VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

KATARAJU M (1BM21CS088)

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 Oct 2022-Feb 2023

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 lab” carried out by KATARAJU M(1BM21CS088), 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 during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Object oriented java programming Lab - (21CS3PCOOJ) work prescribed for the said degree.

Basavaraj Jakalli
Assistant Professor
Department of CSE
BMSCE, Bengaluru `

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

Index Sheet

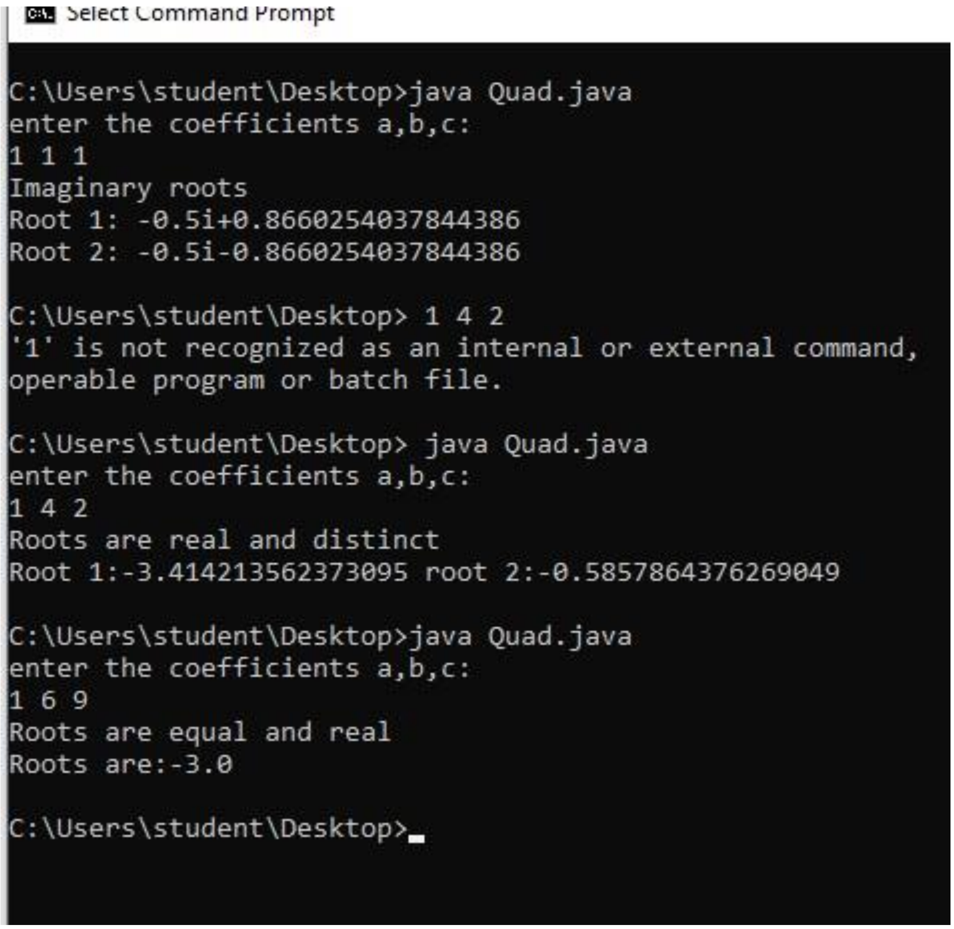
Course Outcome

CO1	Apply the knowledge of Java concepts to find the solution for a given problem.
CO2	Analyze the given Java application for correctness/functionalities.
CO3	Develop Java programs / applications for a given requirement.
CO4	Conduct practical experiments for demonstrating features of Java.

LAB PROGRAM 1: QUADRATIC EQUATIONS

CODE:

```
import java.util.Scanner;
import    java.lang.Math;
public class Trial
{ public static void main(String[] args)
        {
            Scanner s = new Scanner(System.in);
            System.out.println("Enter the coefficients: ");
            float a = s.nextFloat(); float b = s.nextFloat();
            float c = s.nextFloat(); double r1,r2; float d =
            (b*b)-(4.0f*a*c); if(d>0)
            { r1=(-b+Math.sqrt(d))/(2*a); r2=(-b-
                Math.sqrt(d))/(2*a);
                System.out.println("Roots are Real");
                System.out.println("Root 1: "+r1+" Root 2: "+r2);
            }        else
             if(d==0)
            {
                r1=(-b)/(2*a);
                System.out.println("Roots are Equal");
                System.out.println("Root is: "+r1);
            }


        else
            {
                double e =(-b)/(2.0f*a); double f
                =(Math.sqrt(-d))/(2*a);
                System.out.println("Roots are imaginary");
                System.out.println("Root 1: "+e+"i+"+f);
                System.out.println("Root 2: "+e+"i-"+f);
            }
        }
    }
```

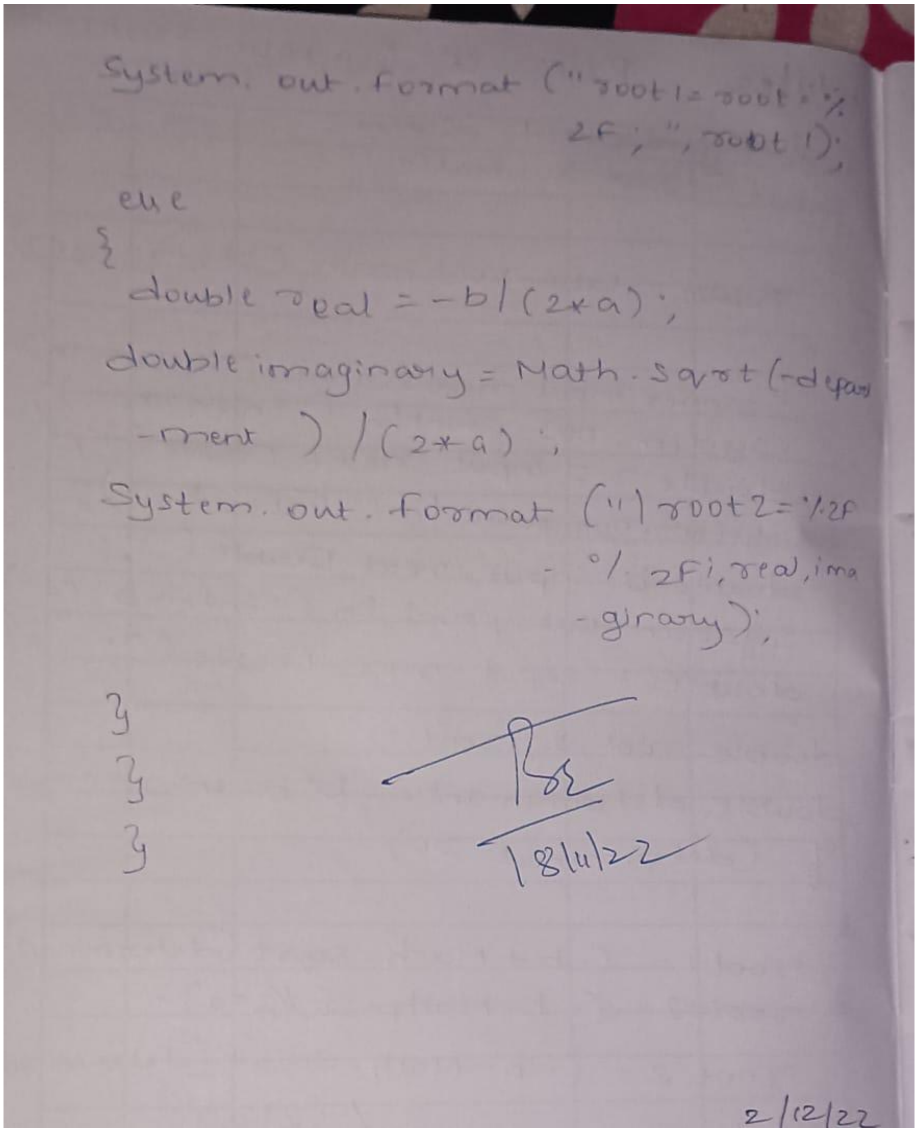


```
C:\Users\student\Desktop>java Quad.java
enter the coefficients a,b,c:
1 1 1
Imaginary roots
Root 1: -0.5i+0.8660254037844386
Root 2: -0.5i-0.8660254037844386

C:\Users\student\Desktop> 1 4 2
'1' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\student\Desktop> java Quad.java
enter the coefficients a,b,c:
1 4 2
Roots are real and distinct
Root 1:-3.414213562373095 root 2:-0.5857864376269049

C:\Users\student\Desktop>java Quad.java
enter the coefficients a,b,c:
1 6 9
Roots are equal and real
Roots are:-3.0

C:\Users\student\Desktop>
```

Observation Book:

18/11/22 ROOT OF QUADRATIC EQUATION

```
import java . until Scanner ;
Public class Quadratic
{
   Public static void main (string args[])
  {
    Scanner input new scanner ( System in)
    system. out. print ln ("Enter a: ");
    double a = input next Double ();
    System. out. print ln (" Enter b= ");
    double b= input . next Double ();
    System. out. print ln (" Enter b c= ");
    double C= input .next Double ();
 double root 1, root ;
 double determin ant = b*b - 4*a*c ;
  if (determinat >0)
 {
    root 1 = (-b + Math. sqrt (determinant))
    ~~root 2 = (- b. Math. sq~~ /(2*a) ;
    root 2 = (-b - Math. sqrt (determinant)
                         / (2*a) ;
 System. out. format ("root 1 = % 2f and
                   root 2 = % 2F", root 1,
                     root2);
 }
   else if
  (determinant = = 0)
 { root 1 = root 2 = - b / (2*a);
```



```
System. out. format ("root1= root = %
                      2f ; ", root 1);

else
{
 double real = -b/(2*a);
double imaginary = Math. sqrt(-discri
 -ment ) / (2*a) ;
System. out. format ("root2= %.2f
                    - % 2fi, real, ima
                    -ginary);

}
 }
 }
```

18/11/22

2/12/22

LAB PROGRAM 2: SGPA CALCULATION

CODE:

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

        String USN; String name;
        int[] credits = new int[20];
        int[] marks = new int[20];
        void input(int n)
        {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter Student USN: ");
        USN = s.nextLine();
        System.out.print("Enter Student Name: "); name
        = s.nextLine();
        for(int i=0;i<n;i++)
        {
        System.out.print("Enter the Subject "+(i+1)+" marks and credits
   respectively: "); marks[i] = s.nextInt();
        credits[i] = s.nextInt();
        }
        }
float calculate(int n)
        {
        int sum_of_credits = 0;
        float     result=0.0f;
        for(int i=0;i<n;i++)
        {
        sum_of_credits+=credits[i];
```

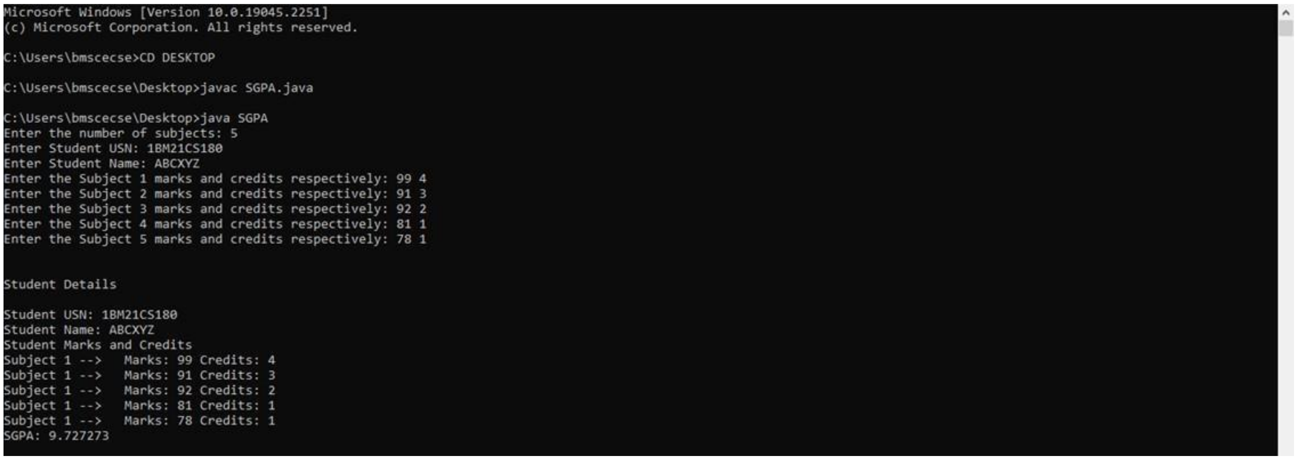
```
  if(calculate_grade_point(marks[i])==-1)
          return -1.0f;

else
{
          result = result +(float) (calculate_grade_point(marks[i])*credits[i]);
}
}

return (result/sum_of_credits);
}


int calculate_grade_point(int marks)
{
if(marks>=90)
return 10;
else if ((marks>=80)&&(marks<90))
return 9;
else if ((marks>=70)&&(marks<80))
return 8;
else if ((marks>=60)&&(marks<70))
return 7;
```

```
else if ((marks>=50)&&(marks<60))

return 6;

else if ((marks>=40)&&(marks<50))

return 5;

return -1;

}


void display(int n,float result)

{

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

System.out.println("Student Details");

System.out.println();
System.out.println("Student USN: "+USN);

System.out.println("Student Name: "+name);

System.out.println("Student Marks and Credits");

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

{

System.out.println("Subject 1 -->\tMarks: "+marks[i]+" Credits: "+credits[i]);

}

System.out.println("SGPA: "+result);

}
```

```
   }

   public class Lab_02_SGPA

   {

        public static void main(String[] args)

        {

        Scanner s = new Scanner(System.in);

        Student s1 = new Student();

        System.out.print("Enter the number of subjects: "); int

        n = s.nextInt();

        s1.input(n);

        float result = s1.calculate(n); if(result

        == -1.0f)

        {

        System.out.println();

        System.out.println("The Student has failed in a subject. SGPA cannot be
   calculated!");

        System.exit(0);

        }
s1.display(n,result);

        }
```

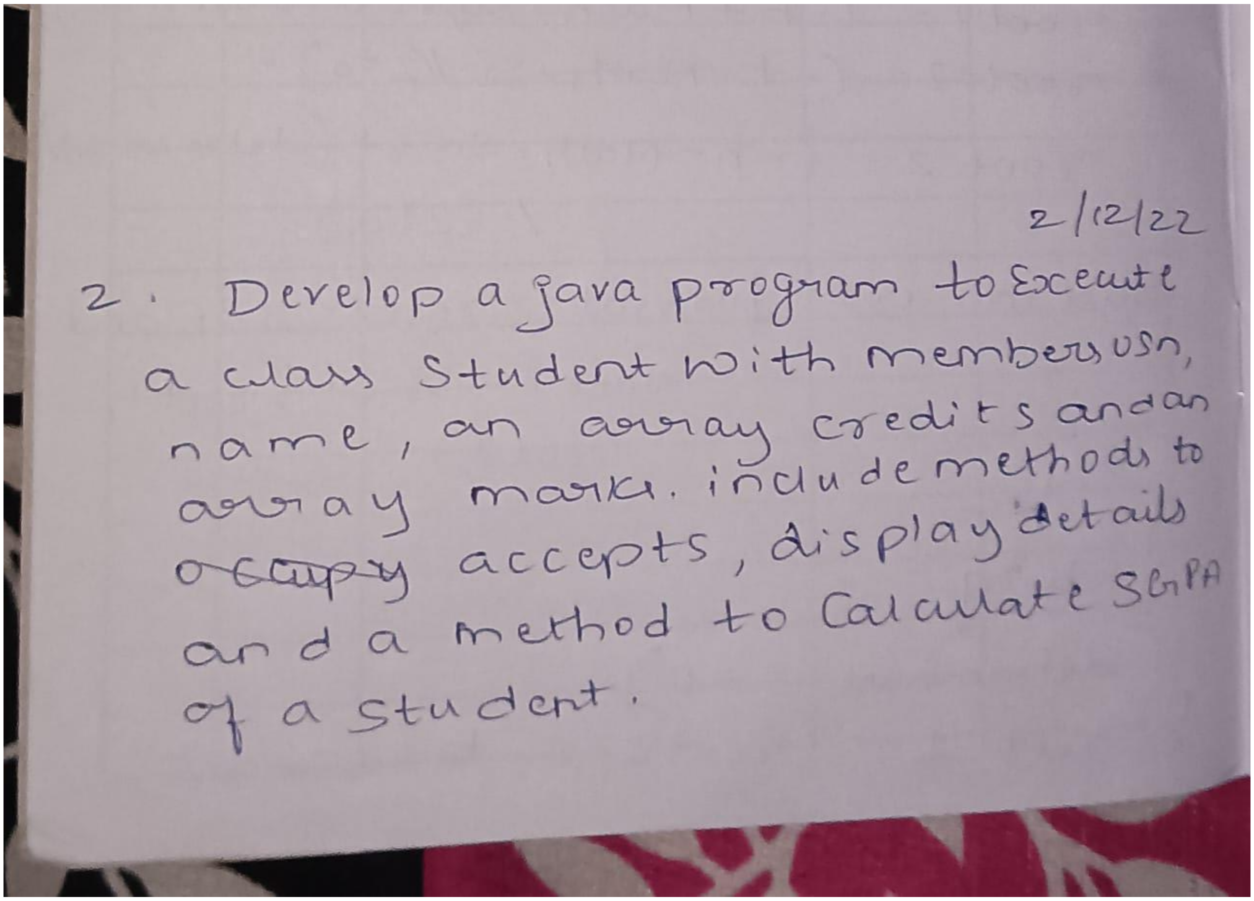
```
}
```

Output:



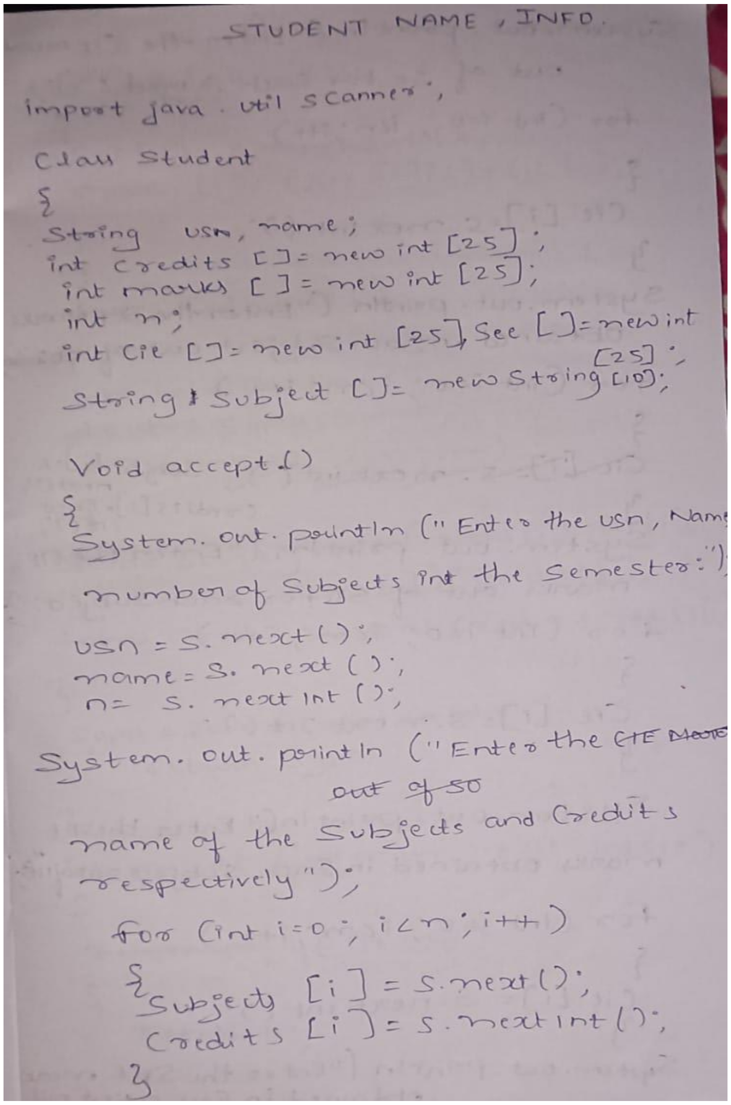
```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmscecse>CD DESKTOP

