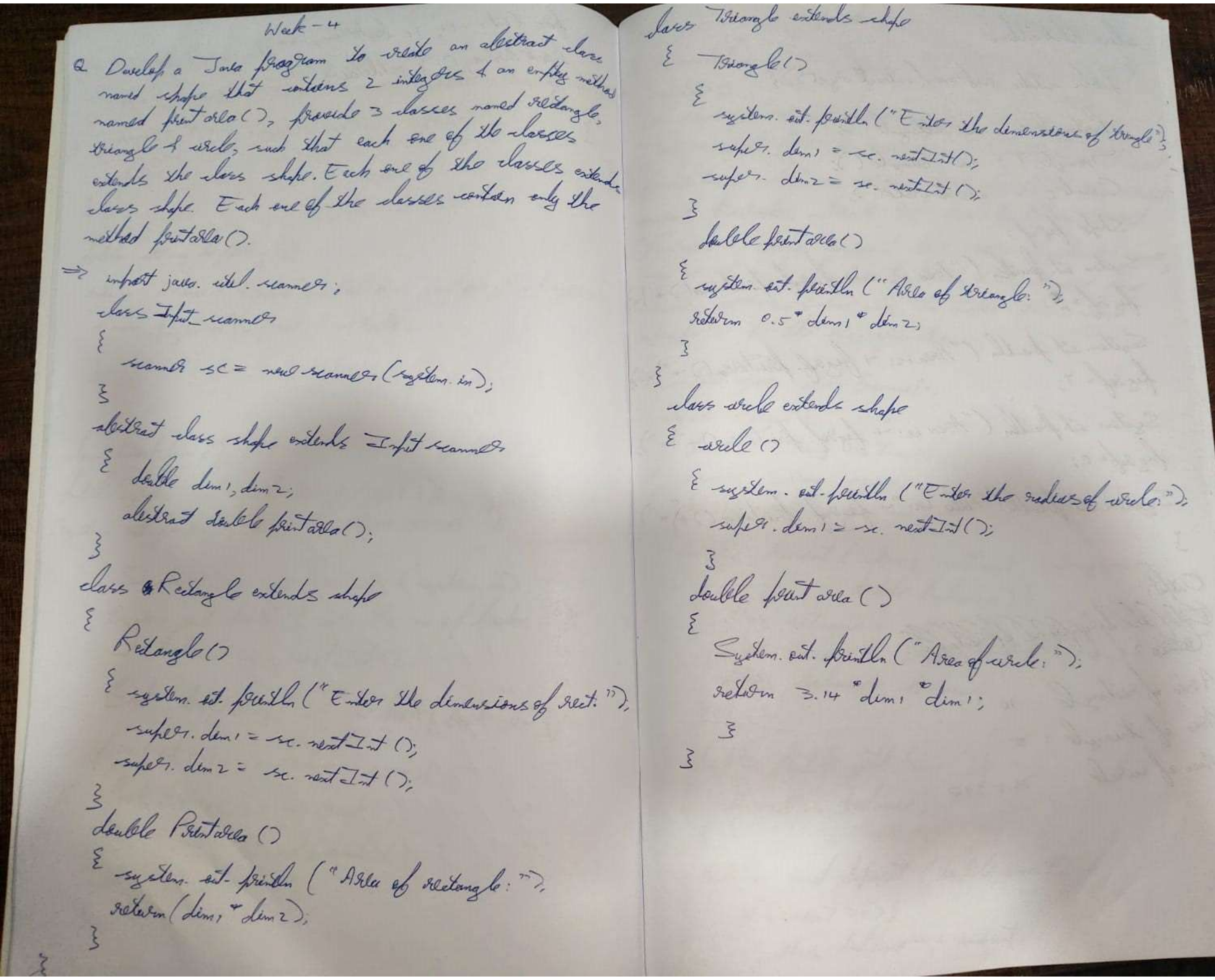


```
Name: the_wings_of_fire
Author: APJ_Abdul_kalam
Pages: 200
Price: 453.0
Name: Geronimo_Stilton
Author: dami
Pages: 345
Price: 543.0
Name: the_lord_of_rings
Author: john_ronald
Pages: 345
Price: 987.0
```

Program 4

Abstract Class shape program

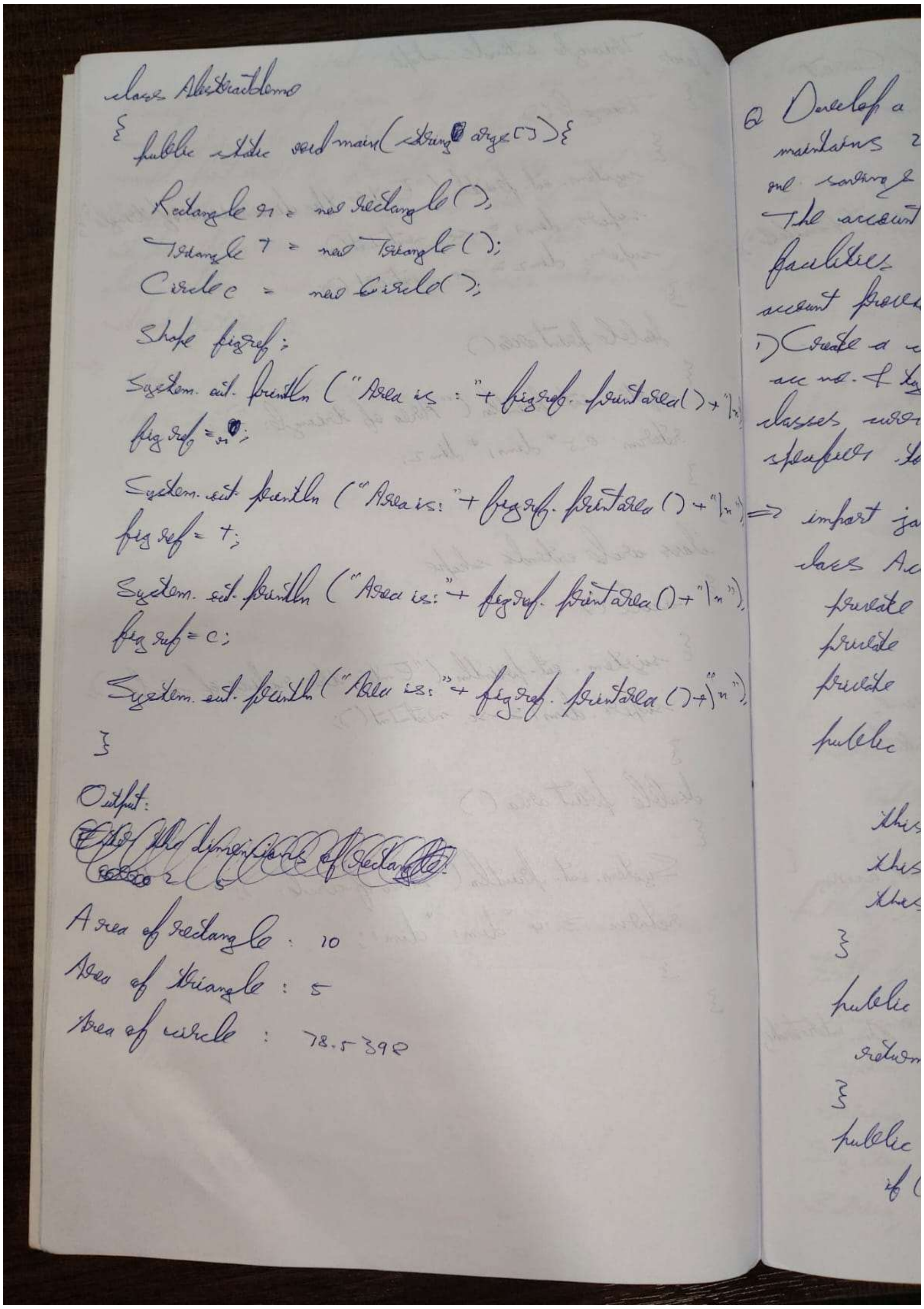
Algorithm



Week - 4

Q Develop a Java program to create an abstract class named shape that contains 2 integers & an empty method named printarea(), provide 3 classes named rectangle, triangle & circle, such that each one of the classes extends the class shape. Each one of the classes extends class shape. Each one of the classes contain only the method printarea().

```
=> import java. util. scanner;
class Input_scanner
{
  scanner sc = new scanner(system. in);
}
abstract class shape extends Input scanner
{ double dim1, dim2;
  abstract double printarea();
}
class Rectangle extends shape
{
  Rectangle()
  { system. out. println ("Enter the dimensions of rect.");
    super. dim1 = sc. nextInt ();
   super. dim2 = sc. nextInt ();
  }
 double Printarea ()
 { system. out. println ("Area of rectangle:");
   return(dim1 * dim2);
 }
}
class Triangle extends shape
 { Triangle()
   {
    system. out. println ("Enter the dimensions of triangle");
    super. dim1 = sc. nextInt();
    super. dim2 = sc. nextInt ();
  }
   double printarea()
  { system. out. println ("Area of triangle:");
   return  0.5 * dim1 * dim2;
  }
}
class circle extends shape
 { circle ()
   { system. out. println ("Enter the radius of circle:");
     super. dim1 = sc. nextInt();
   }
  double print area ()
  {
    System. out. println ("Area of circle:");
    return  3.14 * dim1 * dim1;
    }
  }
```



```
class AbstractDemo
{ public static void main(String args[]){
    Rectangle r1 = new Rectangle();
    Triangle t = new Triangle();
    Circle c = new Circle();
    Shape figref;
   System.out.println("Area is : "+figref.printarea()+"\n");
   figref = r1;
   System.out.println("Area is: "+figref.printarea()+"\n");
   figref = t;
   System.out.println("Area is: "+figref.printarea()+"\n");
   figref = c;
   System.out.println("Area is: "+figref.printarea()+"\n");
 }
```

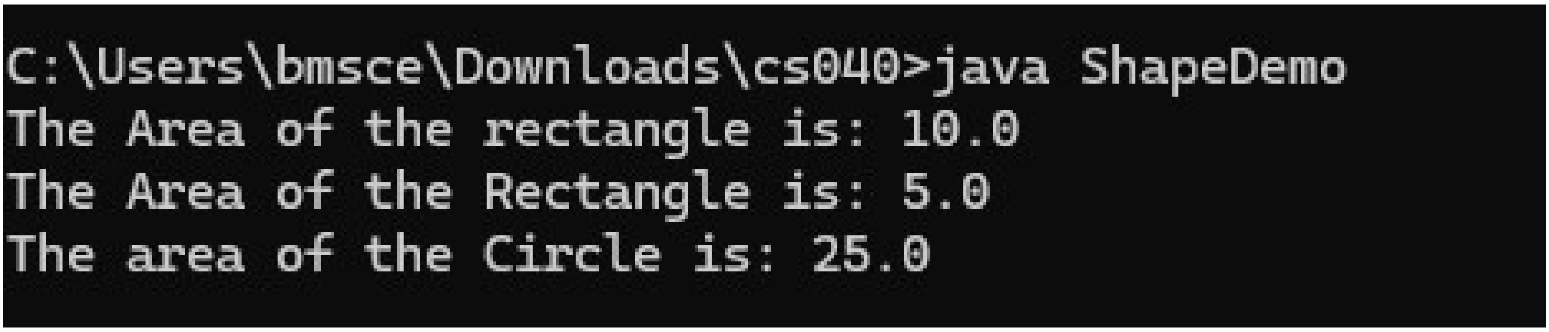
Output:

~~Enter the dimensions of Rectangle~~
~~2 5~~

```
Area of rectangle : 10
Area of triangle : 5
Area of circle : 78.5398
```