C:\Users\bmscecse\Desktop>javac SGPA.java

C:\Users\bmscecse\Desktop>java SGPA
Enter the number of subjects: 5
Enter Student USN: 1BM21CS180
Enter Student Name: ABCXYZ
Enter the Subject 1 marks and credits respectively: 99 4
Enter the Subject 2 marks and credits respectively: 91 3
Enter the Subject 3 marks and credits respectively: 92 2
Enter the Subject 4 marks and credits respectively: 81 1
Enter the Subject 5 marks and credits respectively: 78 1


Student Details

Student USN: 1BM21CS180
Student Name: ABCXYZ
Student Marks and Credits
Subject 1 -->   Marks: 99 Credits: 4
Subject 1 -->   Marks: 91 Credits: 3
Subject 1 -->   Marks: 92 Credits: 2
Subject 1 -->   Marks: 81 Credits: 1
Subject 1 -->   Marks: 78 Credits: 1
SGPA: 9.727273
```

Observation Book:



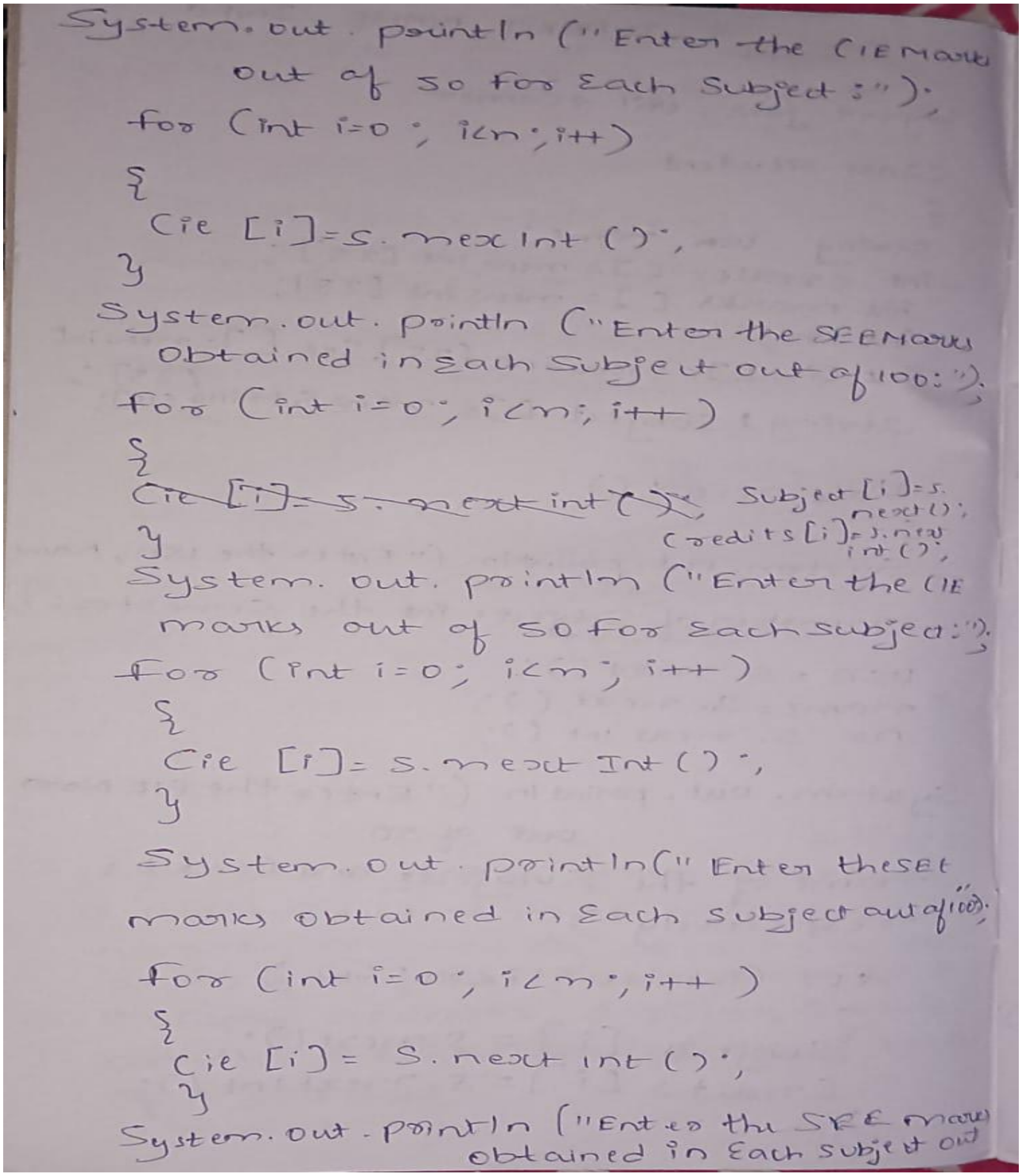
2/12/22

2. Develop a java program to Execute a class Student with members usn, name, an array credits and an array marks. include methods to ~~occupy~~ accepts, display details and a method to Calculate SGPA of a student.

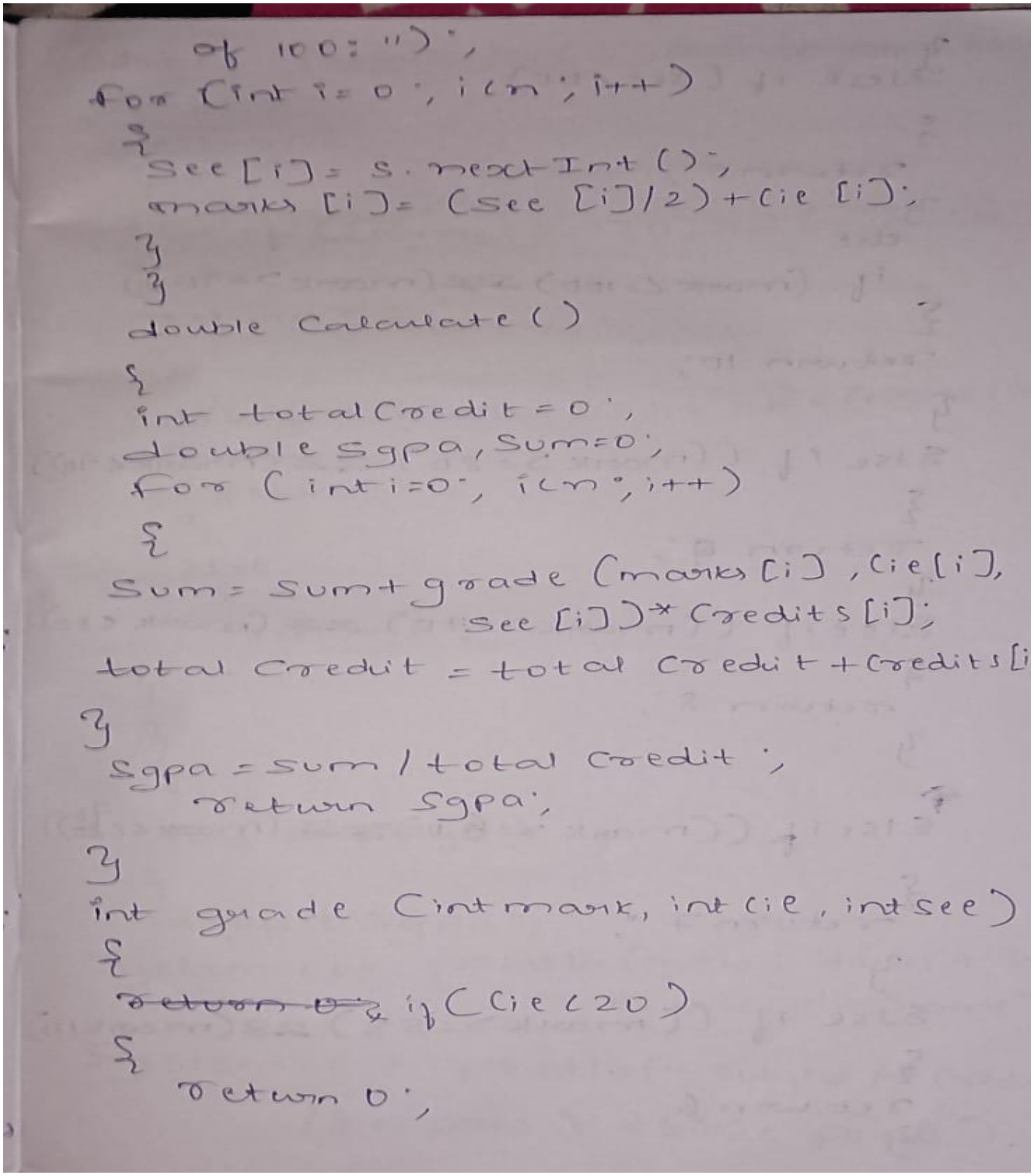


STUDENT NAME , INFO.

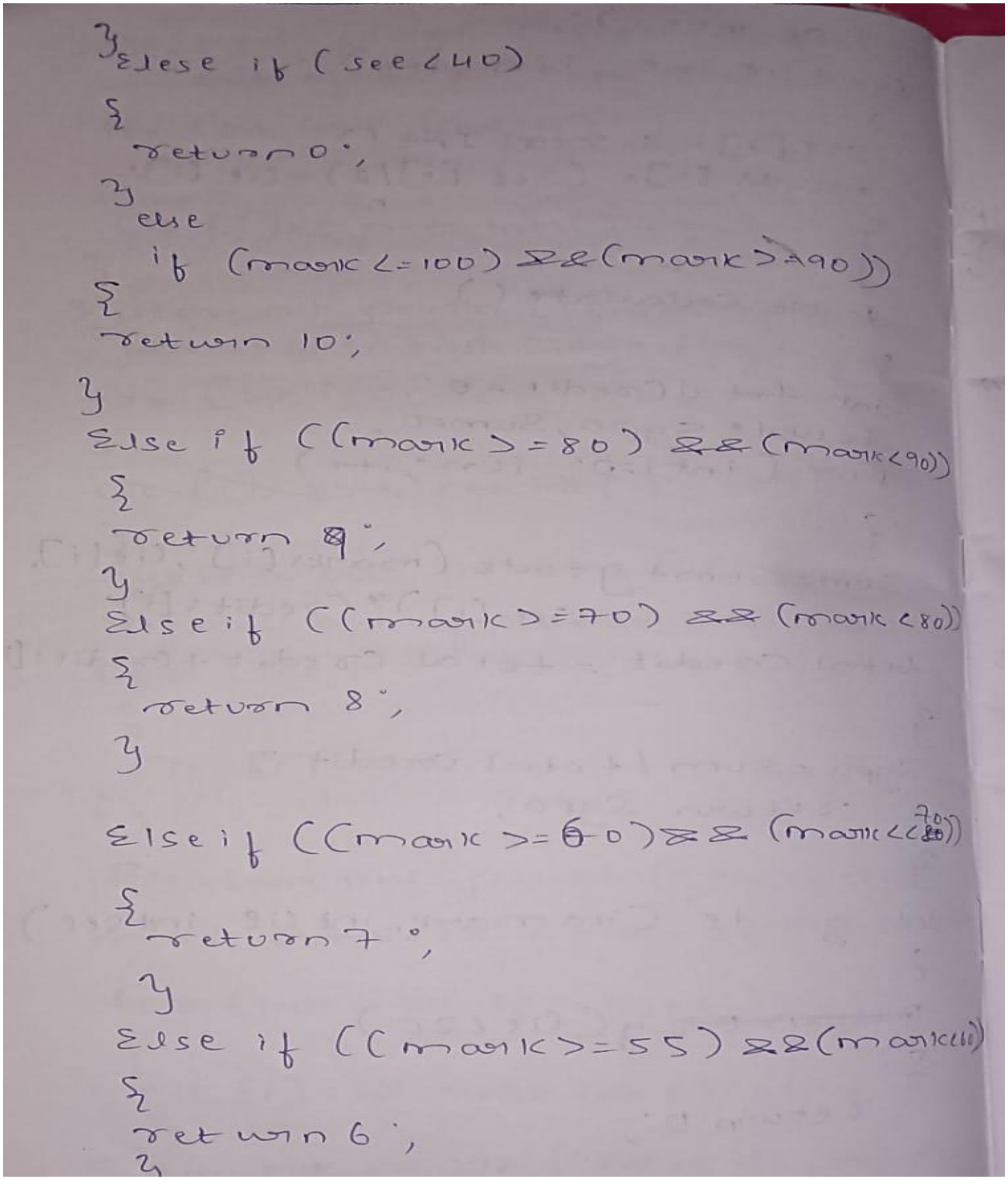
```
import java . util Scanner ;

Class Student
{
String   usn, name ;
int credits [ ]= new int [25] ;
 int marks [ ] = new int [25] ;
 int n ;
 int Cie [ ]= new int [25], See [ ]= new int
                                      [25] ;
 String ~~I~~ Subject [ ]= new String [10] ;

 Void accept ()
 {
 System. out. printIn (" Enter the usn, Name
 number of Subjects int the Semester:")

 usn = S. next () ;
 name = S. next () ;
 n=  S. next Int () ;

System. out. print In (" Enter the CIE Marks
                        ~~out of 50~~
 name of the Subjects and Credits
 respectively ") ;
   for (int i=0 ; i<n ; i++)
   { Subjects [i] = S. next () ;
     Credits [i] = S. next Int () ;
   }
```



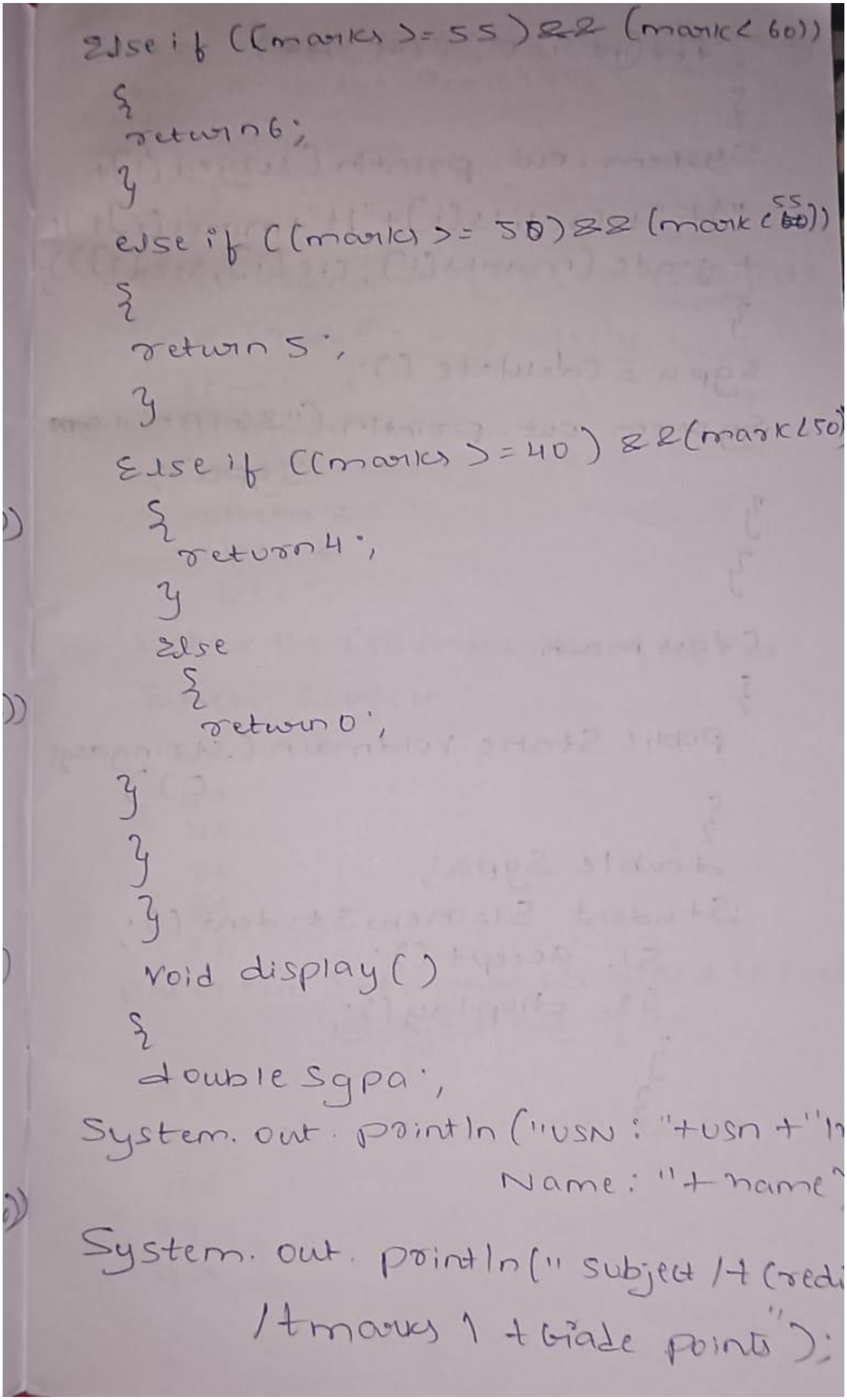
```
System.out.println("Enter the CIE Marks
      out of 50 for Each Subject :");
  for (int i=0; i<n; i++)
  {
   Cie [i]=s.nexInt();
  }
 System.out.println ("Enter the SEEMarks
   Obtained in Each Subject out of 100:");
 for (int i=0; i<n; i++)
  {
  ~~Cie [i]=s.next int ();~~  Subject [i]=s.
                              next();
  }                        Credits [i]=s.next
                              int();
 System. out. println ("Enter the CIE
   marks out of 50 for Each subject:");
 For (int i=0; i<n; i++)
   {
   Cie [i]= s.next Int ();
   }
  System.out.println(" Enter theSEE
  marks obtained in Each subject out of 100:
  for (int i=0; i<n; i++ )
   {
   Cie [i]= S.next int ();
   }
 System.out.println ("Enter the SEE marks
                obtained in Each subject out
```



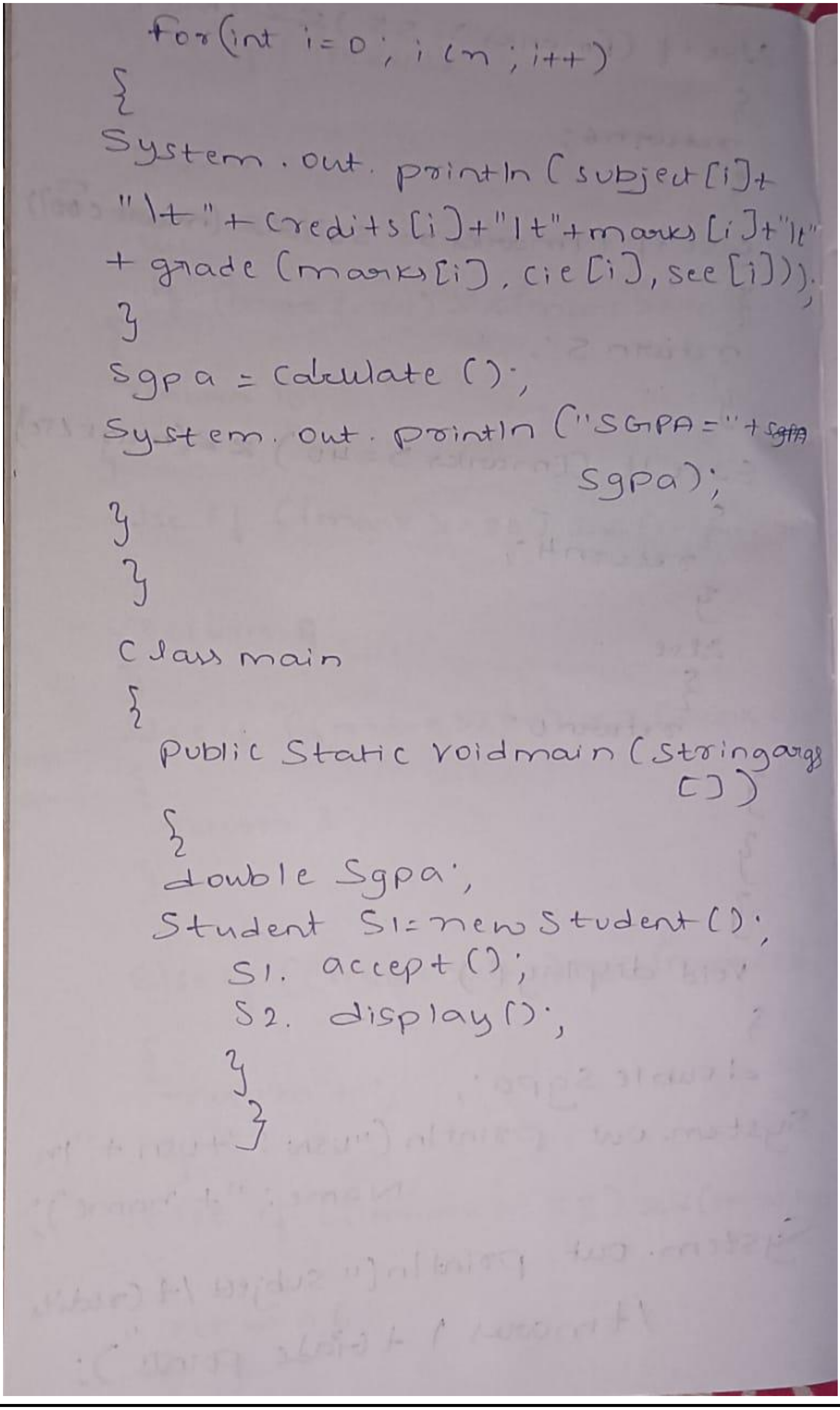
```
    of 100:");
for (int i=0; i<n; i++)
  {
   See[i]= s.nextInt();
   marks [i]= (See [i]/2)+Cie [i];
  }
  }
  double Calculate ()
  {
   int totalCredit=0;
   double sgpa, Sum=0;
   for (int i=0; i<n; i++)
   {
  Sum= Sum+grade (marks [i] ,Cie[i],
                  See [i])* Credits [i];
 total Credit = total Credit + Credits[i
 }
  Sgpa=sum/total Credit;
     return Sgpa;
 }
 int grade (int mark, int cie, int see)
 {
  ~~return 0;~~ if(Cie<20)
  {
     return 0;
```



```
}Elese if (see<40)
{
 return 0;
}
 else
  if (mark<=100) &&(mark>=90))
{
return 10;
}
Else if ((mark>=80) &&(mark<90))
 {
 return 9;
 }
 Elseif ((mark>=70) && (mark<80))
 {
  return 8;
 }

 Elseif ((mark>=60)&& (mark<70))
 {
  return 7;
  }
 Else if ((mark>=55) &&(mark<60))
 {
 return 6;
 }
```



```
Else if ((marks >= 55)&& (mark< 60))
 {
 return6;
 }
 else if ((marks >= 50)&& (mark< ~~60~~55))
 {
  return 5;
  }
 Else if ((marks >= 40) &&(mark<50)
   {
    return4;
   }
   Else
    {
     return0;
 }
 }
  }
  void display()
 {
  double sgpa;
System.out.println("USN: "+usn +"\n
                   Name: "+ name"
System.out.println(" subject /t Credi
         /tmarks \ t Grade points");
```



```
for(int i=0; i<n; i++)
{
System.out.println(subject[i]+
"\t"+credits[i]+"\t"+marks[i]+"\t"
+ grade(marks[i], cie[i], see[i]));
}
sgpa = calculate();
System.out.println("SGPA="+sgpa
                         sgpa);
}
 }
Class main
 {
  public Static voidmain(stringargs
                              [])
  {
  double Sgpa;
  Student S1=new Student();
      S1. accept();
      S2. display();
      }
       }
```

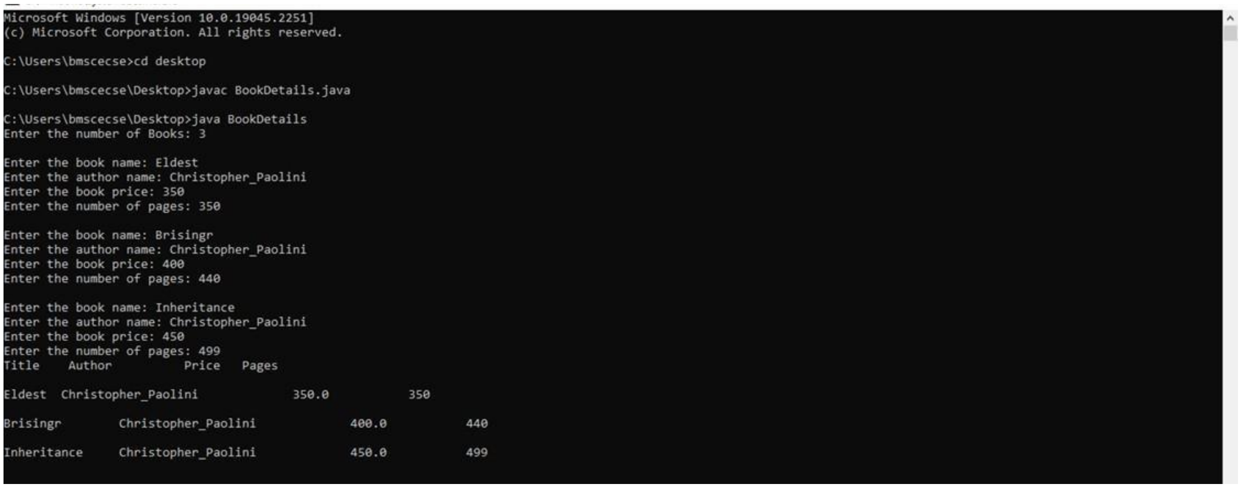
LAB PROGRAM 3: IMPLEMENTING ARRAY OF OBJECTS

CODE

```
    import java.util.*; import
    java.io.*;

    class Book
    {
    String title,author; float
    price;
    int num_pages;

    Book()
    {
    title  =  "Default  Value";
    author = "Default Value";
    price = 0.0f; num_pages =
    0;
    }
void setTitle(String title)
    {
    this.title=title;
    }

void setAuthor(String author)

    {

    this.author=author;

    }

void setPrice(float price)

    { this.price=price;

    }

void setPages(int num_pages)

    {

    this.num_pages = num_pages;

    }



public String toString()

    {

    return title+"\t\t"+author+"\t\t"+price+"\t\t"+num_pages+"\n";
    }




    }


    public class BookDetails
```

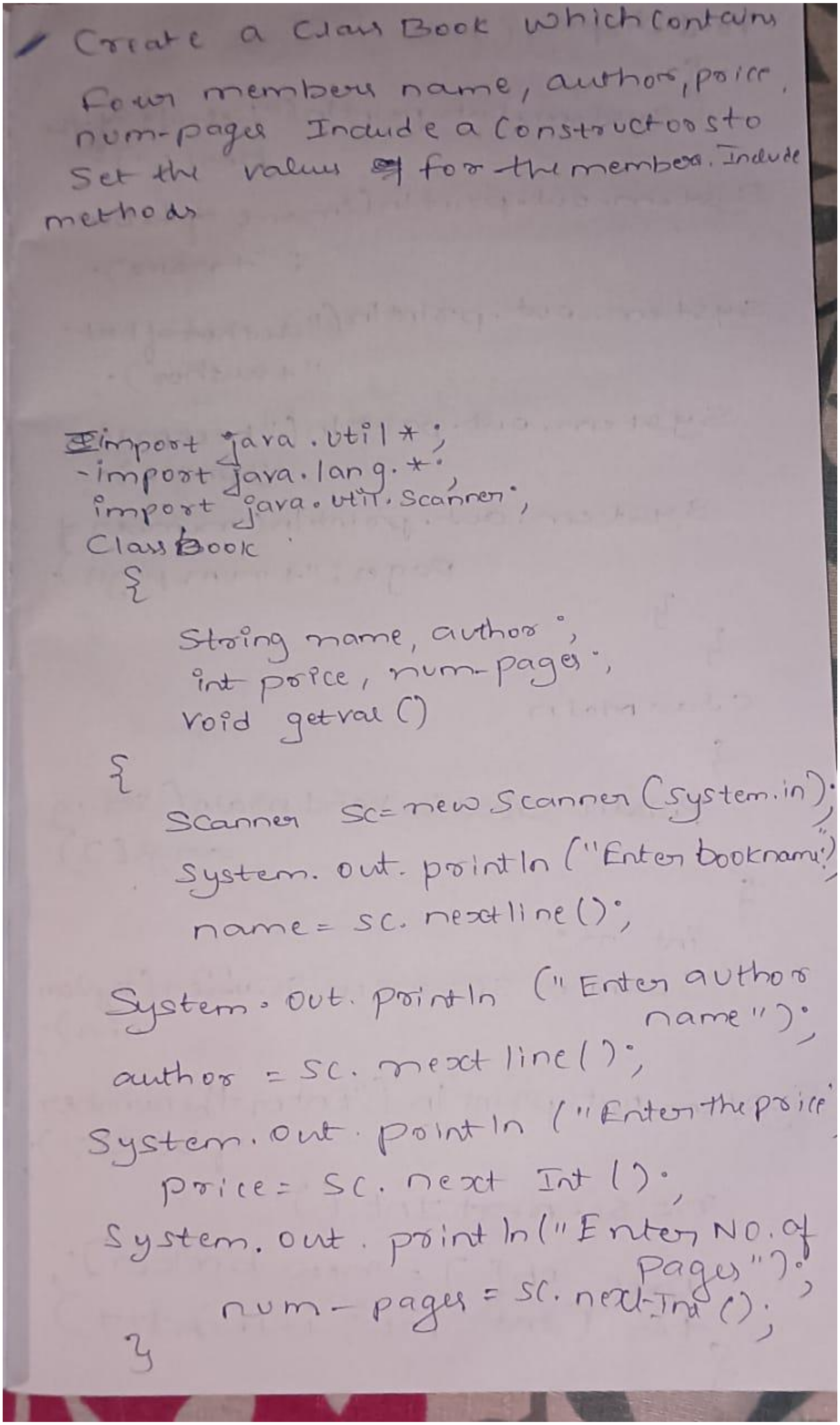
```
{

public static void main(String args[])

{


String t, a;

float p; int

np,n;

Scanner s = new Scanner(System.in);

System.out.print("Enter the number of Books: ");

n = s.nextInt();

Book[] b = new Book[n]; for(int

i=0;i<n;i++)

{

System.out.println();

System.out.print("Enter the book name: ");
t = s.next();

System.out.print("Enter the author name: "); a

= s.next();

System.out.print("Enter the book price: "); p

= s.nextFloat();