Q Develop a
maintains 2
and savings &
The account
facilities
account prov
i) Create a
acc no. & ty
classes wor
specifiers to

```
=> import ja
   class Ac
     private
     private
     private
     public

         this
         this
         this
     }
     public
       return
     }
     public
       if (
import java.util.Scanner;
abstract class Shape  {
        double a;
        double b;
        abstract void printArea();
}
class Rectangle extends Shape{
        double l;
        double br;
  Rectangle(double a, double b){
        l=a;
        br=b;
}

        void printArea(){
                System.out.println("The Area of the rectangle is: "+l*br);
}
}
class Triangle extends Shape{
        double h;
        double b;
        Triangle(double a, double b){
                h=a;
                this.b=b;
        }
        void printArea(){
                System.out.println("The Area of the Triangle is: "+(h*b)/2.0);
        }
}
class Circle extends Shape{
        double r;
        Circle(double r){
                this.r=r;
        }
        void printArea(){
                System.out.println("The area of the Circle is: "+ r*r);

        }
}
class ShapeDemo{
        public static void main(String args[]){
          System.out.println("akul j Shingetagere
          1BM23CS023");
                Rectangle r = new Rectangle(2,5);
                Triangle t = new Triangle(2,5);
                Circle c = new Circle(5);

                r.printArea();
                t.printArea();
                c.printArea();

        }
}
```

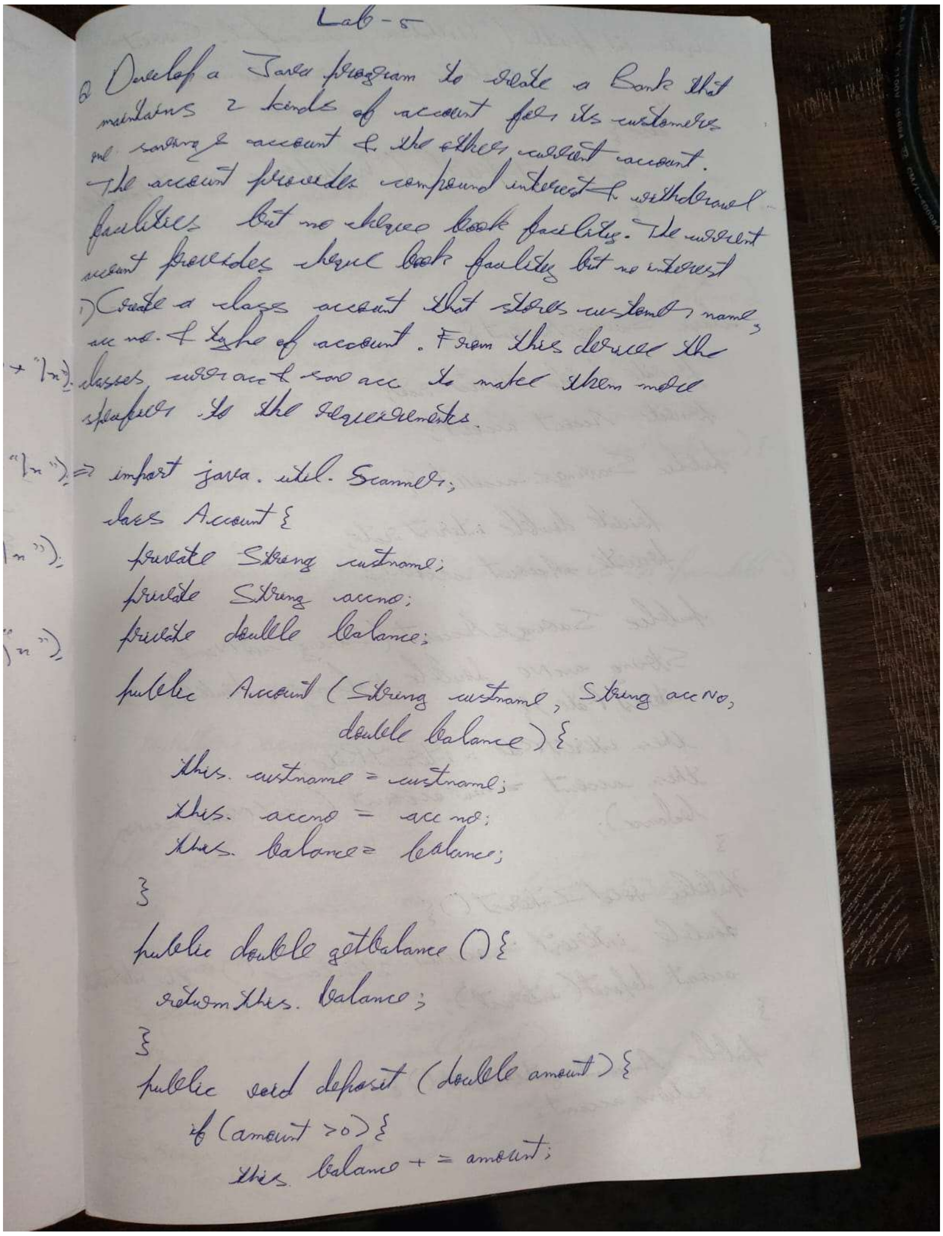


```
C:\Users\bmsce\Downloads\cs040>java ShapeDemo
The Area of the rectangle is: 10.0
The Area of the Rectangle is: 5.0
The area of the Circle is: 25.0
```

Program 5

Bank program

Algorithm

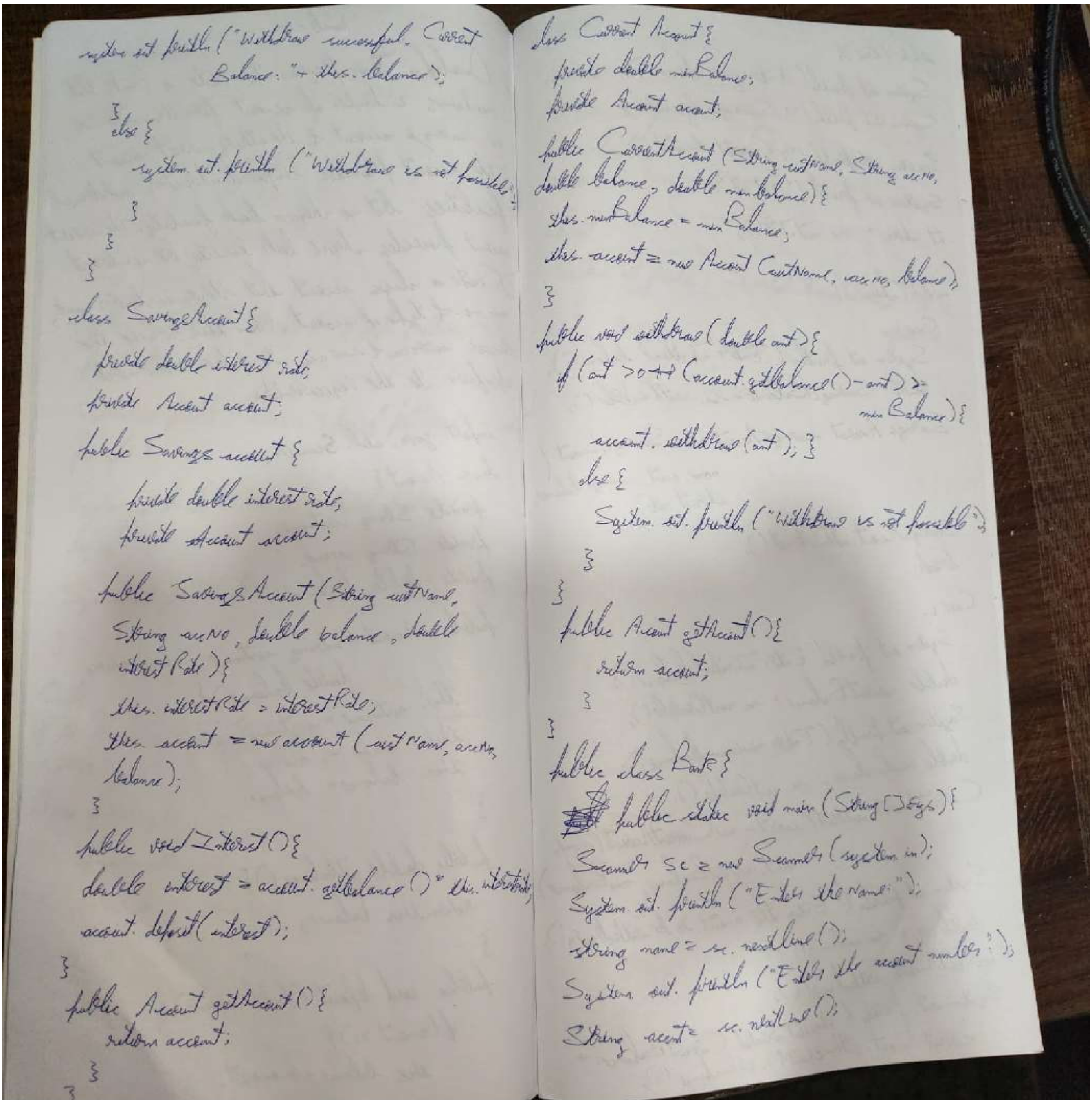


Lab-5

Q Develop a Java program to create a Bank that maintains 2 kinds of account for its customers one saving account & the other current account. The account provides compound interest & withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest

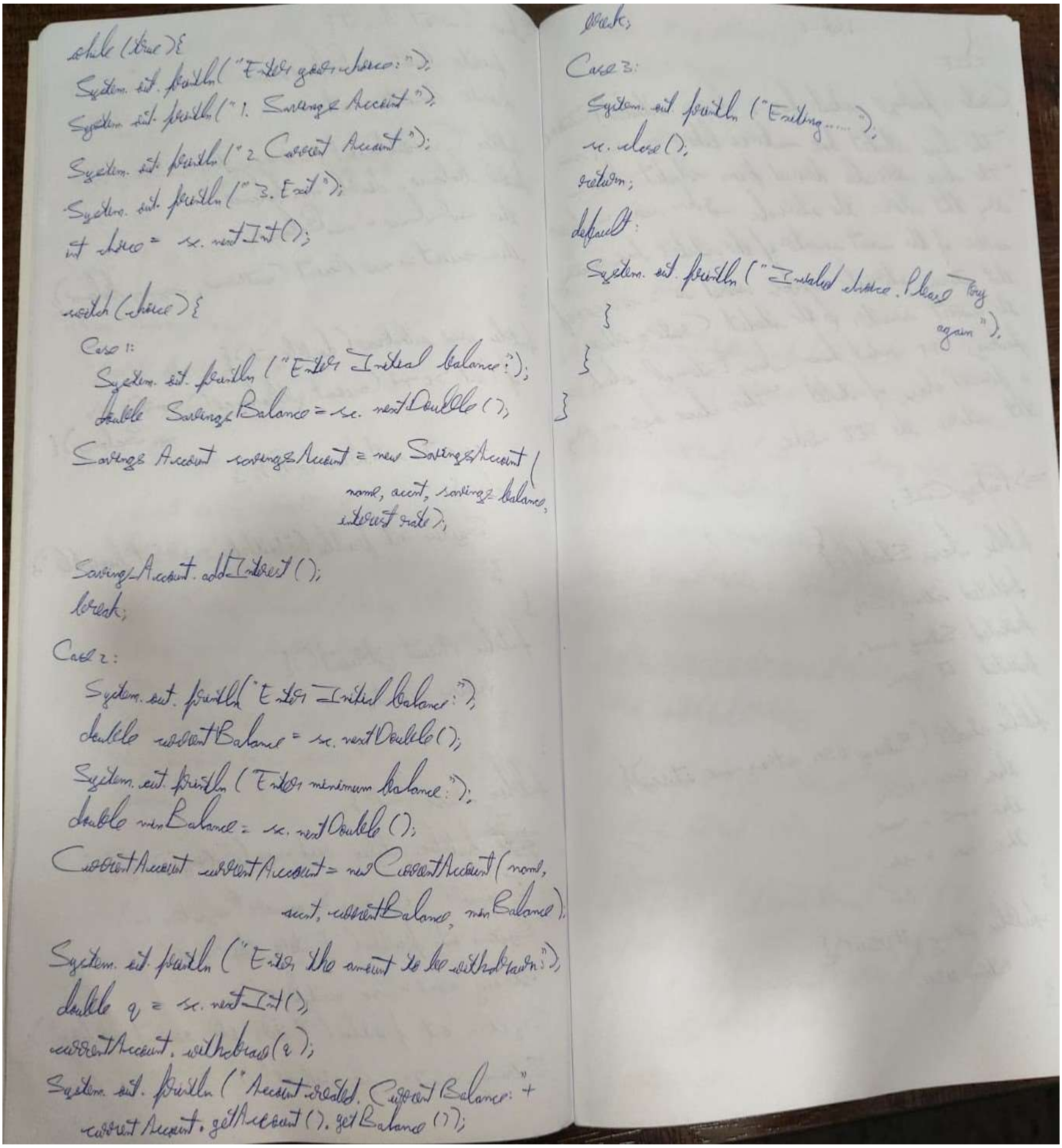
1) Create a class account that stores customer name, acc no. & type of account. From this derive the classes curr acc & sav acc to make them more specific to the requirements.

```
=> import java.util.Scanner;
   class Account {
     private String custname;
     private String accno;
     private double balance;

     public Account (String custname, String accNo,
                          double balance) {
        this. custname = custname;
        this. accno = acc no;
        this. balance = balance;
     }

     public double getbalance () {
       return this. balance;
     }
     public void deposit (double amount) {
        if (amount >0) {
           this. balance + = amount;
```



```
system.out.println("Withdraw successful. Current
                Balance: "+ this.balance);
    }
    else {
        system.out.println("Withdraw is not possible");
        }
    }
}
class SavingsAccount{
  private double interest rate;
  private Account account;
  public Savings account {
      private double interest rate;
      private Account account;

   public SavingsAccount(String custName,
   String accNo, double balance, double
   interestRate){
   this.interestRate = interestRate;
   this.account = new account (custName, accNo,
   balance);
  }
 public void Interest(){
 double interest = account.getBalance()* this.interestRate;
 account.deposit(interest);
}
public Account getAccount(){
   return account;
  }
}
```

```
class Current Account{
 private double minBalance;
 private Account acount;

public CurrentAccount (String custName, String accNo,
double balance, double minBalance){
 this.minBalance = minBalance;
 this.account = new Account (custName, accNo, balance);
}
public void withdraw(double amt){
  if (amt >0 && (account.getBalance()-amt) >=
                                          minBalance){
     account.withdraw(amt); }
    else {
     System.out.println("withdraw is not possible");
    }
}
public Account getAccount(){
    return account;
   }
}
public class Bank{
  public static void main(String[]args){
  Scanner sc = new Scanner(system.in);
  System.out.println("Enter the name:");
  String name = sc.nextLine();
  System.out.println("Enter the account number:");
  String accNo= sc.nextLine();
```