System.out.print("Enter the number of pages: "); np

= s.nextInt();
b[i] = new Book();
b[i].setTitle(t);
b[i].setAuthor(a);
b[i].setPrice(p);
b[i].setPages(np);
}
System.out.println("Title \t\t Author \t\t Price \t\t Pages\n"); for(int
i=0; i<n;i++)
{
System.out.println(b[i]);
}
}
}
```

Output:



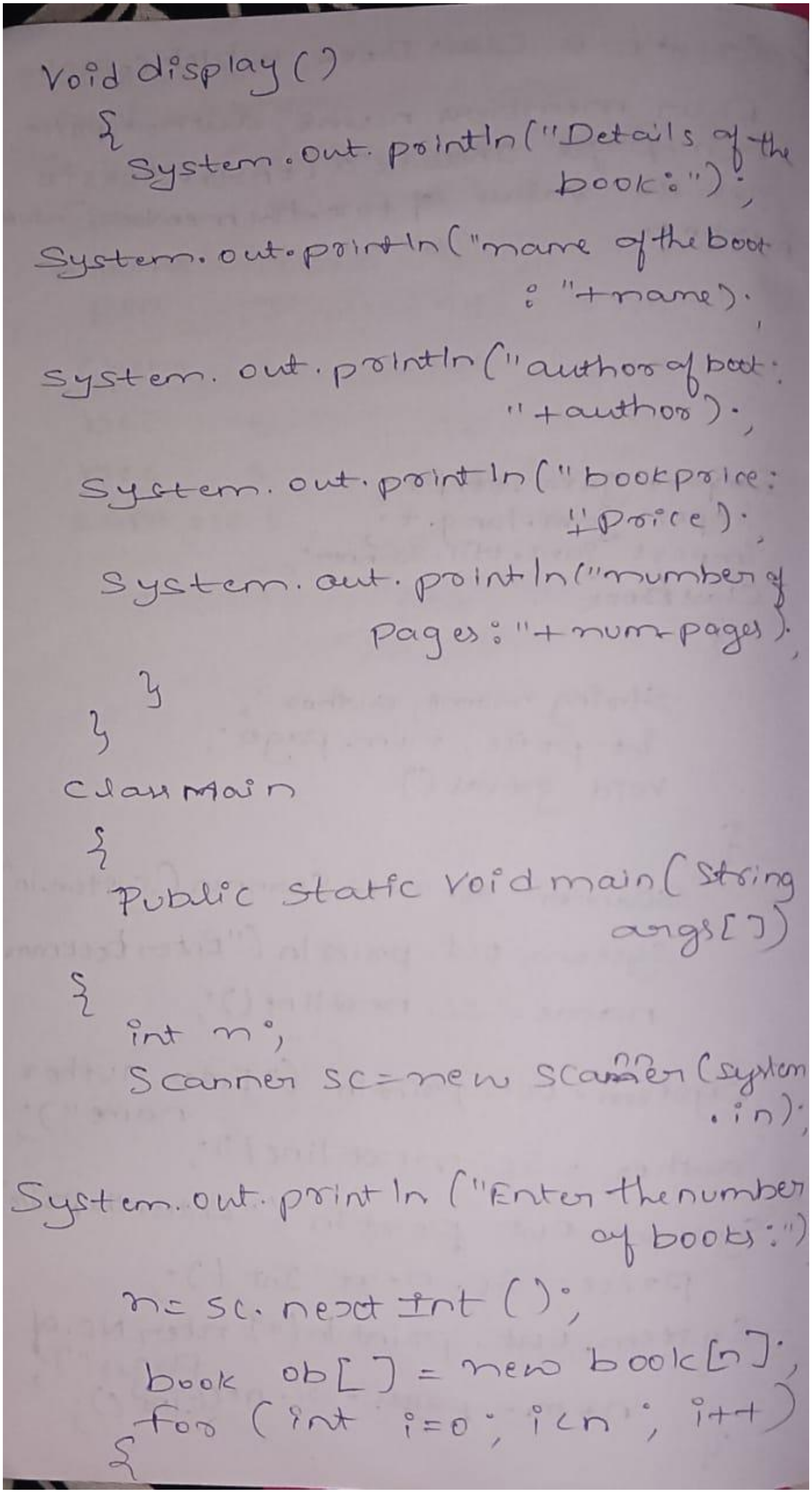
```
Microsoft Windows [Version 10.0.19045.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmscecse>cd desktop

C:\Users\bmscecse\Desktop>javac BookDetails.java

C:\Users\bmscecse\Desktop>java BookDetails
Enter the number of Books: 3

Enter the book name: Eldest
Enter the author name: Christopher_Paolini
Enter the book price: 350
Enter the number of pages: 350

Enter the book name: Brisingr
Enter the author name: Christopher_Paolini
Enter the book price: 400
Enter the number of pages: 440

Enter the book name: Inheritance
Enter the author name: Christopher_Paolini
Enter the book price: 450
Enter the number of pages: 499
Title    Author          Price   Pages

Eldest  Christopher_Paolini             350.0           350

Brisingr        Christopher_Paolini             400.0           440

Inheritance     Christopher_Paolini             450.0           499
```

Observation Book:



Create a Class Book which contains four members name, author, price, num-pages Include a Constructors to Set the values for the member. Include methods

```
import java.util*;
-import java.lang.*;
import java.util.Scanner;
Class Book
  {
     String name, author;
     int price, num-pages;
     void getval ()
  {
     Scanner Sc=new Scanner(System.in);
     System.out.println("Enter bookname");
      name= sc.nextline();

  System.out.println ("Enter author
                          name");
  author = sc.next line();
 System.out.pointln ("Enter the price");
    price= sc.next Int ();
  System.out.println("Enter No.of
                        Pages");
       num-pages = sc.nextInt();
  }
```



```
Void display()
   {
     System.out.println("Details of the
                              book:");
System.out.println("name of the book
                           : "+name);
System.out.println("author of book:
                          "+author);
  System.out.println("bookprice:
                         "+price);
   System.out.println("number of
                pages: "+num-pages);
     }
  }
 Class Main
  {
   Public static void main(String
                        args[])
 {
   int n;
   Scanner sc=new Scanner(system
                          .in);
System.out.println("Enter the number
                      of books:")
   n=sc.nextInt();
    book ob[] = new book[n];
    for (int i=0; i<n; i++)
  {
```

LAB

PROGRAM 4: CALCULATING AREA OF SHAPES (ABSTRACT CLASS)

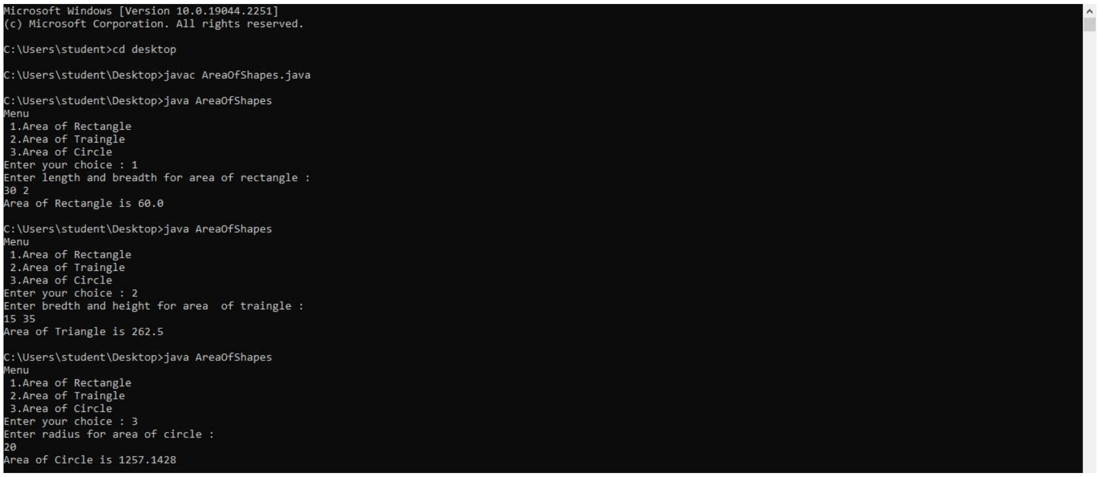
CODE

```
import java.util.Scanner; public
class Shape1
{

        public static void main(String args[])
        {        int
        choice;
        Scanner s = new Scanner(System.in); do
        {
```

```
    System.out.println("1. Calculate Area of Rectangle\n2. Calculate Area of
Triangle\n3. Calculate Area of " +
        "Circle\n4. Exit the Program\n\nEnter the choice: ");
    choice = s.nextInt(); switch(choice)
    {

        case 1: Rectangle r = new Rectangle();
        r.printArea(); break;
        case 2: Triangle t = new Triangle();
        t.printArea(); break;
        case 3: Circle c = new Circle();
        c.printArea(); break;
        case 4: System.out.println("Exiting the
        program!"); System.exit(0); break;
        default: System.out.println("\nInvalid Choice!\n");
    }

    }while(true);
    }
}


abstract class Shape
{

    int a,b;
```

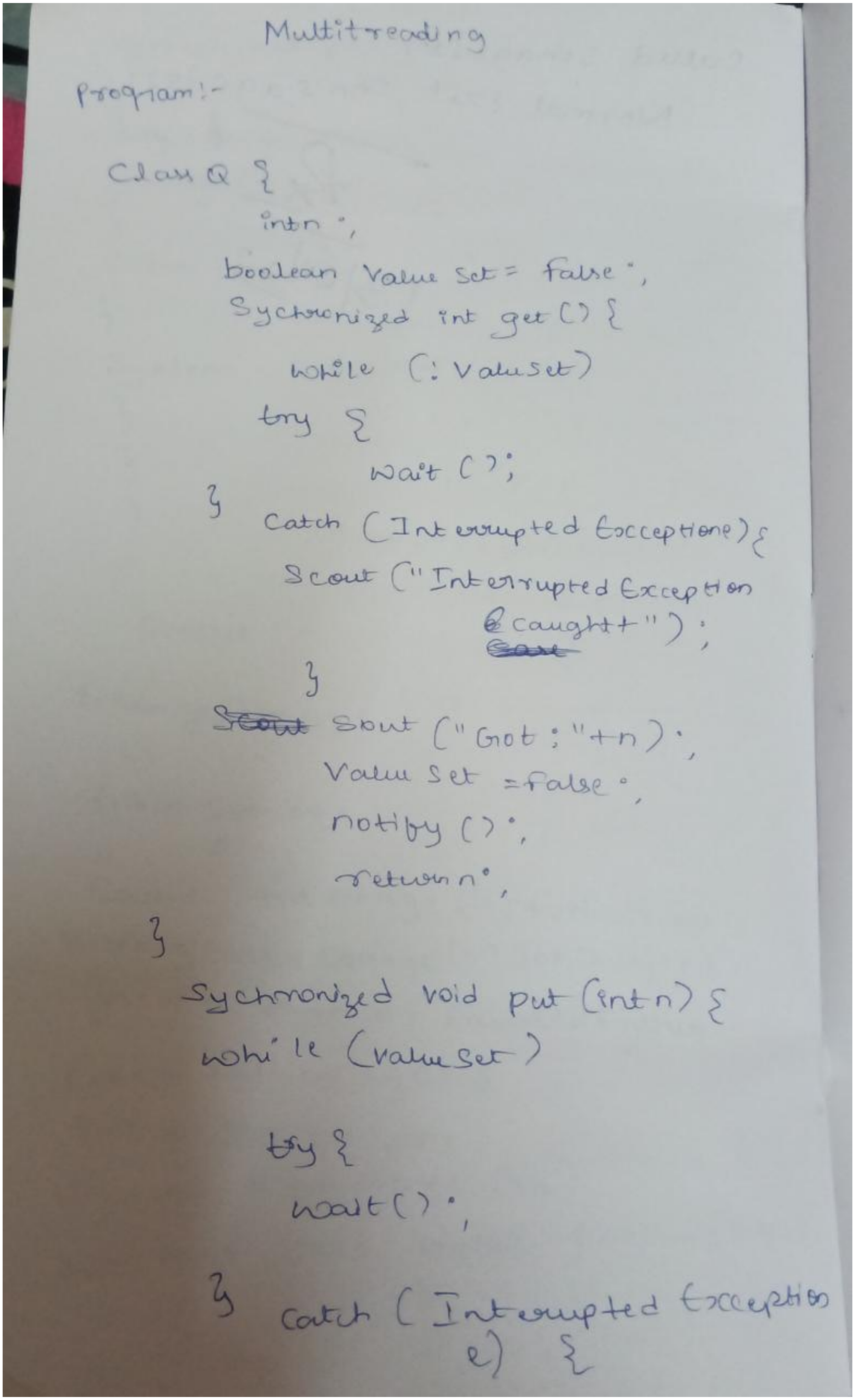
```
        abstract void printArea();
}


class Rectangle extends Shape
{
        void printArea()
        {       int
        area;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the length and breadth of rectangle: ");
        a = s.nextInt(); b = s.nextInt(); area = a*b;
        System.out.println("\nArea of Rectangle: "+area+"\n");
        }
}


class Triangle extends Shape
{

        void printArea()
        {       float
        area;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the base and height of triangle: ");
        a = s.nextInt(); b = s.nextInt(); area = 0.5f*a*b;
        System.out.println("\nArea of triangle: "+area+"\n");
        }
}


class Circle extends Shape
{

        void printArea()
        {     double
        area;
        Scanner    s    =    new    Scanner(System.in);
        System.out.println("Enter the radius of circle: ");
        a = s.nextInt(); area = Math.PI*a;
        System.out.println("Area of Circle: "+area+"\n");
        }
}
```

Output:



```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\student>cd desktop

C:\Users\student\Desktop>javac AreaOfShapes.java

C:\Users\student\Desktop>java AreaOfShapes
Menu
 1.Area of Rectangle
 2.Area of Traingle
 3.Area of Circle
Enter your choice : 1
Enter length and breadth for area of rectangle :
30 2
Area of Rectangle is 60.0

C:\Users\student\Desktop>java AreaOfShapes
Menu
 1.Area of Rectangle
 2.Area of Traingle
 3.Area of Circle
Enter your choice : 2
Enter bredth and height for area  of traingle :
15 35
Area of Triangle is 262.5

C:\Users\student\Desktop>java AreaOfShapes
Menu
 1.Area of Rectangle
 2.Area of Traingle
 3.Area of Circle
Enter your choice : 3
Enter radius for area of circle :
20
Area of Circle is 1257.1428
```

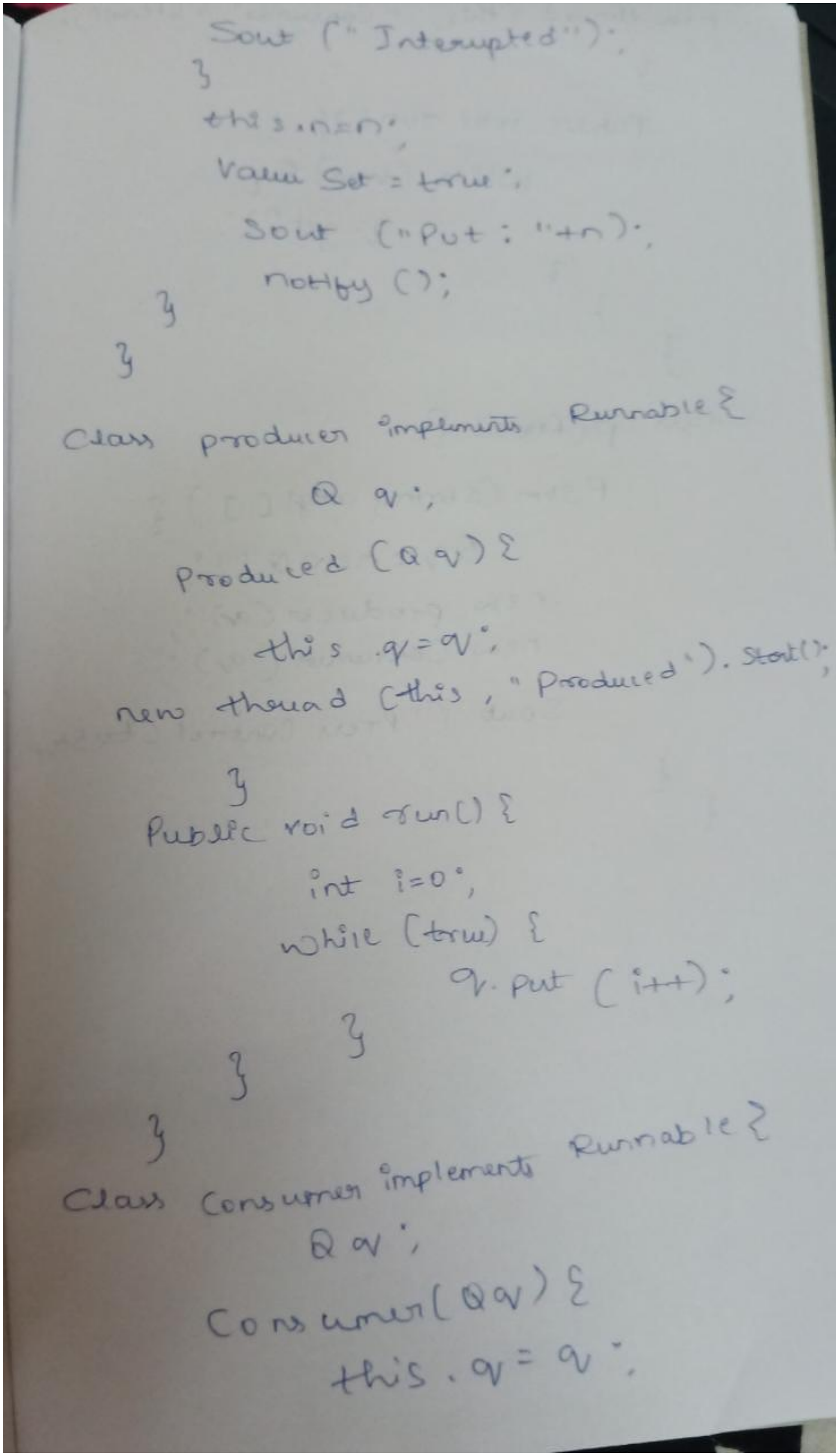
Observation Book:



Multitreading

Program:-

```
Class Q {
        intn ;
      boolean Value Set = false ;
      Sychronized int get () {
          while (! Valuset)
        try {
               wait ();
      }
        Catch (Interrupted Exceptione){
         Scout ("Interrupted Exception
                        caught+");
          }
      ~~Scout~~ Sout ("Got : "+n);
            Valu Set = false ;
            notify ();
            return n;
  }
    Sychronized void put (int n) {
     while (value Set)

        try {
          wait ();

      }  catch ( Interupted Exception
                  e)  {
```



```
        Sout ("Interupted");
      }
      this.n=n;
      Value Set = true;
       Sout ("Put : "+n);
        notify ();
    }
  }
Class producer implements Runnable{
            Q q;
      Produced (Q q){
          this.q=q;
  new thread (this, "Produced"). Start();

          }
     Public void run() {
            int i=0;
           while (true) {
                  q.put (i++);
            }
        }
    }
Class Consumer implements Runnable{
            Q q;
      Consumer(Q q){
           this.q = q;
```

```
new thread (this, "Consumer"). start();
    }
    Public void run(){
            while (true) ~~Q~~ {
                    q.get();
            }
        }
    }
Class pcfixed {
        Psvm (String args[ ]) {
                Qq = new Q();
            new producer (q);
            new Consumer (q);
            Sout ("Press Control C to Stop");