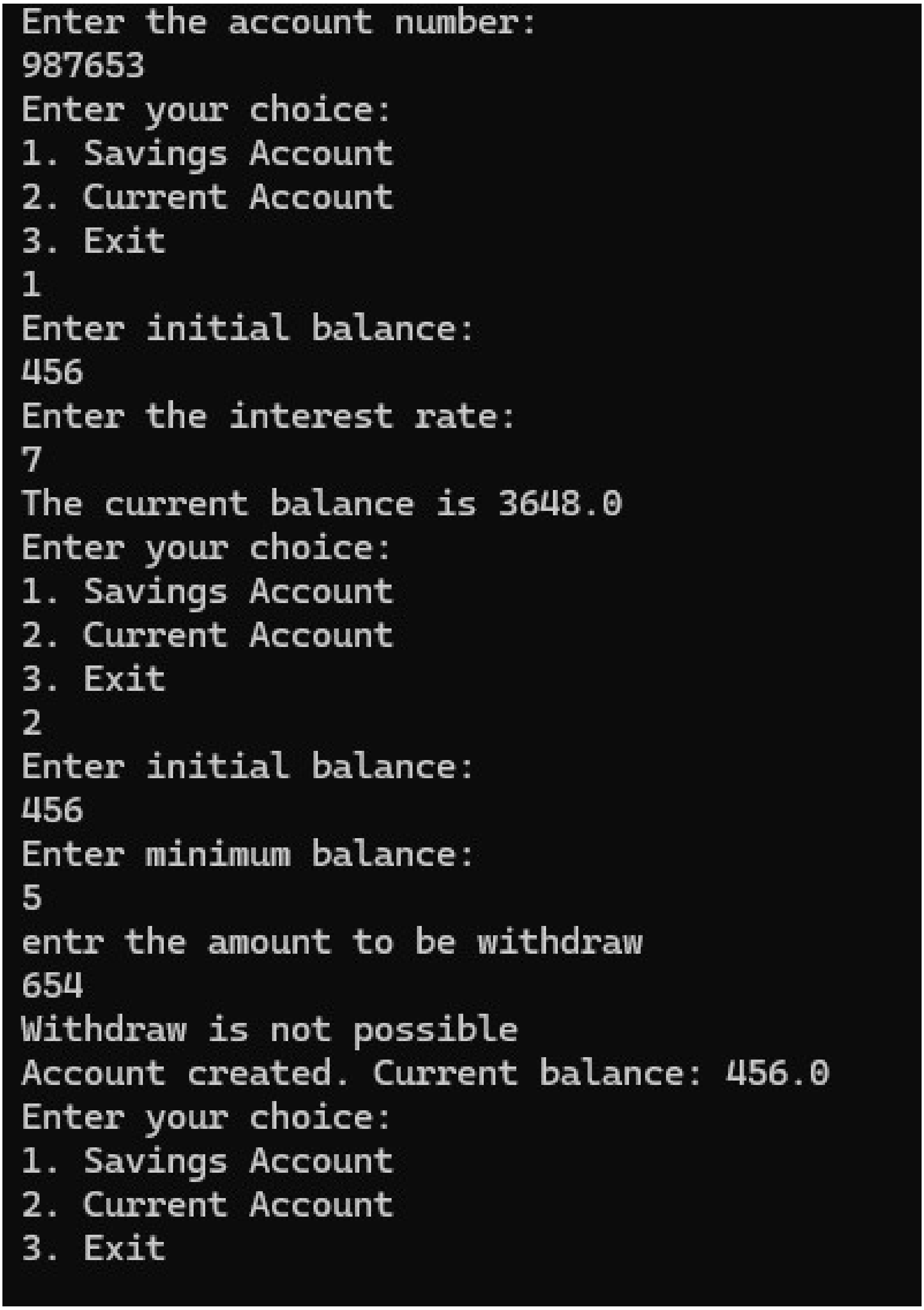


```
while (true){
 System.out.println("Enter your choice:");
 System.out.println("1. Savings Account");
 System.out.println("2 Current Account");
 System.out.println("3. Exit");
 int choice = sc.nextInt();

 switch (choice){
  Case 1:
   System.out.println ("Enter Initial balance:");
   double SavingsBalance = sc.nextDouble ();
  Savings Account savingsAccount = new SavingsAccount (
                                 name, acct, savings_balance,
                                 interest rate);

  Savings_Account.addInterest ();
  break;

 Case 2:
   System.out.println("Enter Initial balance:");
   double currentBalance = sc.nextDouble();
   System.out.println ("Enter minimum balance:");
   double minBalance = sc.nextDouble ();
 CurrentAccount currentAccount = new CurrentAccount (name,
                              acct, currentBalance, minBalance);
 System.out.println ("Enter the amount to be withdrawn:");
 double q = sc.nextInt();
 currentAccount.withdraw(q);
 System.out.println ("Account created. Current Balance:"+
   currentAccount.getAccount (). getBalance ());
 break;
 Case 3:
  System.out.println ("Exiting.....");
  sc.close();
  return;
 default:
  System.out.println ("Invalid choice. Please try
   }                                          again");
 }
}
```

Code:

```
import java.util.Scanner;

class Account {
    private String custName;
    private String accNo;
    private double balance;

    public Account(String custName, String accNo, double balance) {
        this.custName = custName;
        this.accNo = accNo;
        this.balance = balance;
    }

    public double getBalance() {
        return this.balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            this.balance += amount;
            System.out.println("The current balance is " + this.balance);
        } else {
            System.out.println("Amount should not be negative");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && (balance - amount) >= 0) {
            this.balance -= amount;
            System.out.println("Withdraw successful. Current balance: " + this.balance);
        } else {
            System.out.println("Withdraw is not possible");
        }
    }
}

class SavingsAccount {
    private double interestRate;
    private Account account;

    public SavingsAccount(String custName, String accNo, double balance, double interestRate) {
        this.interestRate = interestRate;
        this.account = new Account(custName, accNo, balance);
    }

    public void addInterest() {
        double interest = account.getBalance() * this.interestRate;
        account.deposit(interest);
    }
    public Account getAccount() {
        return account;
    }
}

class CurrentAccount {
    private double minBalance;
    private Account account;

    public CurrentAccount(String custName, String accNo, double balance, double minBalance) {
        this.minBalance = minBalance;
        this.account = new Account(custName, accNo, balance);
    }

    public void withdraw(double amt) {
        if (amt > 0 && (account.getBalance() - amt) >= minBalance) {
            account.withdraw(amt);
        } else {
            System.out.println("Withdraw is not possible");
        }
    }

    public Account getAccount() {
        return account;
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name:");
        String name = sc.nextLine();
        System.out.println("Enter the account number:");
        String accnt = sc.nextLine();

        while (true) {
            System.out.println("Enter your choice:");
            System.out.println("1. Savings Account");
            System.out.println("2. Current Account");
            System.out.println("3. Exit");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("Enter initial balance:");
                    double savingsBalance = sc.nextDouble();
                    System.out.println("Enter the interest rate:");
                    double interestRate = sc.nextDouble();
                    SavingsAccount savingsAccount = new SavingsAccount(name, accnt, savingsBalance, interestRate);
                    savingsAccount.addInterest();
                    break;
                case 2:
                    System.out.println("Enter initial balance:");
                    double currentBalance = sc.nextDouble();
                    System.out.println("Enter minimum balance:");
                    double minBalance = sc.nextDouble();
                    CurrentAccount currentAccount = new CurrentAccount(name, accnt, currentBalance, minBalance);

                    System.out.println("entr the amount to be withdraw");
                    double q = sc.nextInt();
                    currentAccount.withdraw(q);
                    System.out.println("Account created. Current balance: " + currentAccount.getAccount().getBalance());
                    break;

                case 3:
                    System.out.println("Exiting...");
                    sc.close();
                    return;

                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }
}
```

Output:

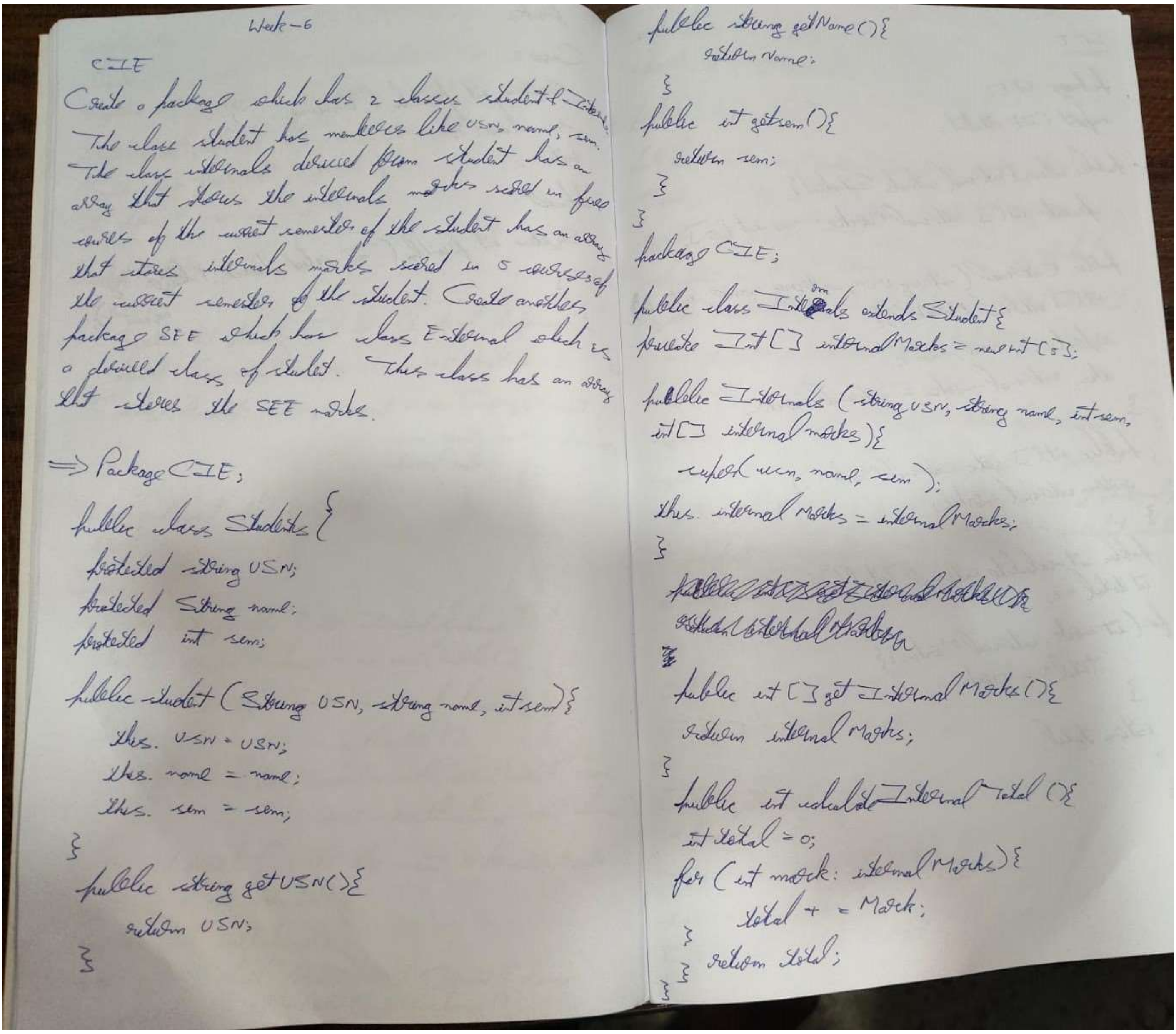


```
Enter the account number:
987653
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
1
Enter initial balance:
456
Enter the interest rate:
7
The current balance is 3648.0
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
2
Enter initial balance:
456
Enter minimum balance:
5
entr the amount to be withdraw
654
Withdraw is not possible
Account created. Current balance: 456.0
Enter your choice:
1. Savings Account
2. Current Account
3. Exit
```

Program 6

Packages

Algorithm

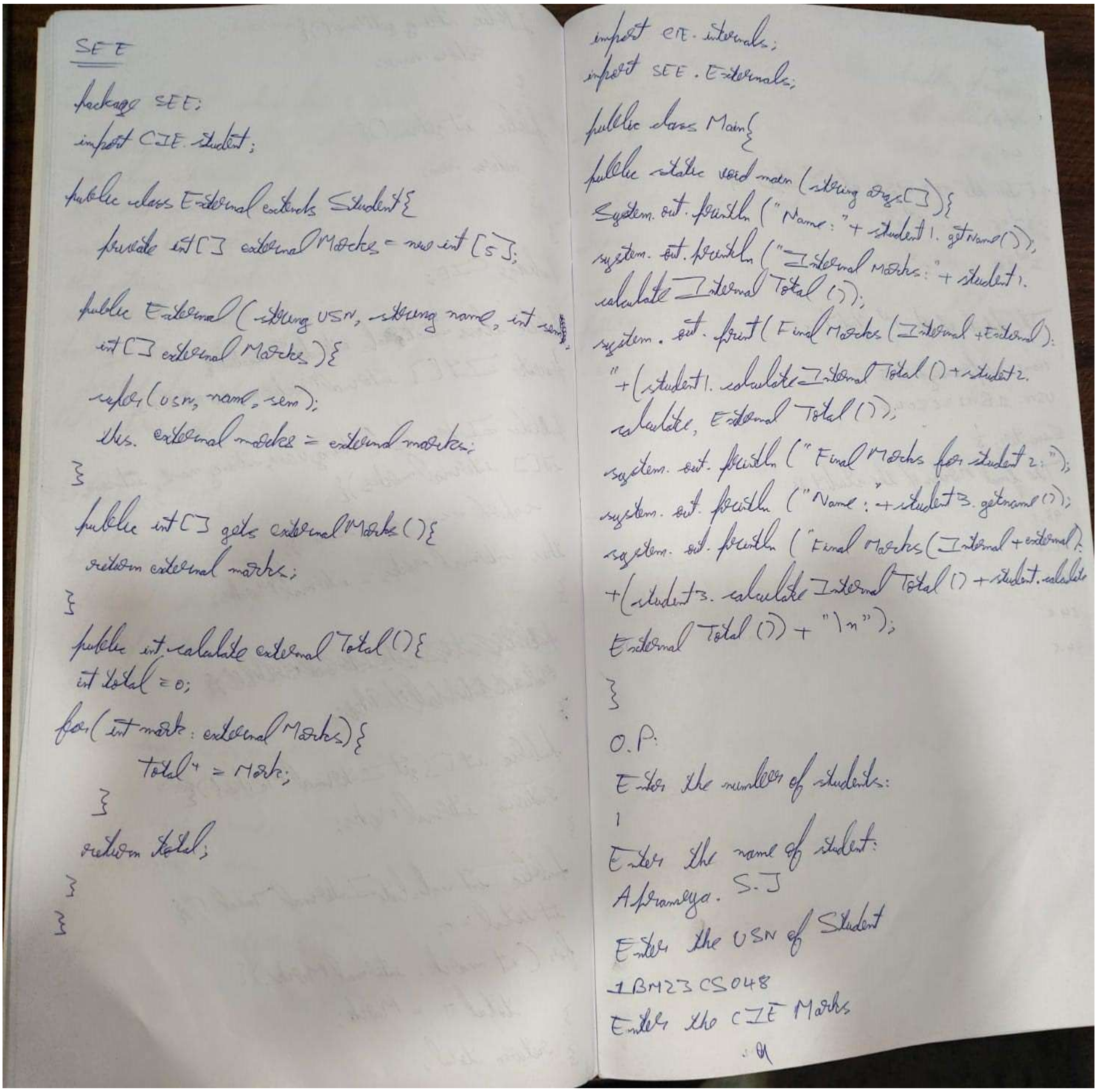


Week-6

CIE

Create a package which has 2 classes student & Internals. The class student has members like usn, name, sem. The class internals derived from student has an array that stores the internals marks scored in five courses of the current semester of the student has an array that stores internals marks scored in 5 courses of the current semester of the student. Create another package SEE which has class External which is a derived class of student. This class has an array that stores the SEE marks.

```
=> Package CIE;
public class Students {
  protected String USN;
  protected String name;
  protected int sem;

public student (String USN, string name, int sem){
   this. USN = USN;
   this. name = name;
   this. sem = sem;
}
public string getUSN(){
    return USN;
}
public string getName(){
    return Name;
 }
public int getsem(){
  return sem;
 }
}
package CIE;
public class Internals extends Student{
private Int[] internalMarks = new int[5];

public Internals (string USN, string name, int sem,
int[] internalmarks){
  super(usn, name, sem);
this. internalMarks = internalMarks;
}

~~public int[] getInternalMarks(){~~
~~return internalMarks~~
}
public int[] getInternalMarks(){
return internalMarks;
}
public int calculateInternalTotal(){
int total = 0;
for (int mark: internalMarks){
     total + = Mark;
 }
 return total;
}
```



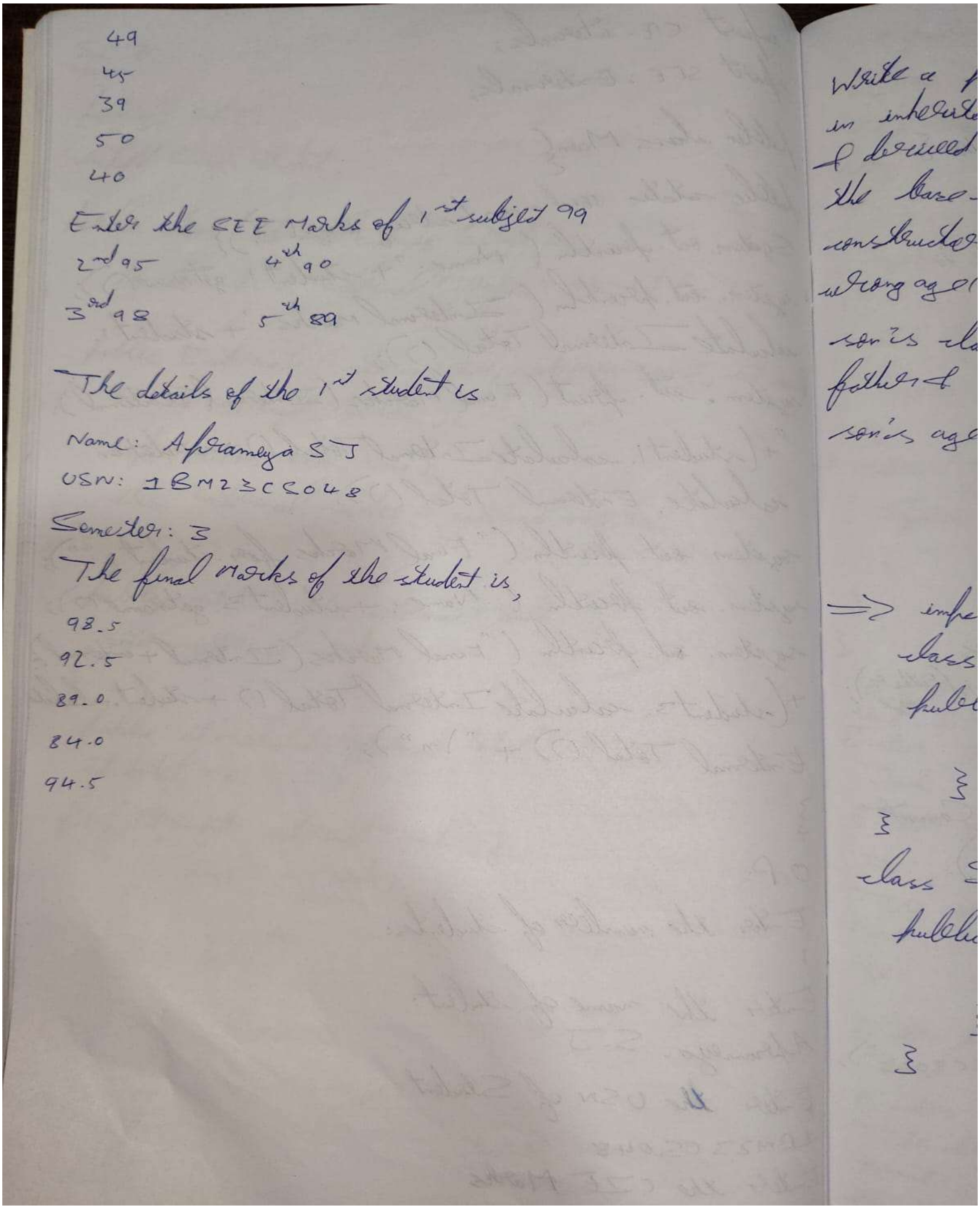
SEE

```
package SEE;
import CIE.student;

public class External extends Student{
  private int[] externalMarks = new int[5];

  public External(string USN, string name, int sem,
   int[] externalMarks){
   super(usn, name, sem);
   this.externalmarks = externalmarks;
  }
  public int[] gets externalMarks(){
   return externalmarks;
  }
  public int calculate externalTotal(){
  int total = 0;
for(int mark: externalMarks){
        Total+ = Mark;
    }
  return total;
 }
}
```

```
import CIE.internals;
import SEE.Externals;
public class Main{
public static void main(string args[]){
System.out.println("Name:"+ student1.getName());
system.out.println("Internal Marks:"+ student1.
calculateInternalTotal());
system.out.print(Final Marks(Internal+External):
 "+(student1.calculateInternalTotal()+student2.
 calculate, ExternalTotal());
system.out.println("Final Marks for student 2:");
system.out.println("Name:"+student3.getname());
system.out.println("Final Marks(Internal+external):
+(student3.calculateInternalTotal()+student.calculate
ExternalTotal() + "\n");

}
```

O.P:

```
Enter the number of students:
1
Enter the name of student:
Apramya. S.J
Enter the USN of Student
1BM23CS048
Enter the CIE Marks
```



```
49
45
39
50
40
Enter the SEE Marks of 1st subject 99
2nd 95        4th 90
3rd 98        5th 89

The details of the 1st student is
Name: Aprameya S J
USN: 1BM23CS048
Semester: 3
The final marks of the student is,
98.5
92.5
89.0
84.0
94.5
```

Write a p
in inherita
& derived
the base
constructor
wrong age(
son is cla
father &
son's age

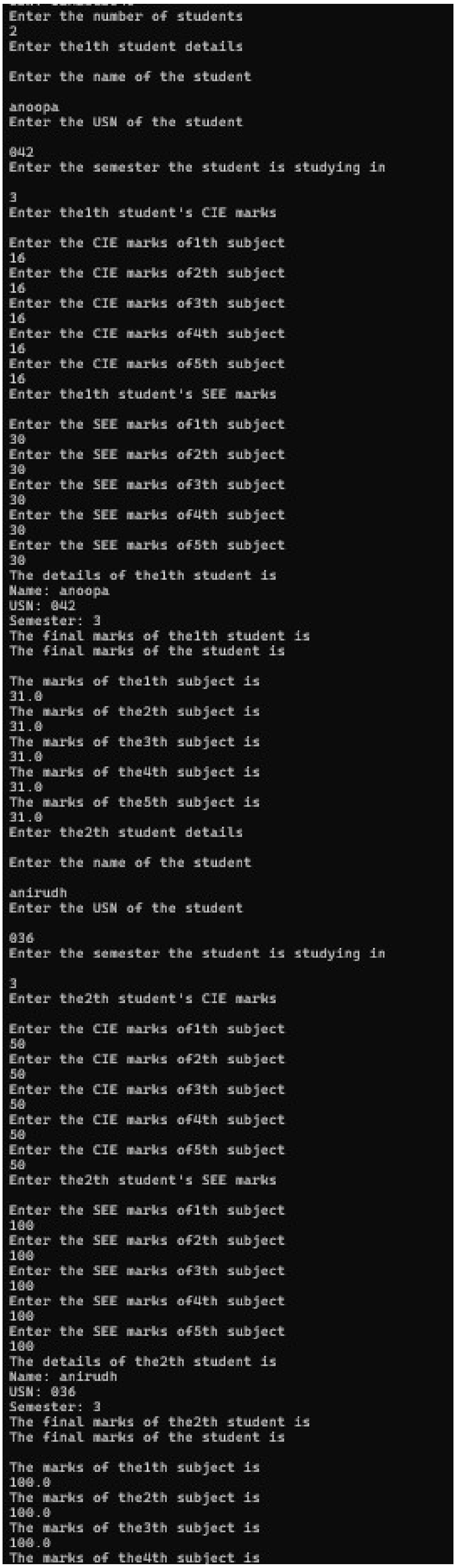
=> impl
class
publ

}
}
class S
publi

}

Code:

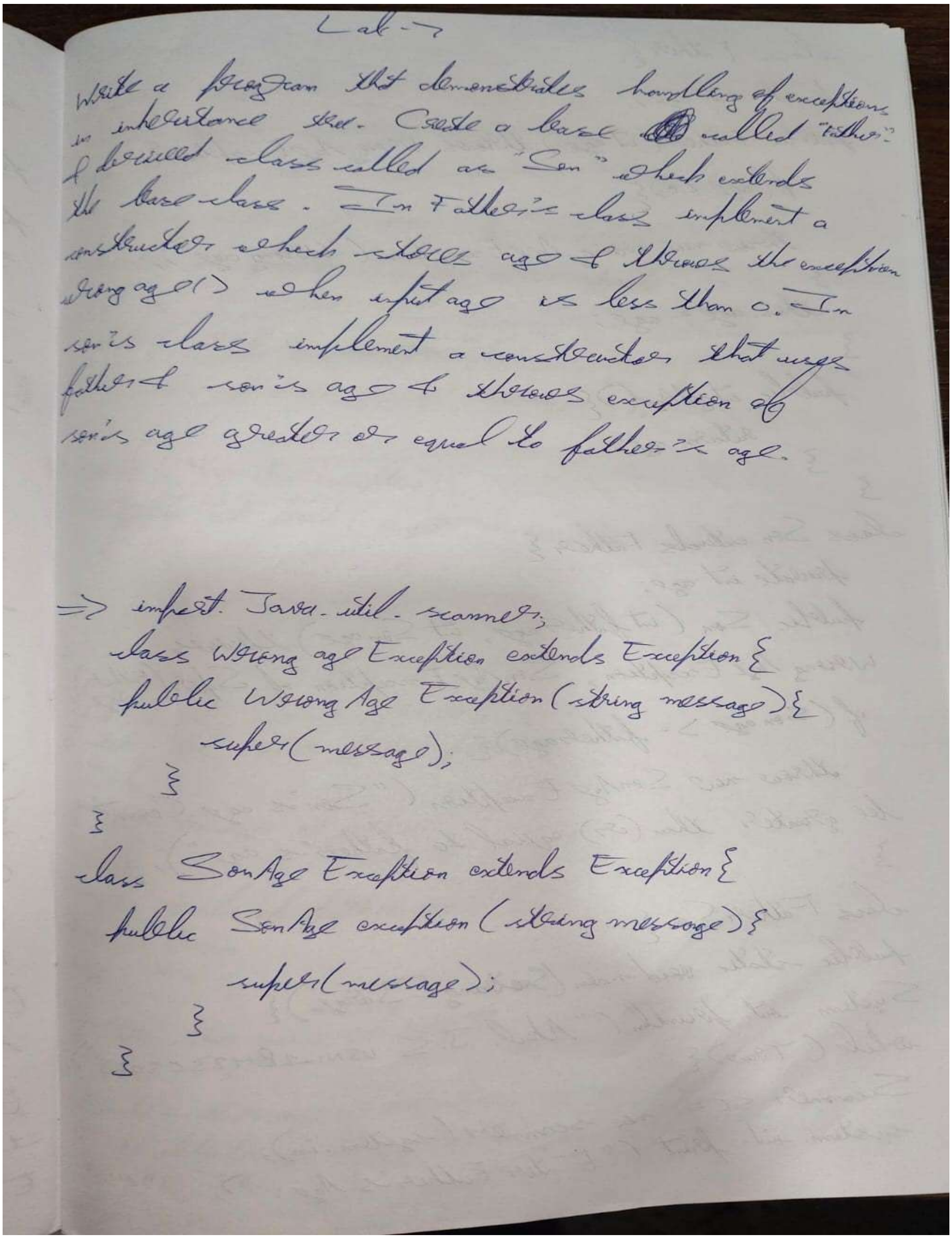
```
package CIE;
import java.util.Scanner;

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.println("Enter the name of the student \n");
            this.name=sc.nextLine();
            System.out.println("Enter the USN of the student \n");
            this.usn=sc.nextLine();
            System.out.println("Enter the semester the student is studying in \n");
            this.sem=sc.nextInt();
        }

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

package CIE;
import java.util.Scanner;

public class Internals extends Student {
        protected double ciemarks[] = new double[5];
        Scanner sc= new Scanner(System.in);
        public void inputCIEmarks(){
            for(int i=0; i<5; i++){
                System.out.println("Enter the CIE marks of"+ (i+1)+"th subject");
                this.ciemarks[i]= sc.nextDouble();
            }
        }
}


package SEE;
import CIE.*;
import java.util.Scanner;

public class Externals extends Internals{
                protected double seemarks[] = new double[5];
                protected double finalMarks[] = new double[5] ;

                public void inputSEEmarks() {
                    Scanner sc = new Scanner(System.in);
                    for(int i=0; i<5; i++){
                        System.out.println("Enter the SEE marks of"+ (i+1)+"th subject");
                        this.seemarks[i]= sc.nextDouble();
                    }
                }

                 public void definefinalmarks(){
                     for(int i=0; i<5; i++){
                        this.finalMarks[i] = ciemarks[i] + (seemarks[i]/2.0);
                    }

                }

                public void displayfinalmarks(){
                    System.out.println("The final marks of the student is \n");
                    for(int i=0; i<5; i++){
                        System.out.println("The marks of the" + (i+1)+"th subject is \t");
                        System.out.println(this.finalMarks[i]);
                    }
                }
}


import SEE.Externals;
import java.util.Scanner;
class Main{
        public static void main(String args[]){
            int n;
            Scanner sc = new Scanner(System.in);
            System.out.println("Name: saran tej");
            System.out.println("USN: 1BM23CS040");
            System.out.println("Enter the number of students \t");
            n= sc.nextInt();
            Externals e[] = new Externals[n];
            for(int i=0; i<n; i++){
                e[i] = new Externals();
                System.out.println("Enter the" +(i+1)+"th student details \n");
                e[i].inputStudentDetails();
                System.out.println("Enter the" +(i+1)+"th student's CIE marks\n");
                e[i].inputCIEmarks();
                System.out.println("Enter the" +(i+1)+"th student's SEE marks\n");
                e[i].inputSEEmarks();
                System.out.println("The details of the" +(i+1)+"th student is");
                e[i].displayStudentDetails();
                e[i].definefinalmarks();
                System.out.println("The final marks of the" +(i+1)+"th student is");
                e[i].displayfinalmarks();
            }

        }
}
```



```
Enter the number of students
2
Enter the1th student details

Enter the name of the student

anoopa
Enter the USN of the student

042
Enter the semester the student is studying in

3
Enter the1th student's CIE marks

Enter the CIE marks of1th subject
16
Enter the CIE marks of2th subject
16
Enter the CIE marks of3th subject
16
Enter the CIE marks of4th subject
16
Enter the CIE marks of5th subject
16
Enter the1th student's SEE marks

Enter the SEE marks of1th subject
30
Enter the SEE marks of2th subject
30
Enter the SEE marks of3th subject
30
Enter the SEE marks of4th subject
30
Enter the SEE marks of5th subject
30
The details of the1th student is
Name: anoopa
USN: 042
Semester: 3
The final marks of the1th student is
The final marks of the student is

The marks of the1th subject is
31.0
The marks of the2th subject is
31.0
The marks of the3th subject is
31.0
The marks of the4th subject is
31.0
The marks of the5th subject is
31.0
Enter the2th student details

Enter the name of the student

anirudh
Enter the USN of the student

036
Enter the semester the student is studying in

3
Enter the2th student's CIE marks

Enter the CIE marks of1th subject
50
Enter the CIE marks of2th subject
50
Enter the CIE marks of3th subject
50
Enter the CIE marks of4th subject
50
Enter the CIE marks of5th subject
50
Enter the2th student's SEE marks

Enter the SEE marks of1th subject
100
Enter the SEE marks of2th subject
100
Enter the SEE marks of3th subject
100
Enter the SEE marks of4th subject
100
Enter the SEE marks of5th subject
100
The details of the2th student is
Name: anirudh
USN: 036
Semester: 3
The final marks of the2th student is
The final marks of the student is

The marks of the1th subject is
100.0
The marks of the2th subject is
100.0
The marks of the3th subject is
100.0
The marks of the4th subject is
```