    {
 {
```

LAB PROGRAM 5: BANK PROGRAM

CODE

```
import java.util.Scanner; class
Account
{
    String
    customer_name;   long
    acc_no; float bal;
    Scanner s = new Scanner(System.in); public
    void input()
    {
        System.out.print("\nEnter the Customer Name: ");

        customer_name = s.nextLine();

        System.out.print("\nEnter the Account Number: ");

        acc_no = s.nextLong();

        System.out.print("\nEnter the Starting Amount (Minimum Amount = 5000):
");
        bal = s.nextFloat(); if(bal<5000f)
        {
            System.out.println("\nAccount Balance cannot be less than 5000.0 \n");
            System.exit(0);
        }
    }
    public void display()
    {
        System.out.println("\nCustomer Name: "+customer_name);
        System.out.println("Account Number: "+acc_no);
        System.out.println("Amount: "+bal);
    }
}


class Savings extends Account
{
    Scanner s = new Scanner(System.in);
    float          deposit,withdraw,interest;
    public void deposit()
    {
        System.out.print("\nEnter the amount to be deposited: ");
        deposit = s.nextFloat(); bal+=deposit;
        System.out.println("\nBalance: "+bal);
    }        public        void
    withdraw()
    {
        System.out.print("\nEnter the amount to be withdrawn: ");
        withdraw = s.nextFloat(); if(bal<5000)
        {
            System.out.println("\nInsufficient Balance");
        }
        else

        {
bal-=withdraw;

          System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance:
    "+bal);

        }


      }


      public void check_Bal()

      { if(bal<5000)

        {

          System.out.println("\nInsufficient Balance!!\nBalance: "+bal);

        } else

        {

          System.out.println("\nBalance: "+bal);

      } } public void interest()

      { interest=(bal*6)/100;

        bal+=interest;

        System.out.println("\nInterest Credited: "+interest+"\nBalance :"+bal) ;

      }
    }


class Current extends Account

    { float deposit, withdraw, penalty;



      public void deposit()
    {
       System.out.print("\nEnter Amount to be deposited: ");
       deposit = s.nextFloat(); bal += deposit;
       System.out.println("Balance: " + bal);
    }


    public void check_Bal()
    { if (bal < 5000)
       { penalty = (0.1f * bal);
          System.out.println("\nInitial Account Balance: "+bal); bal
          = bal-penalty;
          System.out.println("\nLow balance!\nPenalty Amount: " + penalty +
"\nAccount balance: " + bal);
       }
       else
       {
          System.out.println("\n Balance: " + bal); }
    }


    public boolean check_Bal_part_2()
    { if (bal < 5000)
       { penalty = (0.1f * bal);
          System.out.println("\nInitial Account Balance: "+bal); bal
          = bal-penalty;
          System.out.println("\nLow Balance!\nPenalty Amount: " + penalty +
"\nAccount balance: " + bal); return false;
```

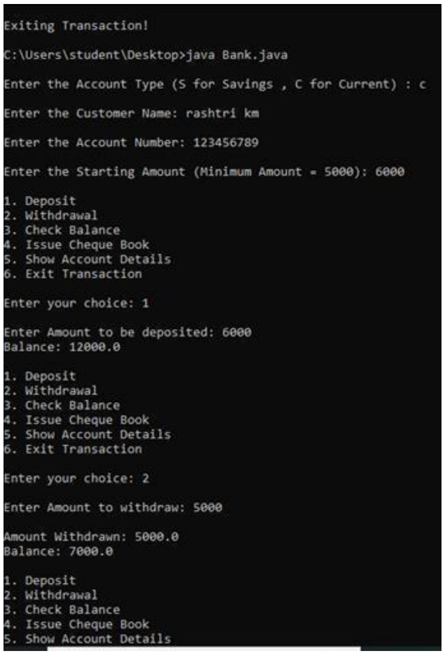
```
        }      return
        true;
    }


    public void withdraw()
    {
        System.out.print("\nEnter Amount to withdraw: ");
        withdraw = s.nextFloat(); if(check_Bal_part_2())
        { bal-=withdraw;
            System.out.pri
            ntln("\nAmoun
            t Withdrawn:
            "+withdraw+"\
            nBalance:
            "+bal);

        }
    }

    public void chequebook()
    {
        System.out.println("\nCheque Book has been Issued!"); }
}



public class Bank
```

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

        Scanner s = new Scanner(System.in);

        String ch;

        int n;

        Current c = new Current();

        Savings sa = new Savings();

        System.out.print("\nEnter the Account Type (S for Savings , C for Current) :
"); ch = s.next();


        switch(ch.toLowerCase())
        {
            case "s" : sa.input(); do
                        {
                            System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check
Balance \n4. Check Interest"
                                    +"\n5. Show Account Details \n6. Exit Transaction\n\nEnter
your choice: ");  n = s.nextInt();
                            switch(n)
                            { case 1 : sa.deposit();
                                break;
                                case 2 : sa.withdraw(); break;
                                case 3 : sa.check_Bal(); break;
                                case 4 : sa.interest(); break;
                                case 5 : sa.display(); break;
```

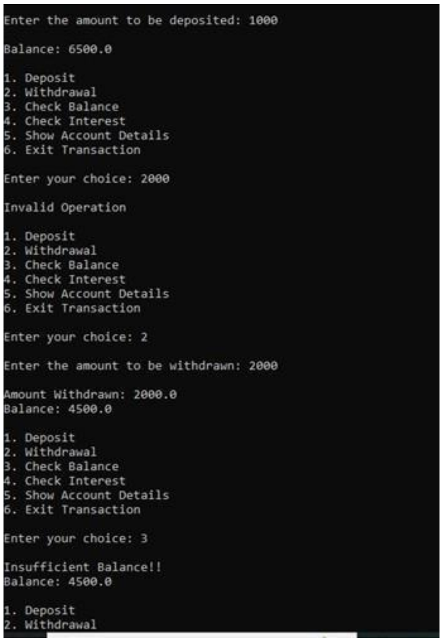
```
                    case 6 : System.out.println("\nExiting
                            Transaction!"); System.exit(0); break;
                    default : System.out.println("\nInvalid Operation");
                  }
              }while(true)
        ; case "c" : c.input();
        do {
                  System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check
Balance \n4. Issue Cheque Book"
                          + "\n5. Show Account Details \n6. Exit Transaction\n\nEnter
your choice: ");  n = s.nextInt();
                  switch   (n)   {
                  case 1:
                          c.deposit(); break;
                    case 2:
                          c.withdraw(); break;
                    case 3:
                          c.check_Bal(); break;
                    case 4:
                          c.chequebook(); break;
                    case 5:
                          c.display(); break;
                    case 6:
                          System.out.println("\nExiting Transaction!");
              System.exit(0); break;
                    default:
                          System.out.println("\nInvalid Operation"); }
```

```
                }while(true); default :
            System.out.println("\nInvalid Choice"); break;
            }
        }
    }
```

Output:

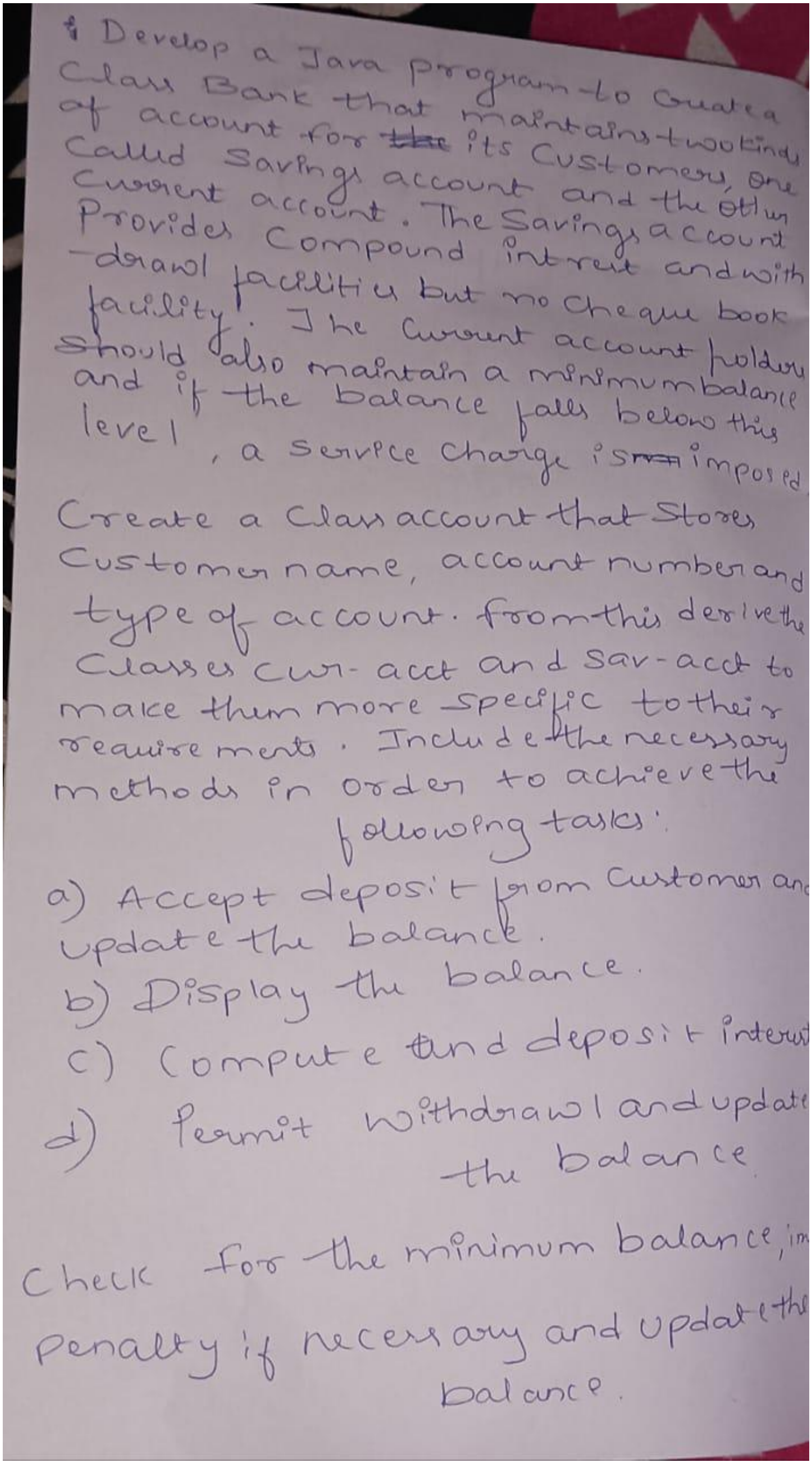


```
Exiting Transaction!

C:\Users\student\Desktop>java Bank.java

Enter the Account Type (S for Savings , C for Current) : c

Enter the Customer Name: rashtri km

Enter the Account Number: 123456789

Enter the Starting Amount (Minimum Amount = 5000): 6000

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 1

Enter Amount to be deposited: 6000
Balance: 12000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
6. Exit Transaction

Enter your choice: 2

Enter Amount to withdraw: 5000

Amount Withdrawn: 5000.0
Balance: 7000.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Issue Cheque Book
5. Show Account Details
```



```
Enter the amount to be deposited: 1000

Balance: 6500.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 2000

Invalid Operation

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 2

Enter the amount to be withdrawn: 2000

Amount Withdrawn: 2000.0
Balance: 4500.0

1. Deposit
2. Withdrawal
3. Check Balance
4. Check Interest
5. Show Account Details
6. Exit Transaction

Enter your choice: 3

Insufficient Balance!!
Balance: 4500.0

1. Deposit
2. Withdrawal
```

Observation Book:



i Develop a Java program to Create a Class Bank that maintains two kinds of account for ~~the~~ its Customers, one Called Savings account and the other Current account. The Savings account Provides Compound intrest and with-drawl facilities but no Cheque book facility. The Current account holders Should also maintain a minimum balance and if the balance falls below this level, a service Charge is imposed.

Create a Class account that Stores Customer name, account number and type of account. from this derive the Classes cur-acct and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

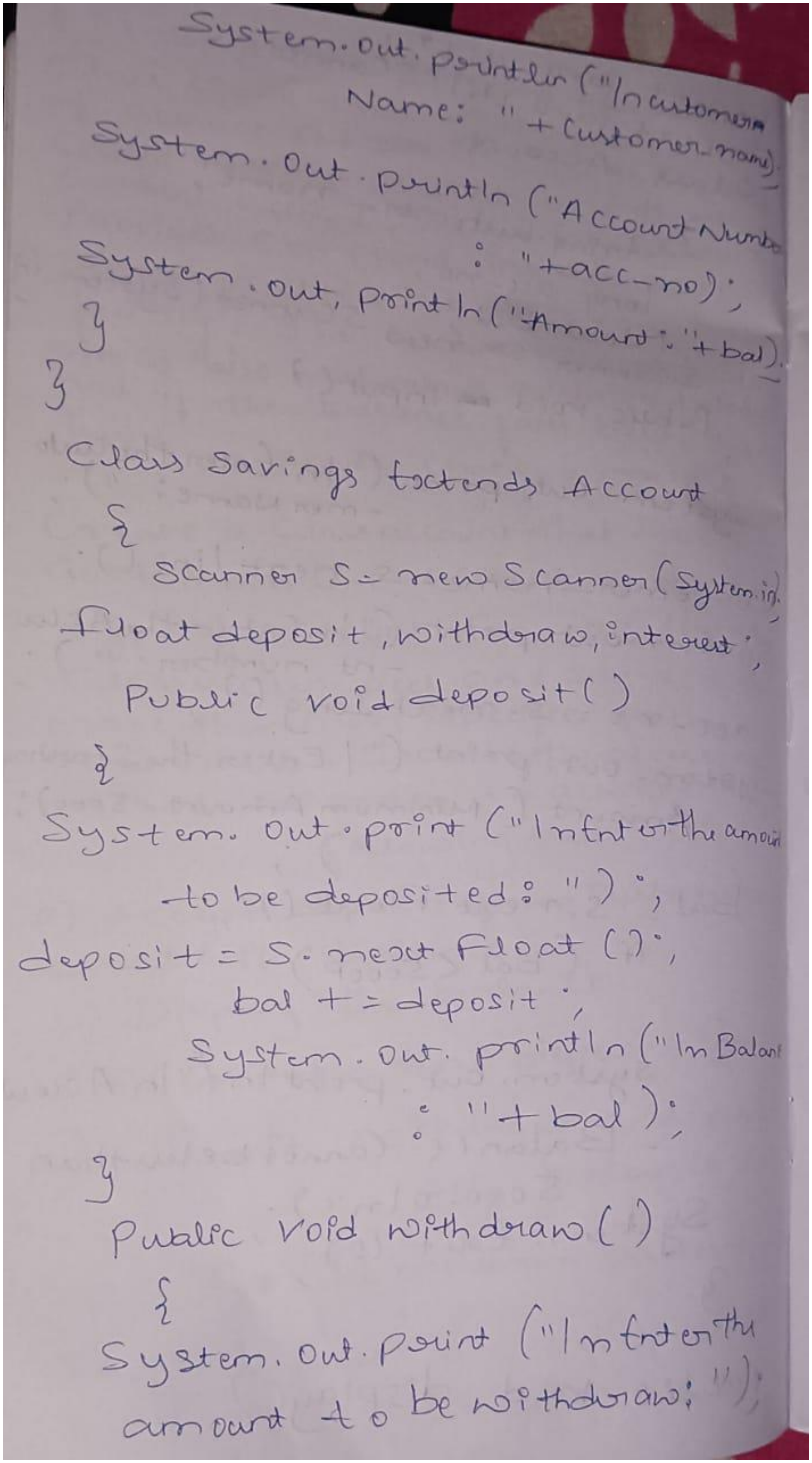
a) Accept deposit from Customer and update the balance.

b) Display the balance.