Program 7
Exception handling

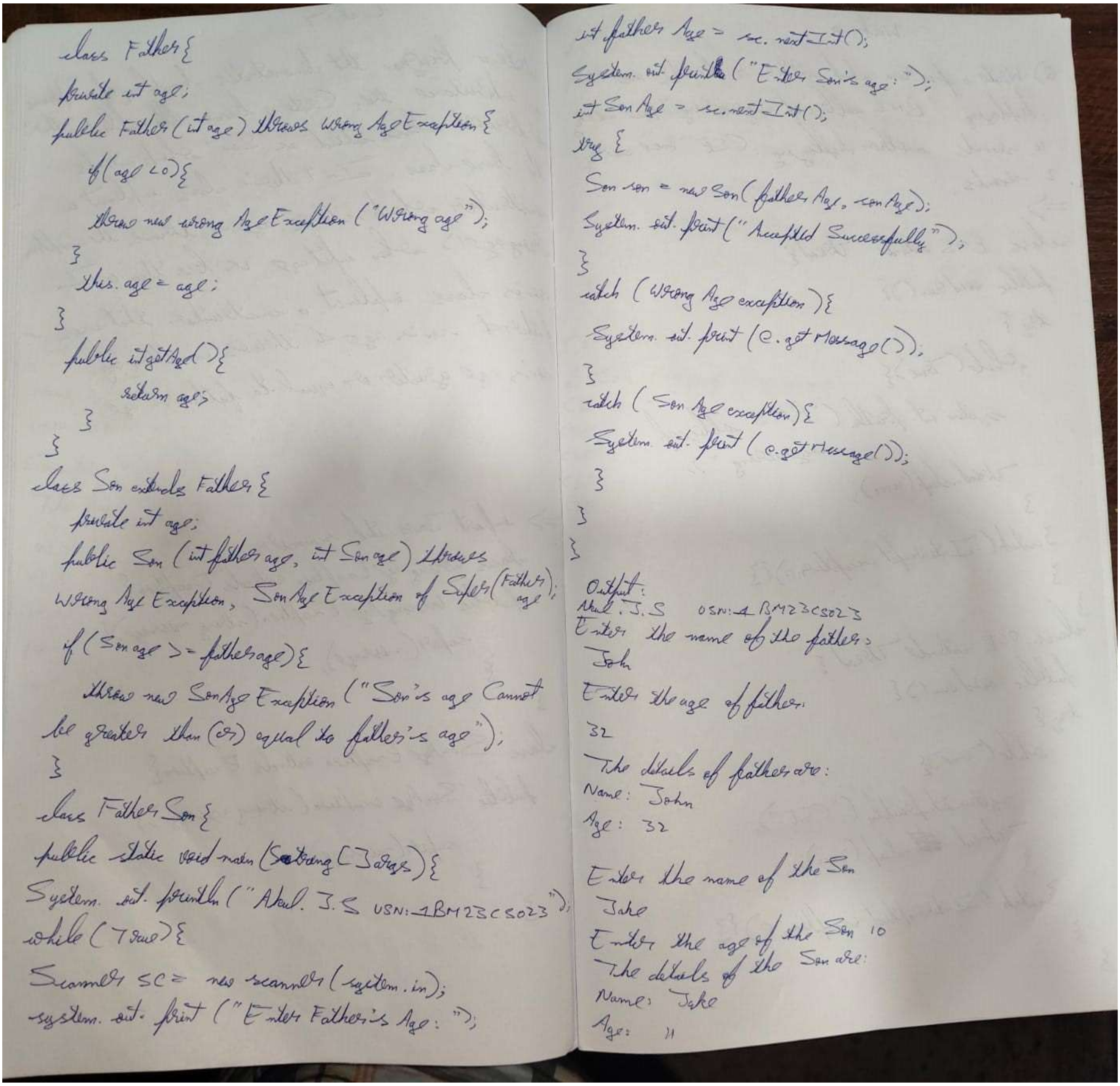
Algorithm



Lab-7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base called "Father" & derived class called as "Son" which extends the base class. In Father's class implement a constructor which takes age & throws the exception wrong age() when input age is less than 0. In son's class implement a constructor that uses father & son's age & throws exception if son's age greater or equal to father's age.

```
=> import. Java. util. scanner;
 class Wrong age Exception extends Exception {
  public Wrong Age Exception (string message) {
        super(message);
     }
 }
class SonAge Exception extends Exception {
  public SonAge exception (string message) {
          super(message);
       }
  }
```



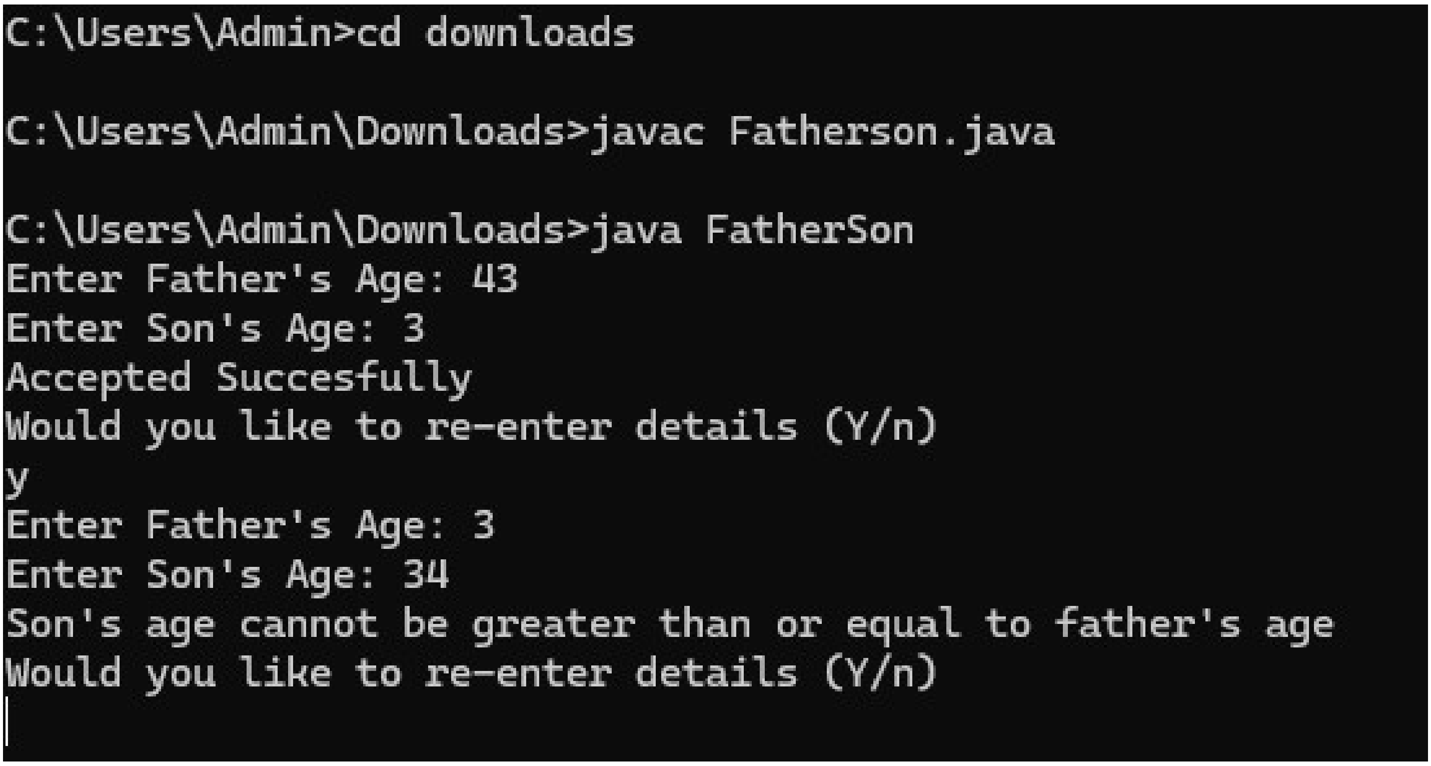
```
class Father{
private int age;
public Father(int age) throws WrongAgeException{
   if(age <0){
    throw new wrongAgeException ("Wrong age");
  }
  this.age = age;
 }
 public int getAge(){
       return age;
   }
}
class Son extends Father{
  private int age;
  public Son (int fatherage, int Sonage) throws
  WrongAgeException, SonAgeException of Super(Father age);
  if(Sonage >= fatherage){
    throw new SonAgeException ("Son's age Cannot
 be greater than (or) equal to father's age");
 }
class FatherSon{
public static void main(String[]args){
System.out.println("Akul. J.S USN:1BM23CS023");
while(True){
Scanner sc= new scanner(system.in);
system.out.print("Enter Father's Age: ");
int fatherAge = sc.nextInt();
System.out.println("Enter Son's age: ");
int SonAge = sc.nextInt();
try {
 Son son = new Son(fatherAge, sonAge);
 System.out.print("Accepted Successfully");
}
catch (WrongAge exception){
 System.out.print(e.getMessage());
}
catch (SonAge exception){
 System.out.print(e.getMessage());
  }
}
}
```

Output:

```
Akul. J.S    USN:1BM23CS023
Enter the name of the father:
 John
Enter the age of father:
 32
The details of father are:
Name: John
Age: 32

Enter the name of the Son
 Jake
Enter the age of the Son 10
The details of the Son are:
Name: Jake
Age:  11
```

```
import java.util.Scanner;
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}

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

class Father {
    private int age;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Wrong age");
        }
        this.age = age;
    }
    public int getAge() {
        return age;
    }
}

class Son extends Father {
    private int sonAge;
    public Son(int fatherAge, int sonAge) throws WrongAgeException, SonAgeException {
        super(fatherAge);
        if (sonAge >= fatherAge) {
            throw new SonAgeException("Son's age cannot be greater than or equal to father's age");
        }
        this.sonAge = sonAge;
    }
    public int getSonAge() {
        return sonAge;
    }
}
public class FatherSon{
    public static void main(String[] args) {
        while(true){
            Scanner sc = new Scanner(System.in);
            System.out.print("Enter Father's Age: ");
            int fatherAge = sc.nextInt();
            System.out.print("Enter Son's Age: ");
            int sonAge = sc.nextInt();
            try {
                Son son = new Son(fatherAge, sonAge);
                System.out.println("Accepted Succesfully");
            }
            catch (WrongAgeException e) {
                System.out.println(e.getMessage());
            }
            catch (SonAgeException e) {
                System.out.println(e.getMessage());
            }
            System.out.println("Would you like to re-enter details (Y/n)");
            String input = sc.next();
            if (input.equalsIgnoreCase("n")) {
                break;
            }
        }
    }
}
```

Output

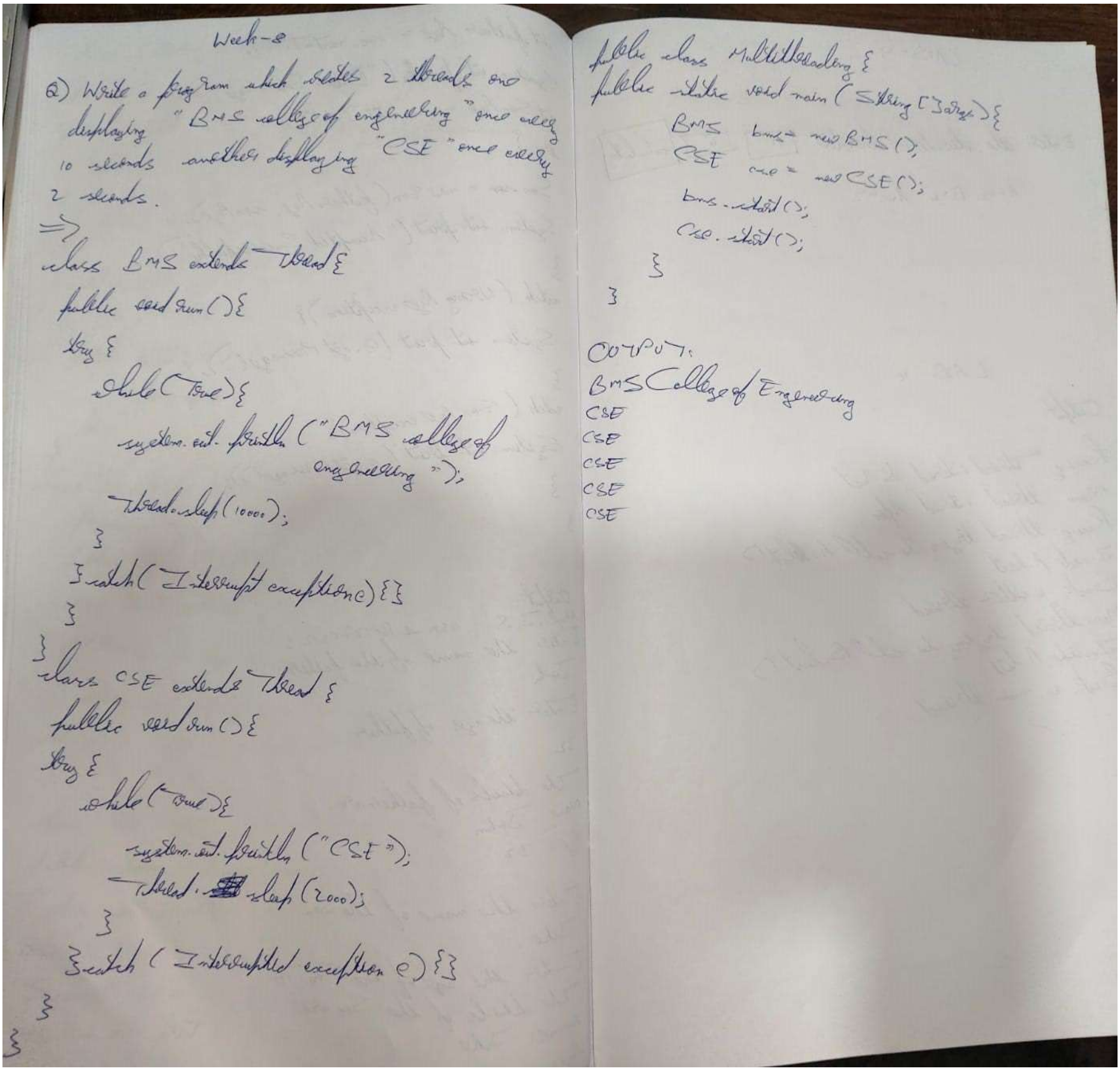


```
C:\Users\Admin>cd downloads

C:\Users\Admin\Downloads>javac Fatherson.java

C:\Users\Admin\Downloads>java FatherSon
Enter Father's Age: 43
Enter Son's Age: 3
Accepted Succesfully
Would you like to re-enter details (Y/n)
y
Enter Father's Age: 3
Enter Son's Age: 34
Son's age cannot be greater than or equal to father's age
Would you like to re-enter details (Y/n)
```

Program 8

Multithreading

Algorithm



Week-8

Q) Write a program which creates 2 threads one displaying "BMS college of engineering" once every 10 seconds another displaying "CSE" once every 2 seconds.

⟹

```
class BMS extends Thread{
 public void run(){
 try {
   while(True){
     system.out.println("BMS college of
                       engineering");
     Thread.sleep(10000);
   }
  }catch(Interrupt exceptione){}
 }
}
class CSE extends Thread {
 public void run(){
 try {
   while(true){
     system.out.println("CSE");
     Thread.sleep(2000);
   }
  }catch (Interrupted exception e){}
 }
}
```

```
public class Multithreading {
public static void main (String []args){
      BMS  bms = new BMS();
      CSE  cse = new CSE();
      bms.start();
      cse.start();
   }
 }
```

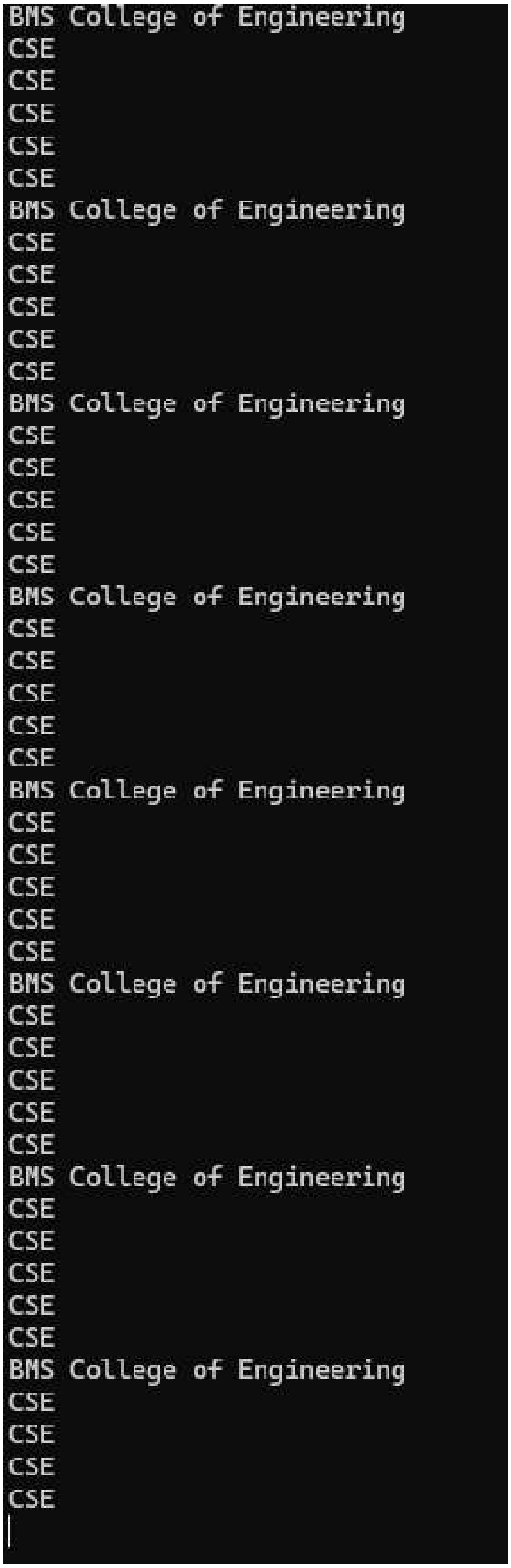
OUTPUT:

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

Code:

```
class BMS extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS College of Engineering");
                Thread.sleep(10000); // Sleep for 10 seconds
            }
        }catch (InterruptedException e) {}
    }
}
class CSE extends Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("CSE");
                Thread.sleep(2000); // Sleep for 2 seconds
            }
        }catch (InterruptedException e) {}
    }
}

public class Multithreading{
    public static void main(String[] args) {
        System.out.println("saran tej 1BMCS23CS040");
        BMS bms = new BMS();
        CSE cse = new CSE();
        bms.start();
        cse.start();
    }
```

output

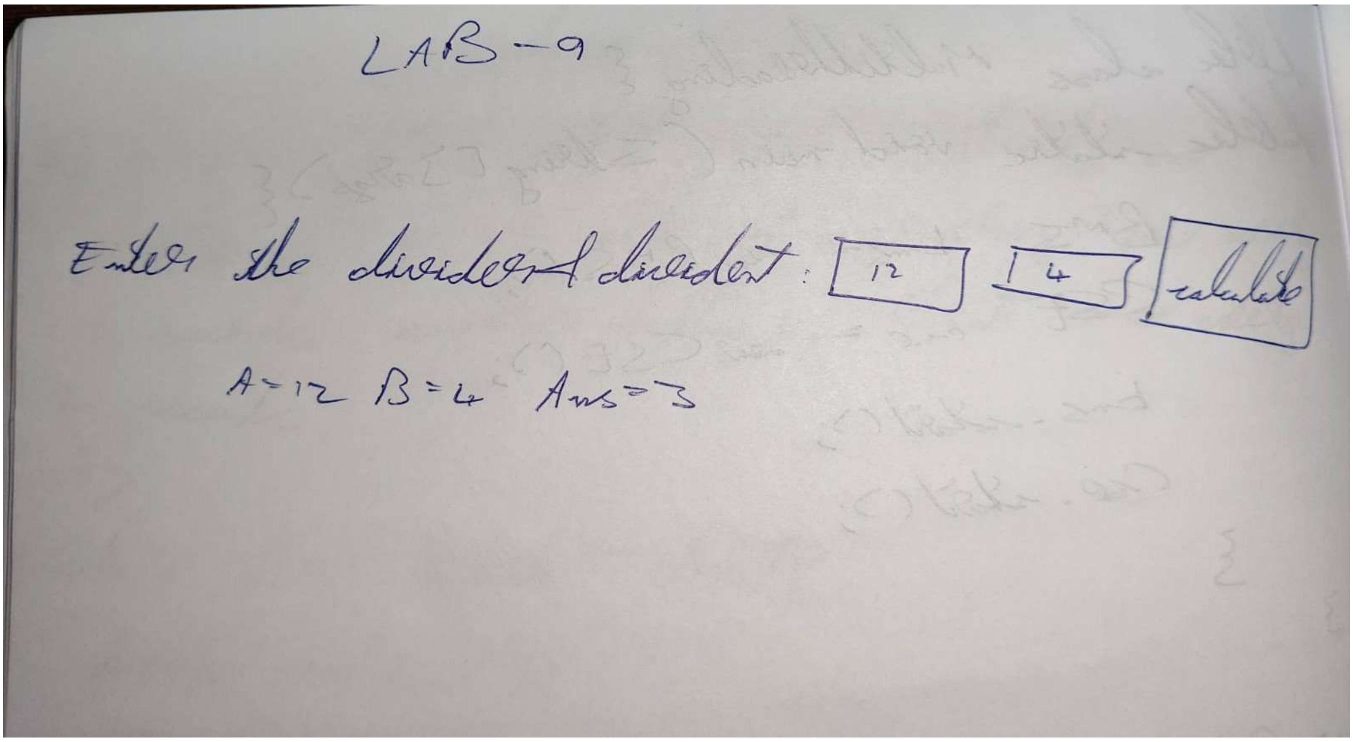


```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of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

Program 9

Integer division with user interface

Algorithm



LAB - 9

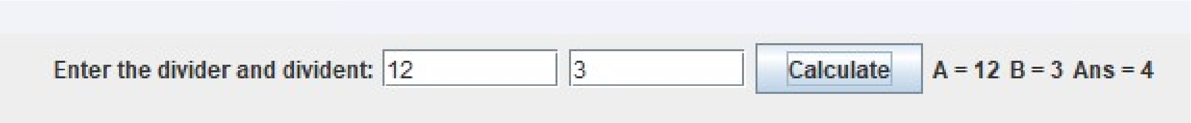
Enter the dividend & divident : 12 4 calculate

A=12 B=4 Ans=3

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();
}
});
}
}
```