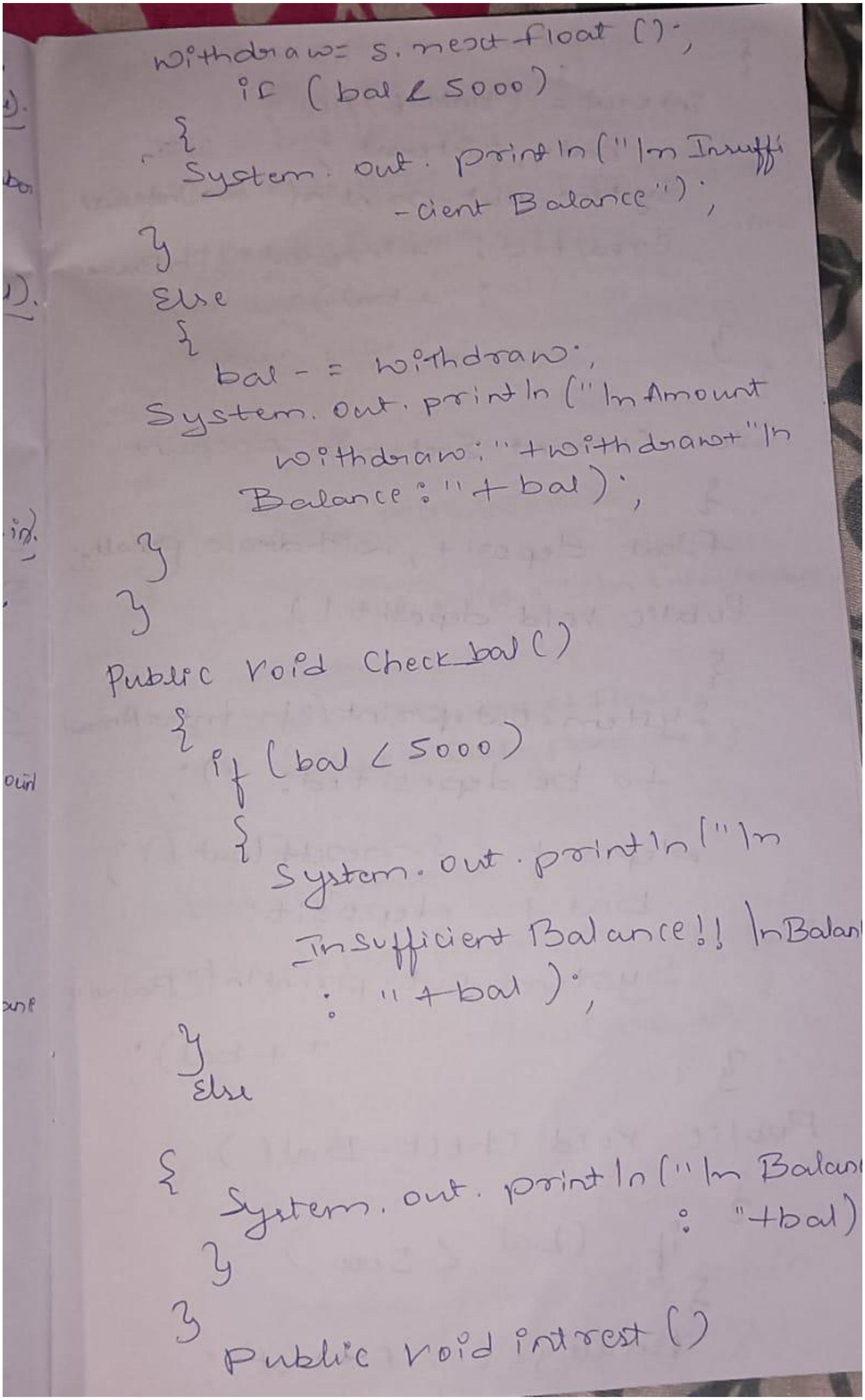
c) Compute and deposit interest

d) Permit withdrawl and update the balance

Check for the minimum balance, im penalty if necessary and update the balance.



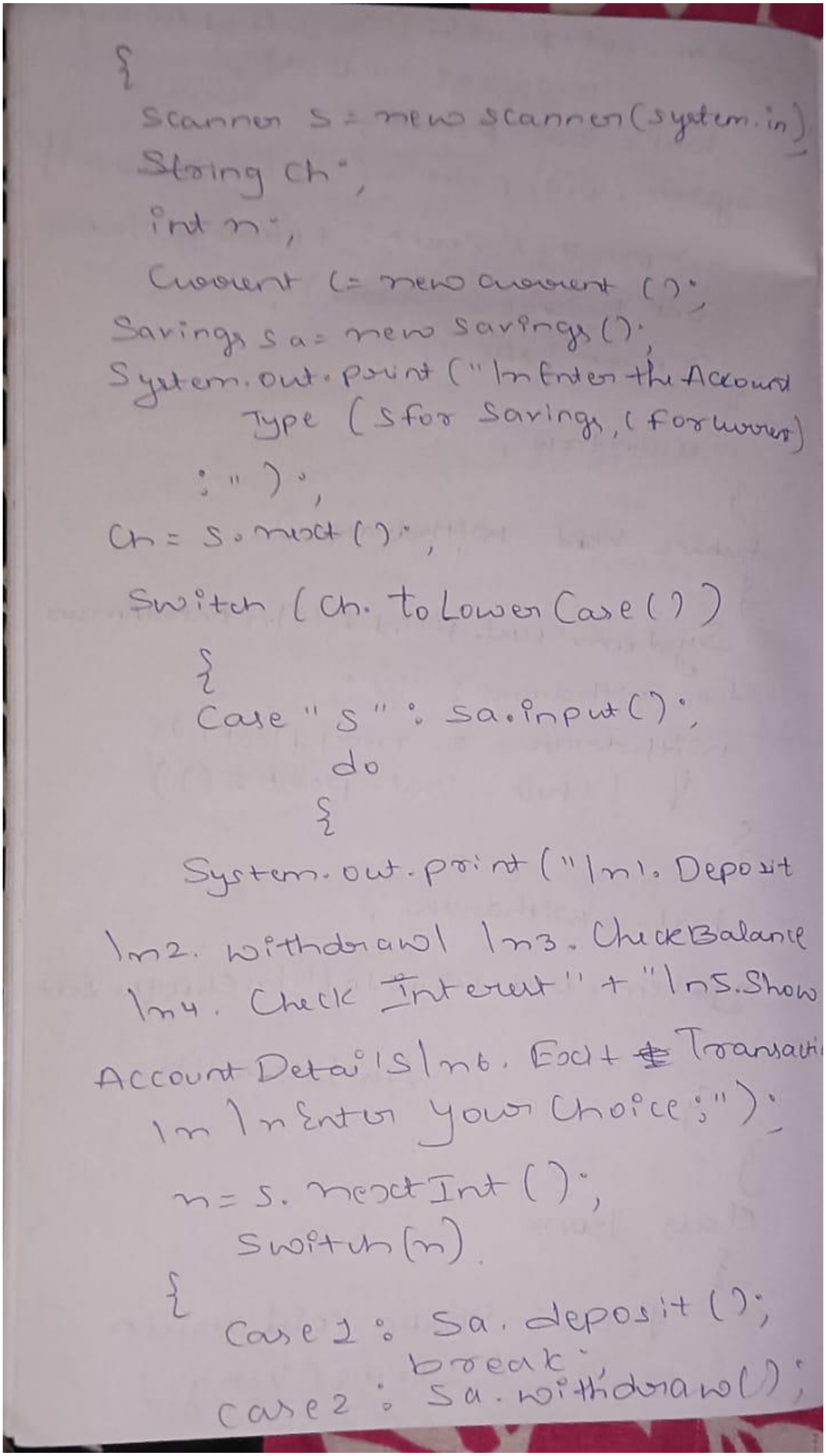
```
System.out.println("\n customer
            Name: " + Customer_name);
System.out.println("Account Number
                : "+acc_no);
System.out.println("Amount: "+bal);
}
}
Class Savings Extends Account
{
   Scanner S= new Scanner(System.in);
float deposit, withdraw, interest;
   Public void deposit()
{
System.out.print("\n Enter the amount
      to be deposited: ");
deposit= S.nextFloat();
        bal += deposit;
       System.out.println("\n Balance
                  : "+bal);
}
 Public void withdraw()
  {
System.out.print("\n Enter the
amount to be withdraw: ");
```



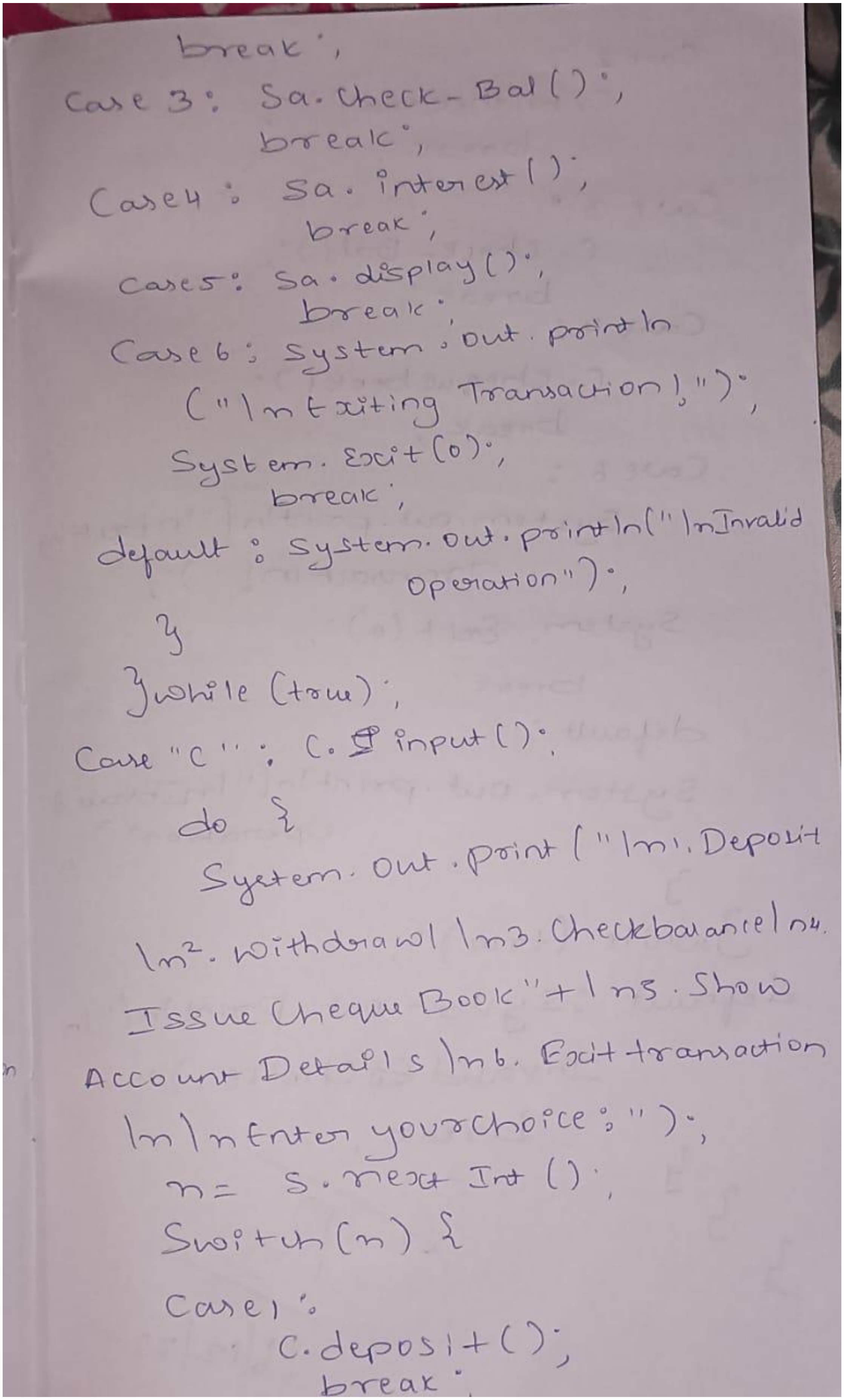
```
withdraw= s.next float ();
    if (bal < 5000)
  {
  System.out.println("\n Insuffi
             -cient Balance");
  }
  Else
  {
    bal -= withdraw;
  System.out.println("\n Amount
          withdraw:"+withdraw+"\n
        Balance:"+bal);
  }
 }
Public void Check_bal()
   {
    if (bal < 5000)
      {
       System.out.println("\n
        Insufficient Balance!! \nBalan
          : "+bal);
   }
    Else
   {
    System.out.println("\n Balan
                        : "+bal)
    }
   }
    Public void intrest ()
```

```
System. out . println ("\n Initial Account
            Balance: "+ bal);
   bal = bal - penalty;
 System. out. println ("\n Low Balance !|
     n penalty Amount : "+ penalty+ "\n Acco
   -unt balance : " +bal);
   return false ;
  }
  return true;
  }
 Public void   withdraw( )
  {
   System. out. printf ("\n EnterAmount
 - to withdraw : ");
   withdraw = s. nextFloat();
     if ( Check - Bal - part 2())
  {
    bal - = withdraw ;
  System. out. println ("\n Checque Book
                        has issued!");
     }
  }
  class  Bank
   {
        Public Static void main
                ( String [] args)
```

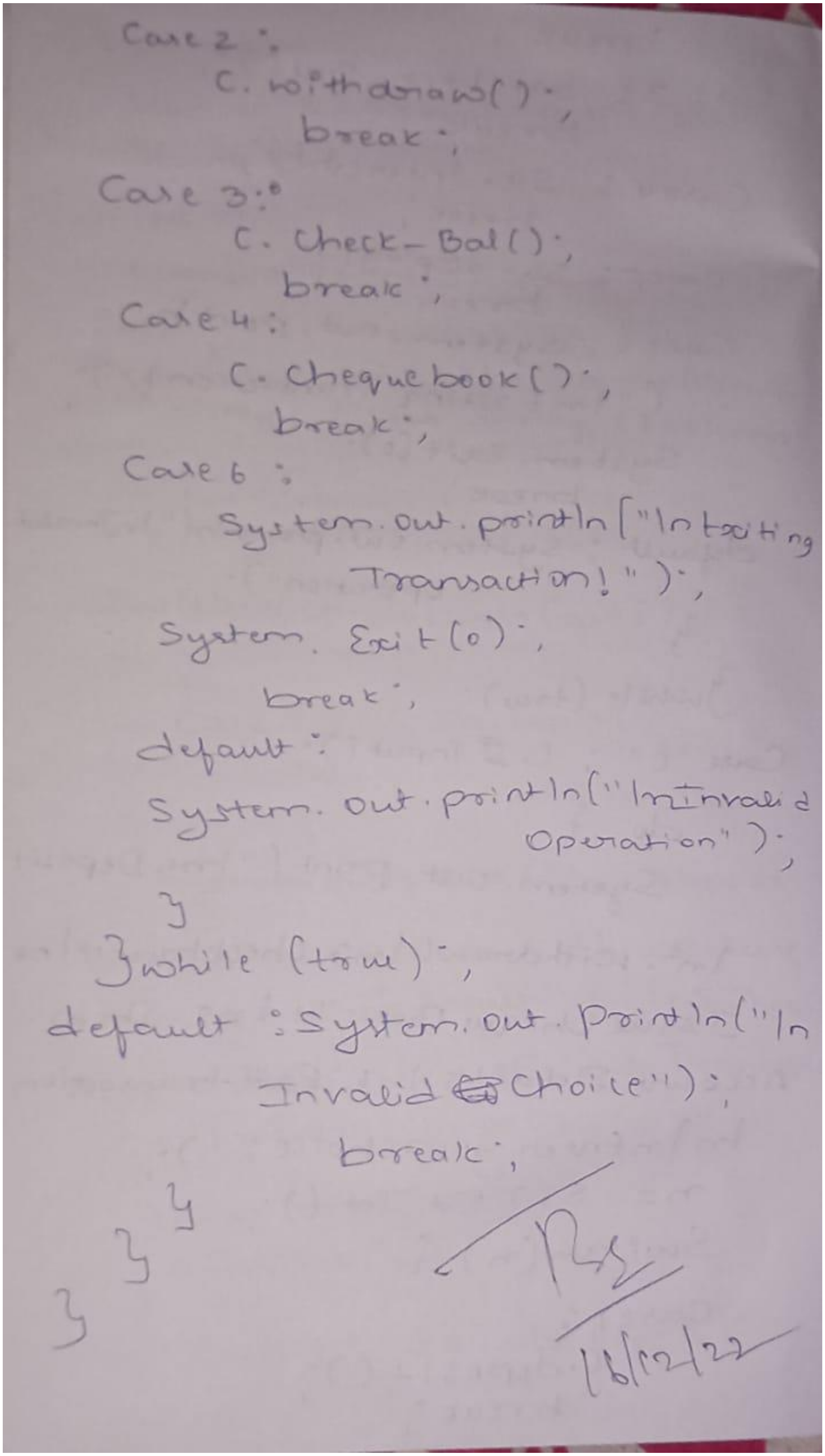
```
{
  interest = (bal* 6)/100 ;
      bal+ =interest ;
    System.out.println("\nInterest
     Credited: "+interest+"\nBalance
                : "+bal);
 }
}
Class Current Extends Account
   {
   float deposit, withdraw, penalty;
  Public void deposit ()
    {
     System.out.print("\n Enter Amount
        to be deposited: ");
      deposit = S.nextFloat();
        bal += deposit;
       System.out.println("Balance
                         " + bal);
    }
Public void Check-Bal()
    {   if  (bal < 5000)
       {
         Penalty= (0.1f*bal);
```



```
{
 Scanner s = new scanner(system.in)
 String ch;
 int n;
 Current c= new current ();
Savings sa= new savings();
System.out.print ("\nEnter the Account
      Type (s for Savings, c for current)
   :");
Ch= s.next();
 Switch (ch.toLowerCase())
    {
    Case "s": sa.input();
          do
          {
    System.out.print("\n1. Deposit
\n2. withdrawl \n3. CheckBalance
 \n4. Check Interest" + "\n5.Show
Account Details\n6. Exit Transacti
 \n \nEnter your Choice:");
   n= s.nextInt();
      switch(n)
  {
      Case 1: sa.deposit();
              break;
    case2: sa.withdrawl();
```



```
        break;
Case 3:  Sa.check-Bal();
         break;
 Case4 :  Sa.interest();
           break;
  Case5:  Sa.display();
           break;
  Case 6: System.out.println
     ("\nExiting Transaction!");
     System.Exit(0);
        break;
 default : System.out.println("\nInvalid
                 Operation");
   }
  }while (true);
Case "C" : C.input();

     do {
      Syetem.out.print("\n1.Deposit
   \n2.withdrawl\n3.Checkbalance\n4.
   Issue Cheque Book"+\n5.Show
 Account Details\n6.Exit transaction
   \n\nEnter yourchoice:");
     n= S.next Int();
     Switch(n) {

     Case1:
            C.deposit();
              break;
```



```
Case 2 :
      C. withdraw();
          break;
Case 3:
      C. Check-Bal();
          break;
Case 4 :
      C. Chequebook();
          break;
Case 6 :
      System.out.println("\nExiting
              Transaction!");
   System. Exit(0);
          break;
 default:
 System.out.println("\nInvalid
                  Operation");
   }
 }while (true);
default :System.out.println("\n
        Invalid Choice");
           break;
     }
   }
}
```

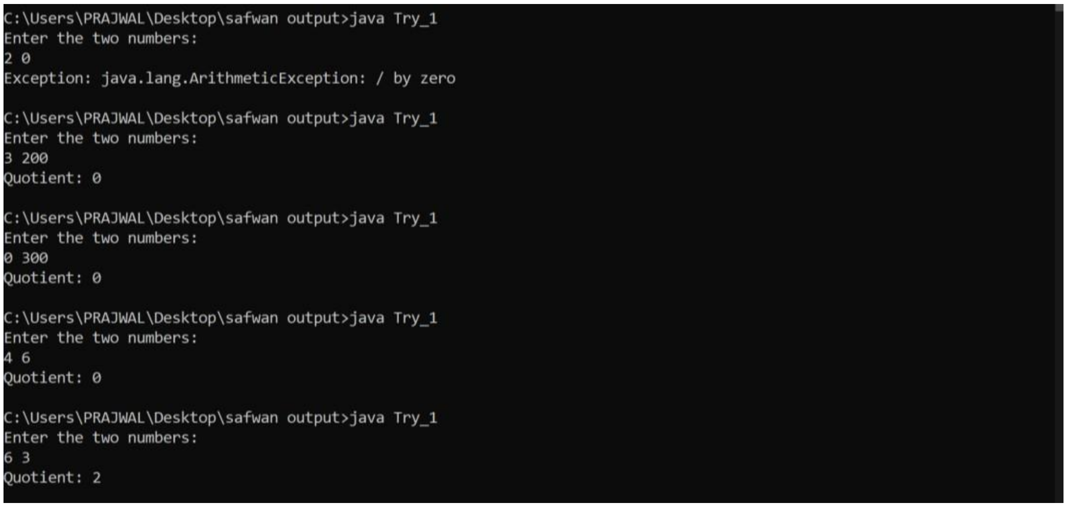
16/12/22

LAB PROGRAM 6: NUMBER OPERATIONS - EXCEPTION HANDLING

CODE

```
    import java.util.InputMismatchException; import
    java.util.Scanner;
interface Z
    { public int calc(int a,int b);
    }


    class Y implements Z
    { public int calc(int a, int b)
      { int c = a/b;
        return c;
      }
    }


    public class Try_1
    { public static void main(String[] args)
      {
        Scanner s = new Scanner(System.in);
        Y o = new Y(); int
        num1,num2;
        try
        {
           System.out.println("Enter the two numbers: ");
           num1 = s.nextInt(); num2 = s.nextInt(); int c =
           o.calc(num1,num2);
           System.out.println("Quotient: "+c);
        }  catch(ArithmeticException  |  InputMismatchException
        e1)
        {
           System.out.println("Exception: "+e1);
        }
     }
  }
```

Output:



```
C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
2 0
Exception: java.lang.ArithmeticException: / by zero

C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
3 200
Quotient: 0

C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
0 300
Quotient: 0

C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
4 6
Quotient: 0

C:\Users\PRAJWAL\Desktop\safwan output>java Try_1
Enter the two numbers:
6 3
Quotient: 2
```

Observation Book:

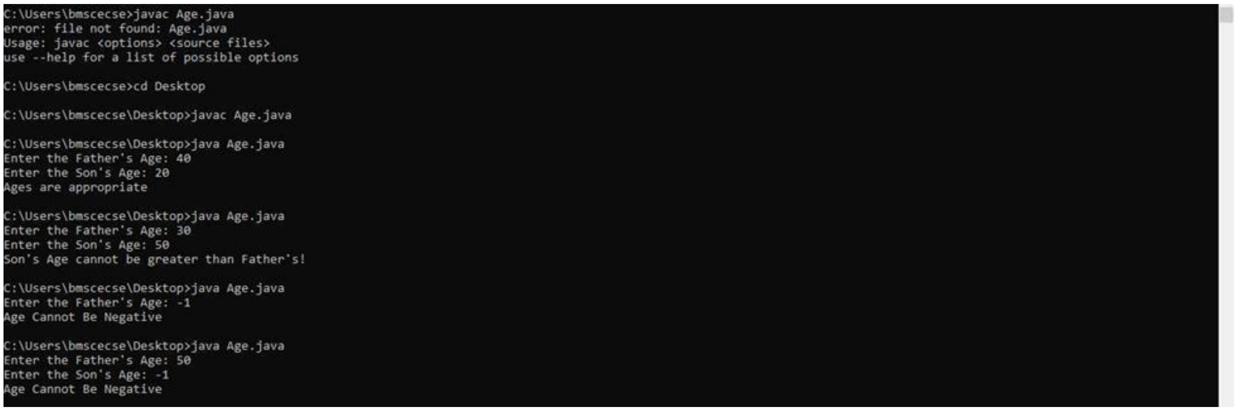
LAB PROGRAM 7: AGE EVALUATION - EXCEPTION HANDLING

CODE

```
    import java.util.Scanner;

public class Age

    {

          public static void main(String[] args) throws WrongAge,InvalidAge

          {       new

          Son();

          }

    }



    class WrongAge extends Exception

    {

          public String getMessage()

          {

          return "Age Cannot Be Negative";
          }
}



class InvalidAge extends Exception
{ public String getMessage()

          {

          return "Son's Age cannot be greater than Father's!"; }

}

class Father

{


          Scanner s = new Scanner(System.in);

          int f;

          Father() throws WrongAge

          {

          System.out.print("Enter the Father's Age:

          "); f = s.nextInt(); try { if(f<0) throw new

          WrongAge();

          }    catch(WrongAge

          e1)

          {

            System.out.println(e1.getMessage());

          System.exit(0);
```

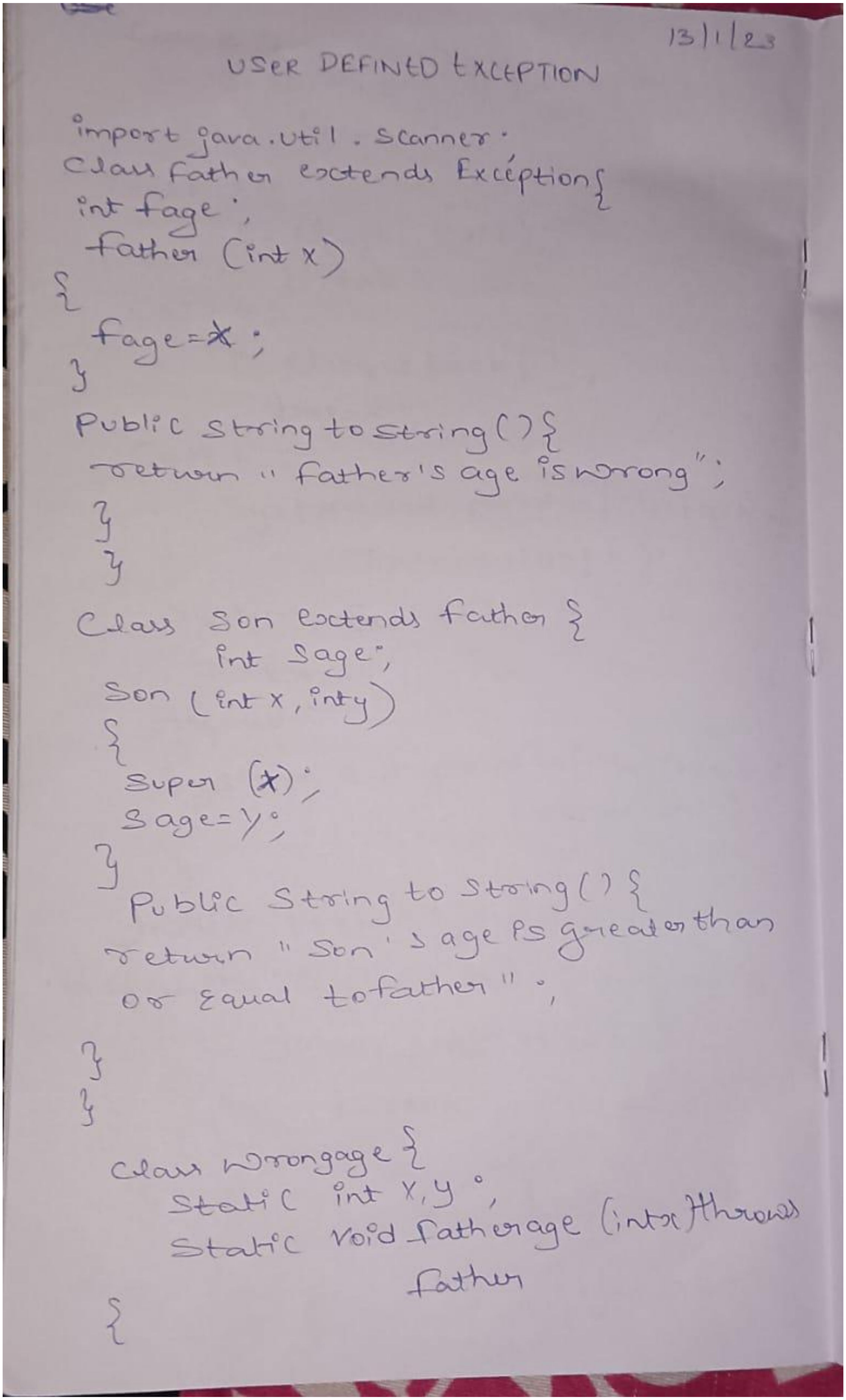
```
        }
        }
}


class Son extends Father
{

        int son;
        Son() throws WrongAge,InvalidAge
        { super();
        System.out.print("Enter the Son's Age:
        "); son = s.nextInt(); try { if(son<0)
        throw new WrongAge();
        }    catch(WrongAge
        e2)
        {
          System.out.println(e2.getMessage());
            System.exit(0);
        } try { if(son>f) throw
        new InvalidAge();
        }    catch(InvalidAge
        e3)
        {
              System.out.println(e3.getMessage());
                System.exit(0);
            }
            System.out.println("Ages are appropriate");
            }
    }
```

Output:



```
C:\Users\bmscecse>javac Age.java
error: file not found: Age.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\bmscecse>cd Desktop

C:\Users\bmscecse\Desktop>javac Age.java

C:\Users\bmscecse\Desktop>java Age.java
Enter the Father's Age: 40
Enter the Son's Age: 20
Ages are appropriate

C:\Users\bmscecse\Desktop>java Age.java
Enter the Father's Age: 30
Enter the Son's Age: 50
Son's Age cannot be greater than Father's!

C:\Users\bmscecse\Desktop>java Age.java
Enter the Father's Age: -1
Age Cannot Be Negative

C:\Users\bmscecse\Desktop>java Age.java
Enter the Father's Age: 50
Enter the Son's Age: -1
Age Cannot Be Negative
```

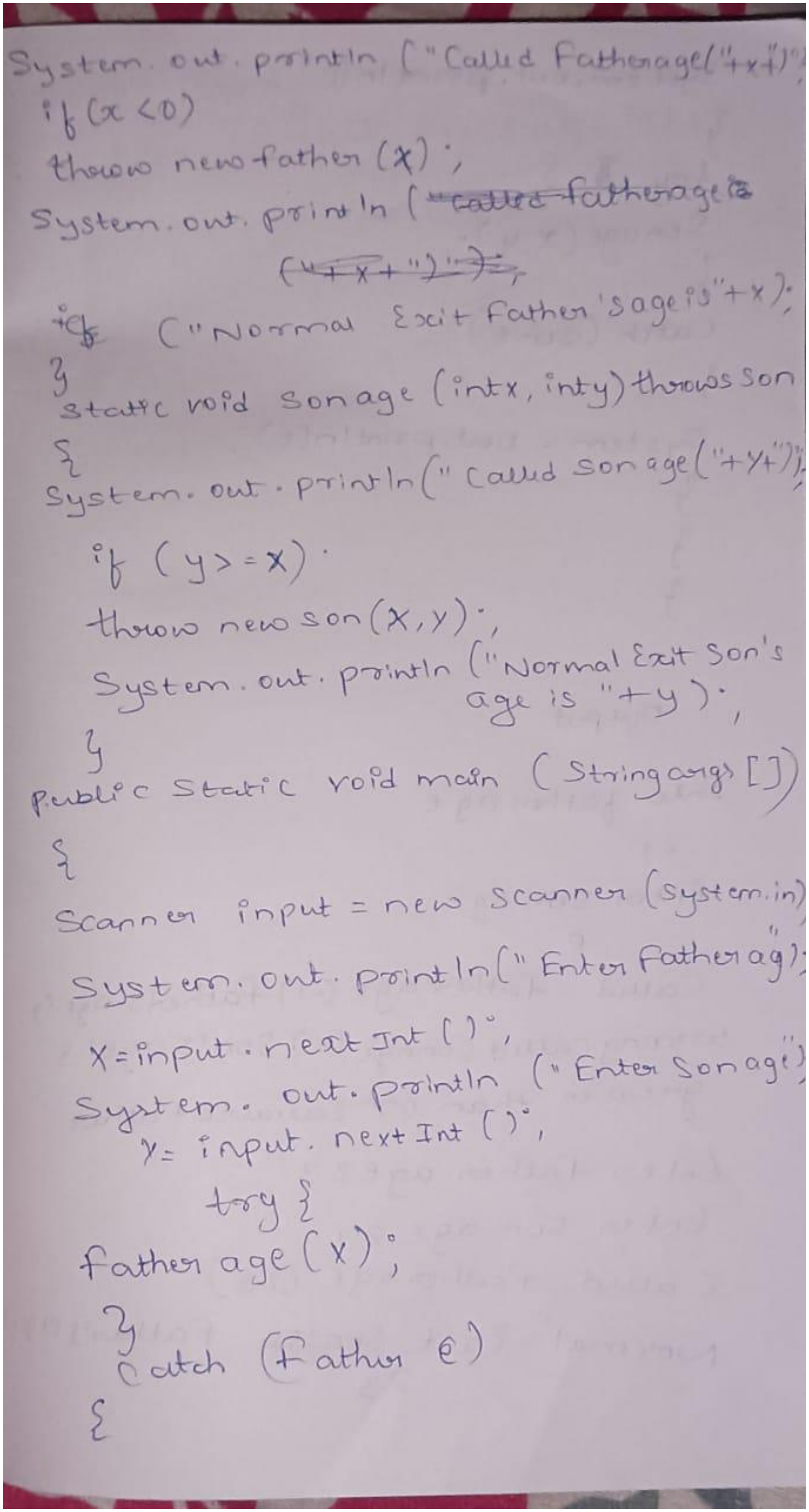
Observation Book:



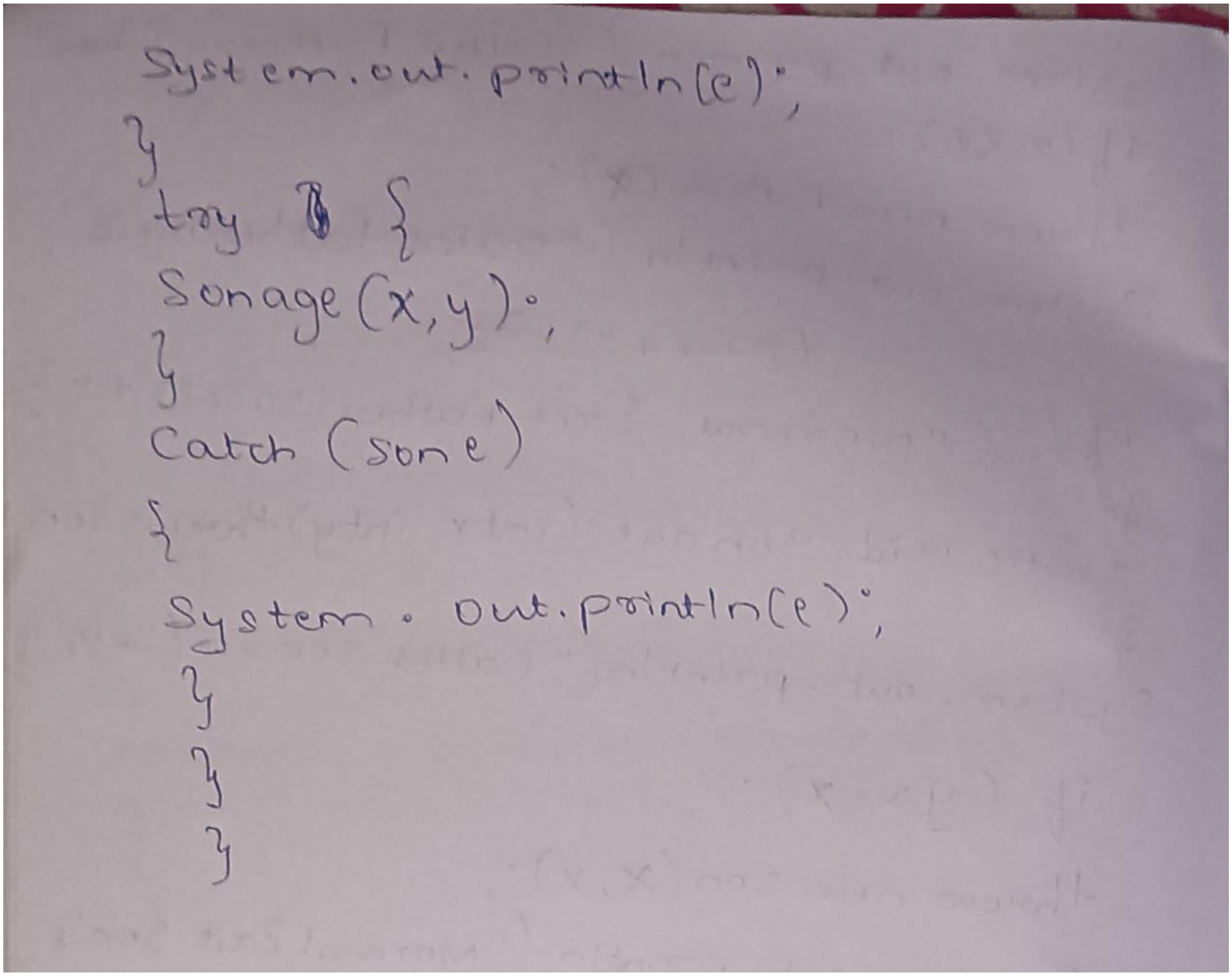
13/1/23

USER DEFINED EXCEPTION

```
import java.util.scanner.
Class father extends Exception{
 int fage;
  father (int x)
{
  fage=x;
 }
 Public String tostring(){
  return " father's age is wrong";
  }
  }

 Class Son extends father {
          int Sage;
   Son (int x, inty)
   {
    Super (x);
    Sage=y;
   }
    Public String to String() {
   return " Son's age is greater than
    or Equal tofather";

  }
  }

   Class wrongage {
       Static int x,y;
       Static void fatherage (intx)throws
                        father
   {
```



```
System.out.println ("Called Fatherage("+x+")");
 if (x<0)
 throw new father (x);
System.out.println (~~"called fatherage is~~
               ~~("+x+")");~~
 ~~if~~  ("Normal Exit Father's age is"+x);
}
static void sonage (int x, int y) throws son
{
System.out.println(" Called sonage("+y+")");
  if (y>=x)
  throw new son(x,y);
  System.out.println ("Normal Exit Son's
                      age is "+y);
}
public Static void main (String args [])
{
Scanner input = new scanner (system.in)
System.out.println(" Enter Father ag");
X=input.next Int();
System.out.println (" Enter Son age");
   y= input.next Int();
       try {
father age (x);
}
catch (Father e)
{
```



```
System.out.println(e);
}
try {
Sonage(x,y);
}
Catch (Sone)
{
System.out.println(e);
}
}
}
```