Output



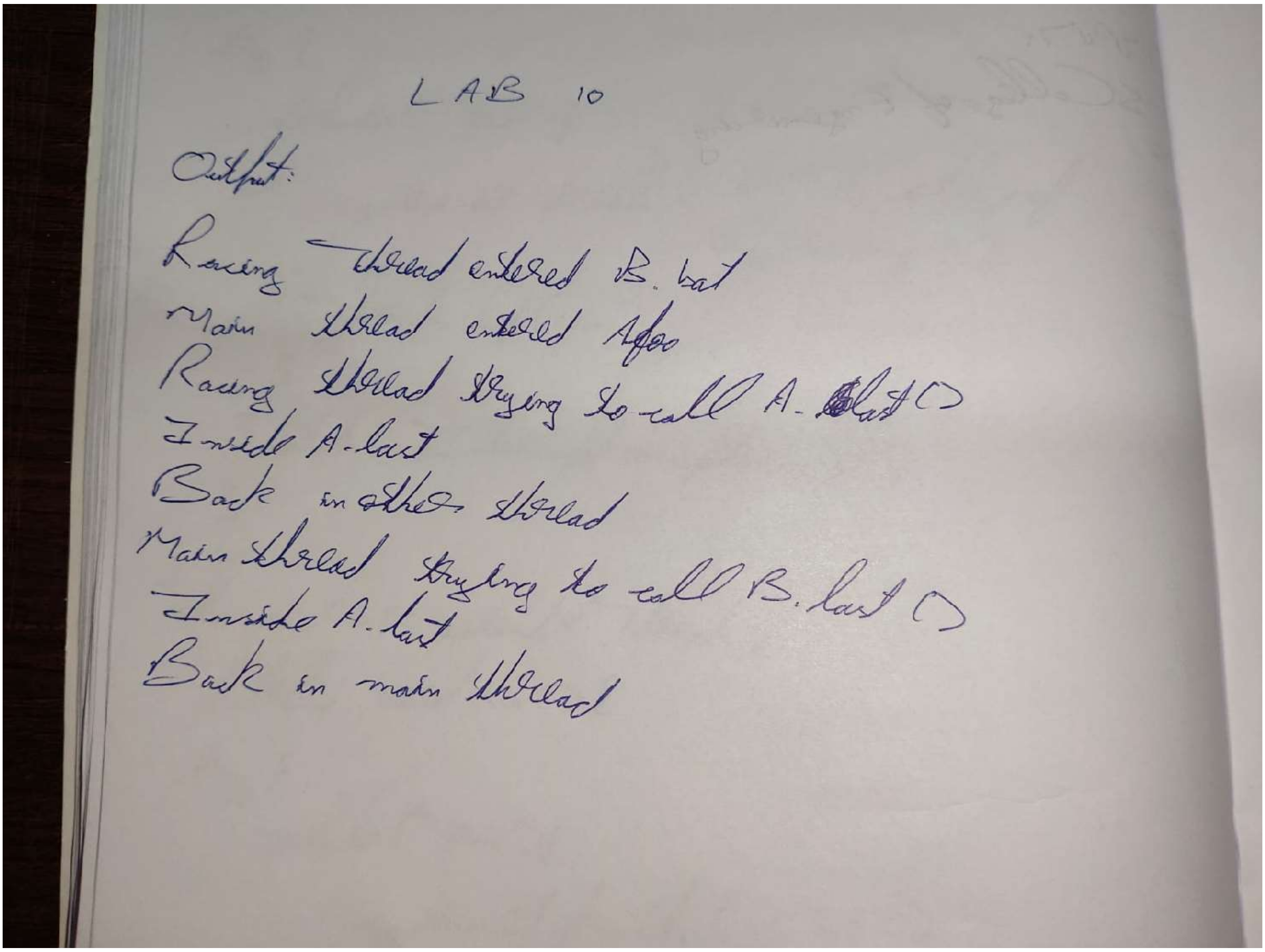
Enter the divider and divident: 12 3

A = 12 B = 3 Ans = 4

Program 10

Inter process communication and deadlock

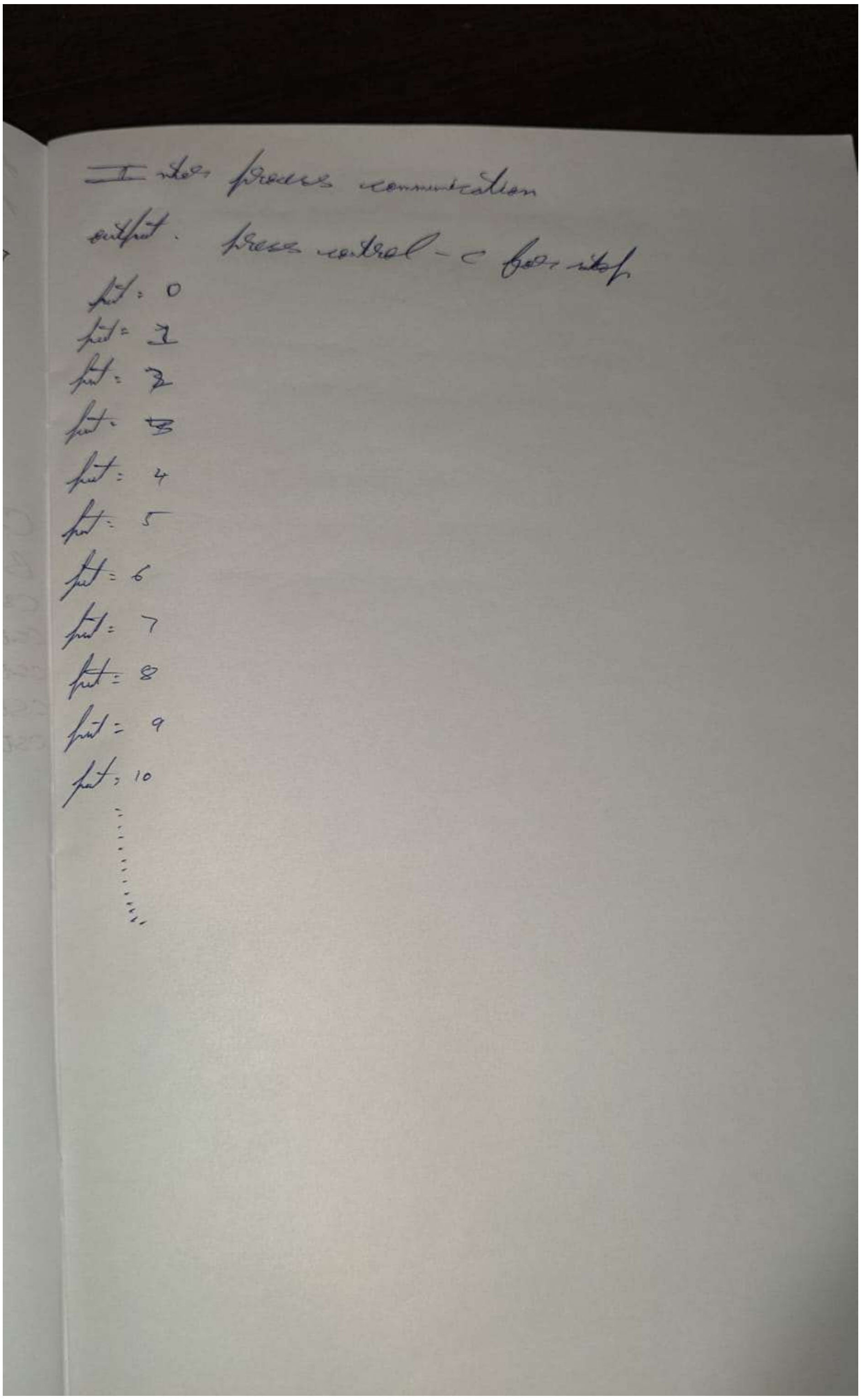
Algorithm



LAB 10

Output:

```
Racing thread entered B.last
Main thread entered A.foo
Racing thread trying to call A.last()
Inside A.last
Back in other thread
Main thread trying to call B.last()
Inside A.last
Back in main thread
```



Inter process communication

output: press control-c for stop

```
put: 0
put= 1
put= 2
put= 3
put: 4
put= 5
put= 6
put= 7
put= 8
put= 9
put: 10
:
:
```

Code:

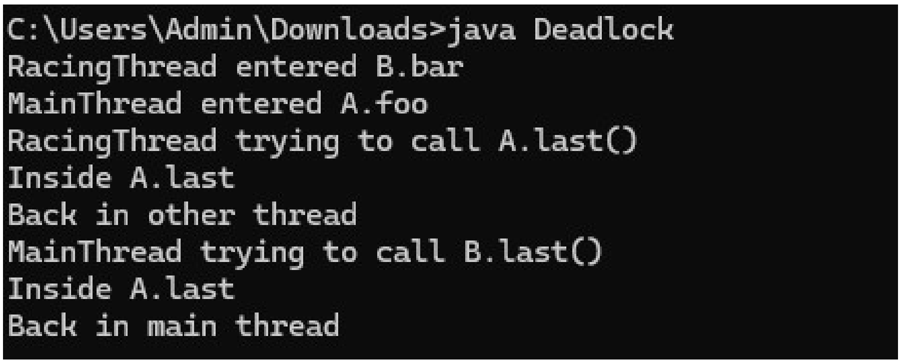
```
class Q {

int n;

boolean valueSet = false;

synchronized int get() {

while(!valueSet)

try {

System.out.println("\nConsumer waiting\n");

wait();

} catch(InterruptedException e) {

System.out.println("InterruptedException caught");

}
System.out.println("Got: " + n);

valueSet = false;

System.out.println("\nIntimate Producer\n");

notify();

return n;

}

synchronized void put(int n) {

while(valueSet)

try {

System.out.println("\nProducer waiting\n");

wait();

} catch(InterruptedException e) {

System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}
class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}
}
class Consumer implements Runnable {
Q q;
Consumer(Q q) {
```

```
this.q = q;

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

}

public void run() {

int i=0;

while(i<15) {

int r=q.get();

System.out.println("consumed:"+r);

i++;

}

}

}

class PCFixed{
public static void main(String args[]) {
System.out.println("Name:Anirudh");
System.out.println("USN:1BM23CS036");
Q q = new Q();

new Producer(q);

new Consumer(q);

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

}

}

class A {

synchronized void foo(B b) {

String name =
Thread.currentThread().getName();
```

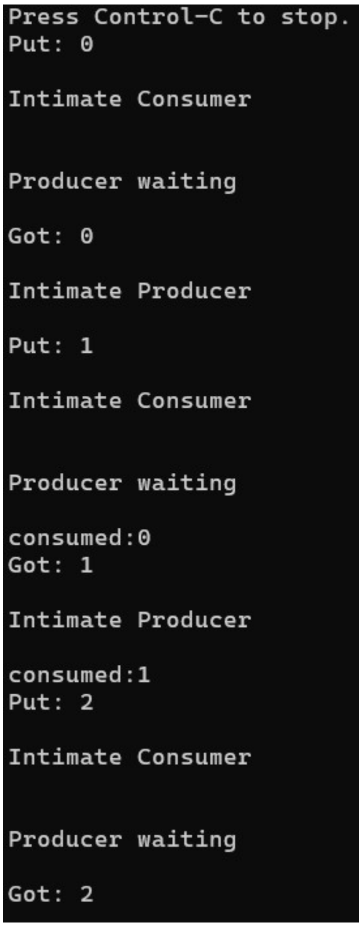
```
System.out.println(name + " entered A.foo");

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 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();

}}
```

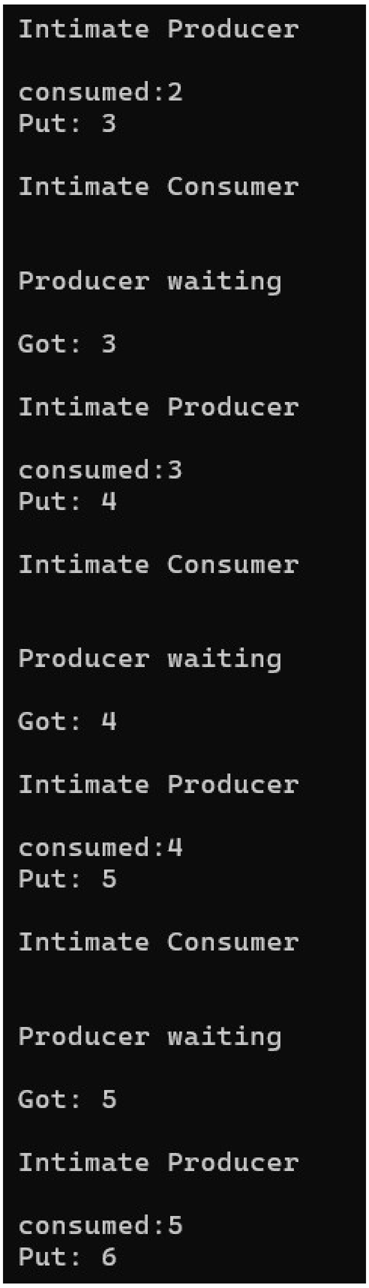
Output



```
C:\Users\Admin\Downloads>java Deadlock
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
Inside A.last
Back in other thread
MainThread trying to call B.last()
Inside A.last
Back in main thread
```



```
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


Producer waiting

Got: 4

Intimate Producer

consumed:4
Put: 5

Intimate Consumer


Producer waiting

Got: 5

Intimate Producer

consumed:5
Put: 6
```