LAB PROGRAM 8: MULTI-THREADING

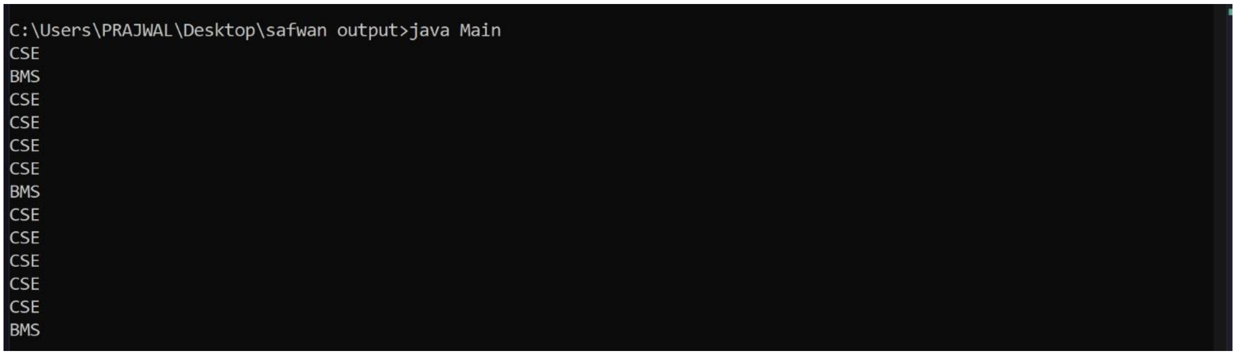
CODE

```
class MyThread extends Thread
{ long time;
        private volatile boolean running = true;
        MyThread(){
```

```
    System.out.println("Default"); }
MyThread(String name, long time)
{ super(name);
    this.time = time;
}    public    void
pause()
{ running = false;
}   public   void
run()
{
    try
    {
        while(running)
        {
            System.out.println(this.getName());
            Thread.sleep(time*1000);
        }
    }   catch(InterruptedException
     ie)
    {
        System.out.println("Exception caught in method");
    }


}
}
```

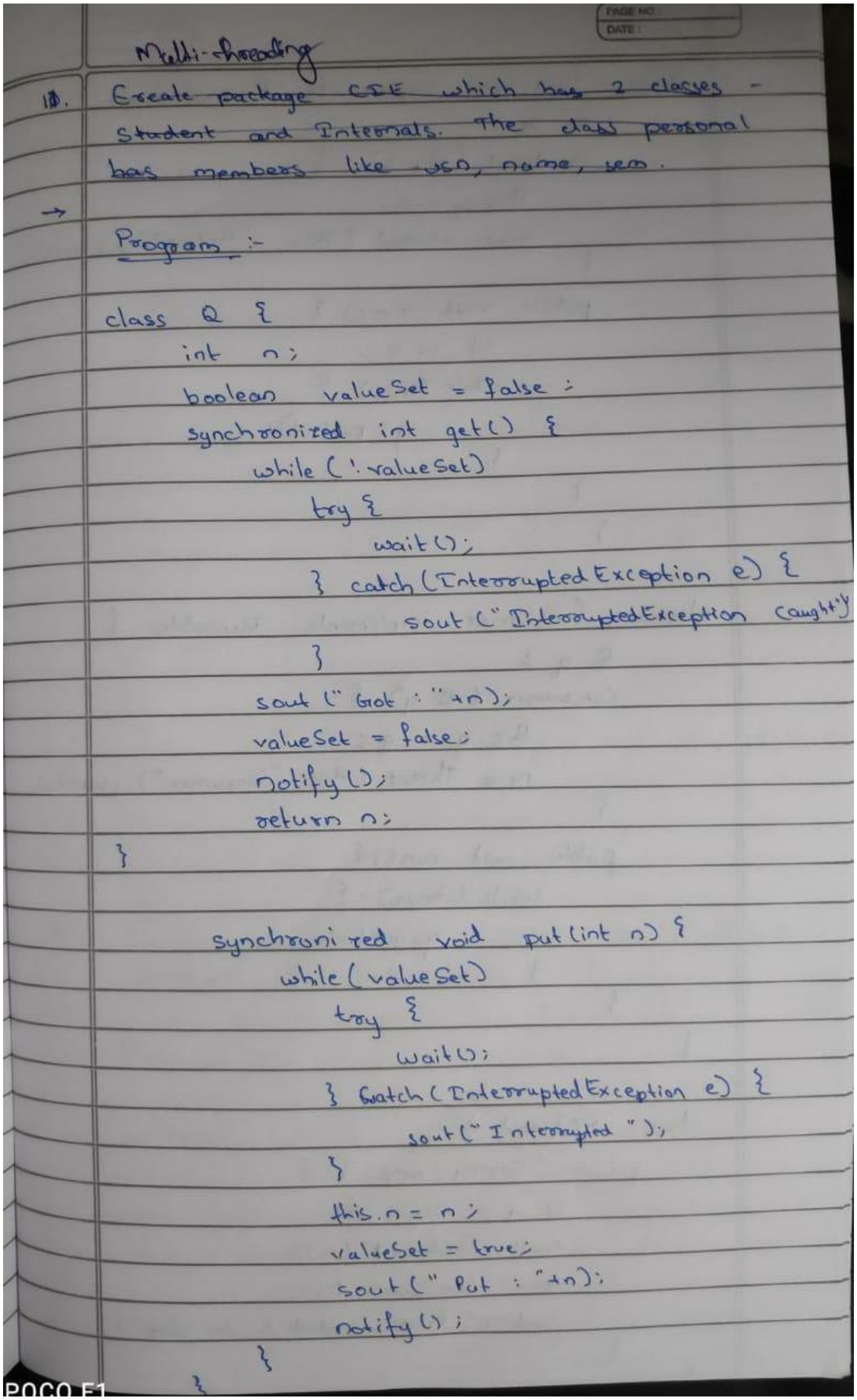
```
class Main
{ public static void main(String [] args)
    {
        MyThread mt1 = new MyThread("BMS",
        10); MyThread mt2 = new MyThread("CSE",
        2); mt1.start(); mt2.start();
        Try
        {
            Thread.sleep(20*1000);
            mt1.pause();
            mt2.pause();
        }
        catch(InterruptedException ie)
        {
            System.out.println("Exception caught in main"); }
    }
}
```

Output:



```
C:\Users\PRAJWAL\Desktop\safwan output>java Main
CSE
BMS
CSE
CSE
CSE
CSE
BMS
CSE
CSE
CSE
CSE
CSE
BMS
```

Observation Book:



PAGE NO :
DATE :

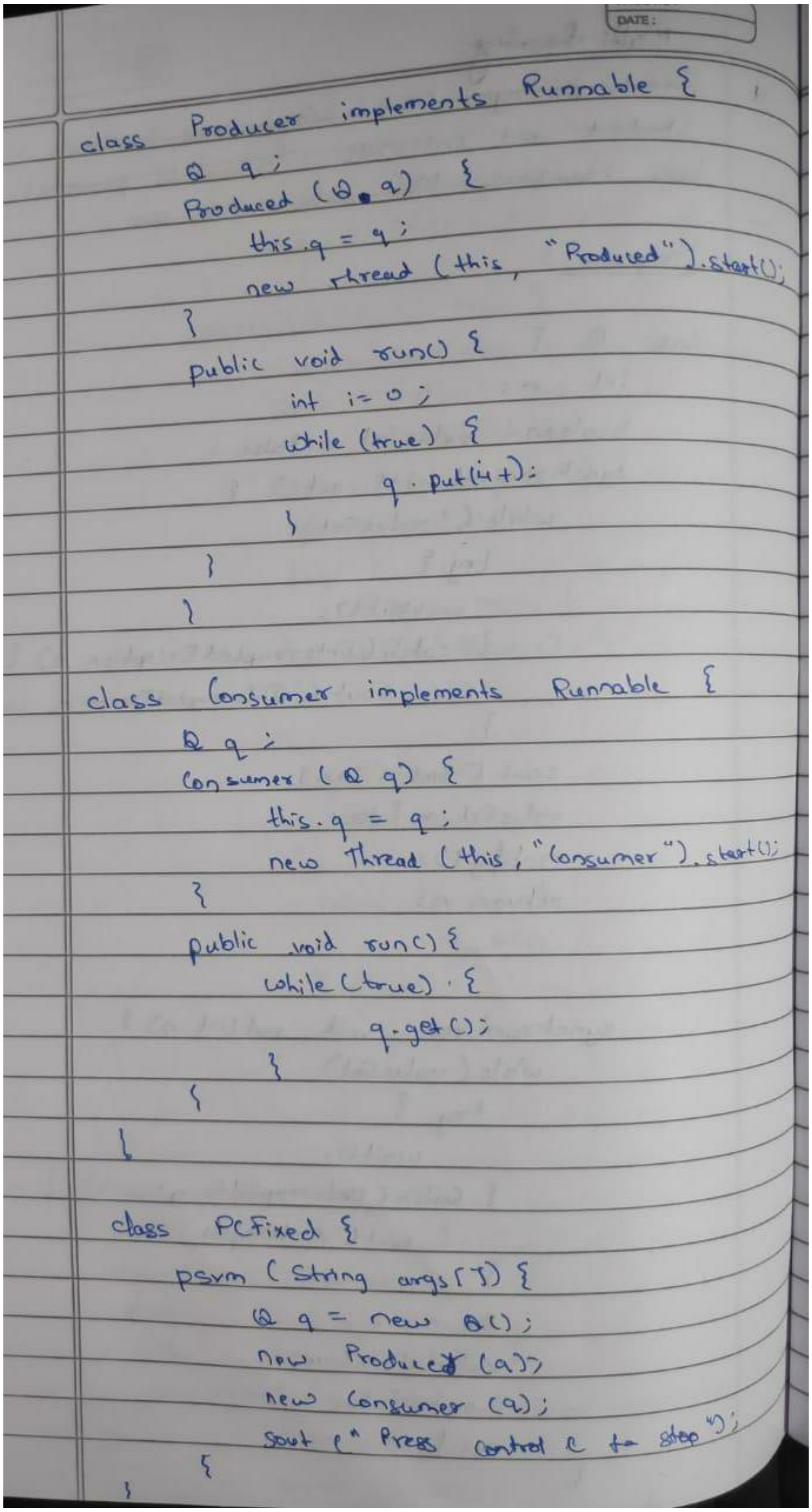
Multi-threading

10. ~~Create package CIE which has 2 classes - Student and Internals. The class personal has members like usn, name, sem.~~

→

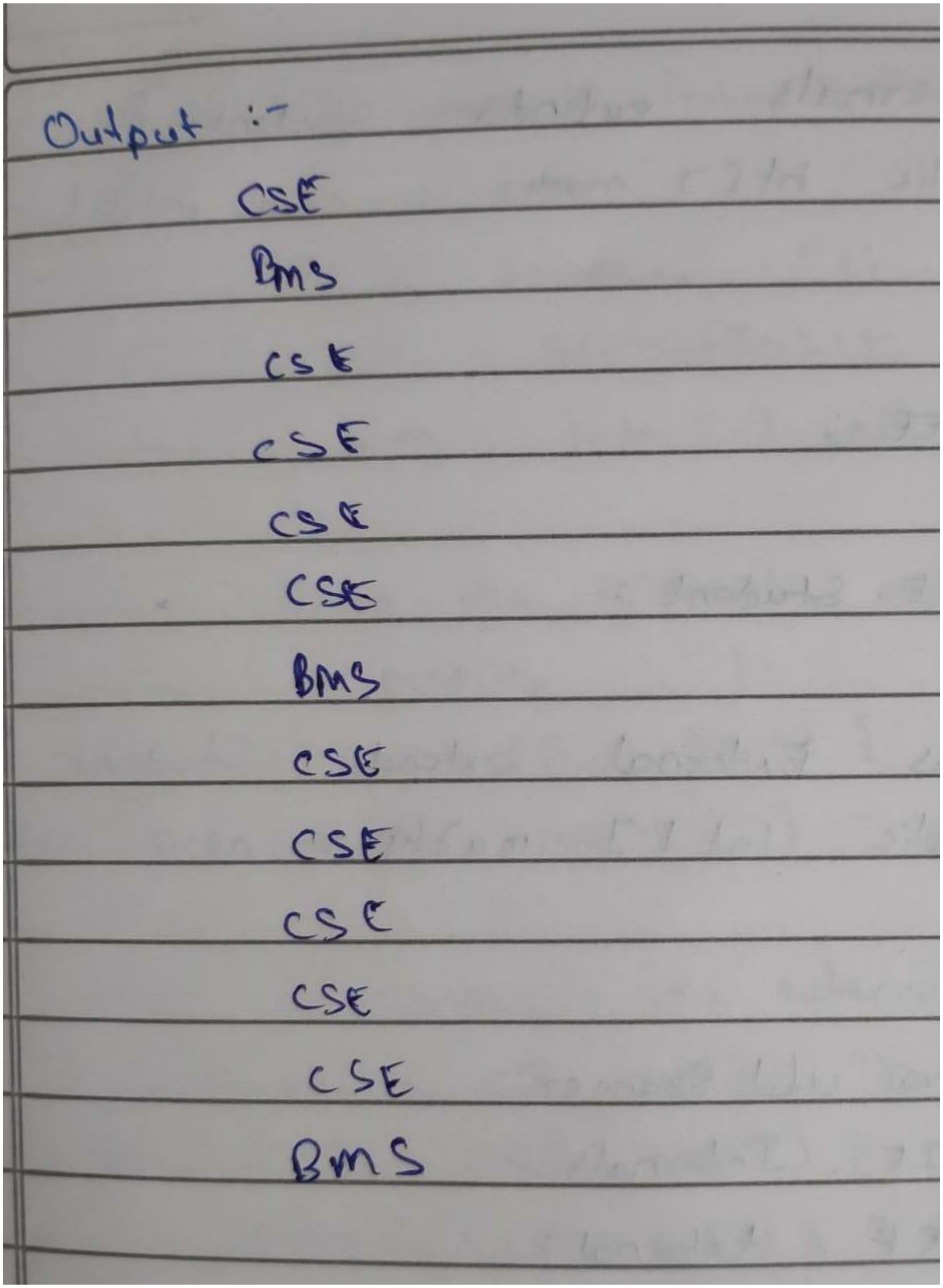
Program :-

```
class Q {
        int n;
        boolean valueSet = false;
        synchronized int get() {
              while (!valueSet)
                    try {
                          wait();
                    } catch (InterruptedException e) {
                          sout ("InterruptedException caught");
                    }
              sout (" Got : "+n);
              valueSet = false;
              notify();
              return n;
}

           synchronized void put(int n) {
                 while (valueSet)
                       try {
                             wait();
                       } catch (InterruptedException e) {
                             sout(" Interrupted ");
                       }
                       this.n = n;
                       valueSet = true;
                       sout (" Put : "+n);
                       notify();
           }
     }
```



DATE :

```
class Producer implements Runnable {
    Q q ;
    Produced (Q q) {
        this.q = q ;
        new thread (this, "Produced").start();
    }
    public void run() {
        int i= 0 ;
        while (true) {
            q . put(i++);
        }
    }
    }

class Consumer implements Runnable {
    Q q ;
    Consumer ( Q q) {
        this. q = q ;
        new Thread (this, "Consumer").start();
    }
    public void run() {
        while (true) . {
            q.get();
        }
    {
}

class PCFixed {
    psvm ( String args[]) {
        Q q = new Q();
        new Producer (q);
        new Consumer (q);
        sout (" Press control c to stop ");
    {
}
```



Output :-

```
        CSE
        Bms
        CSE
        CSE
        CSE
        CSE
        BMS
        CSE
        CSE
        CSE
        CSE
        CSE
        BmS
```