VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT

on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

SRUSHTI SUNKAD (1BM23CS341)

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-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**" carried out by **SRUSHTI SUNKAD(1BM23CS341),** who is a 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 2024-25. The Lab report has been approved as it satisfies the academic requirements in respect of **Object-Oriented Java Programming Lab - (23CS3PCOOJ)** work prescribed for the said degree.

Dr. Nandhini Vineeth

Associate Professor,
Department of CSE,
BMSCE, Bengaluru

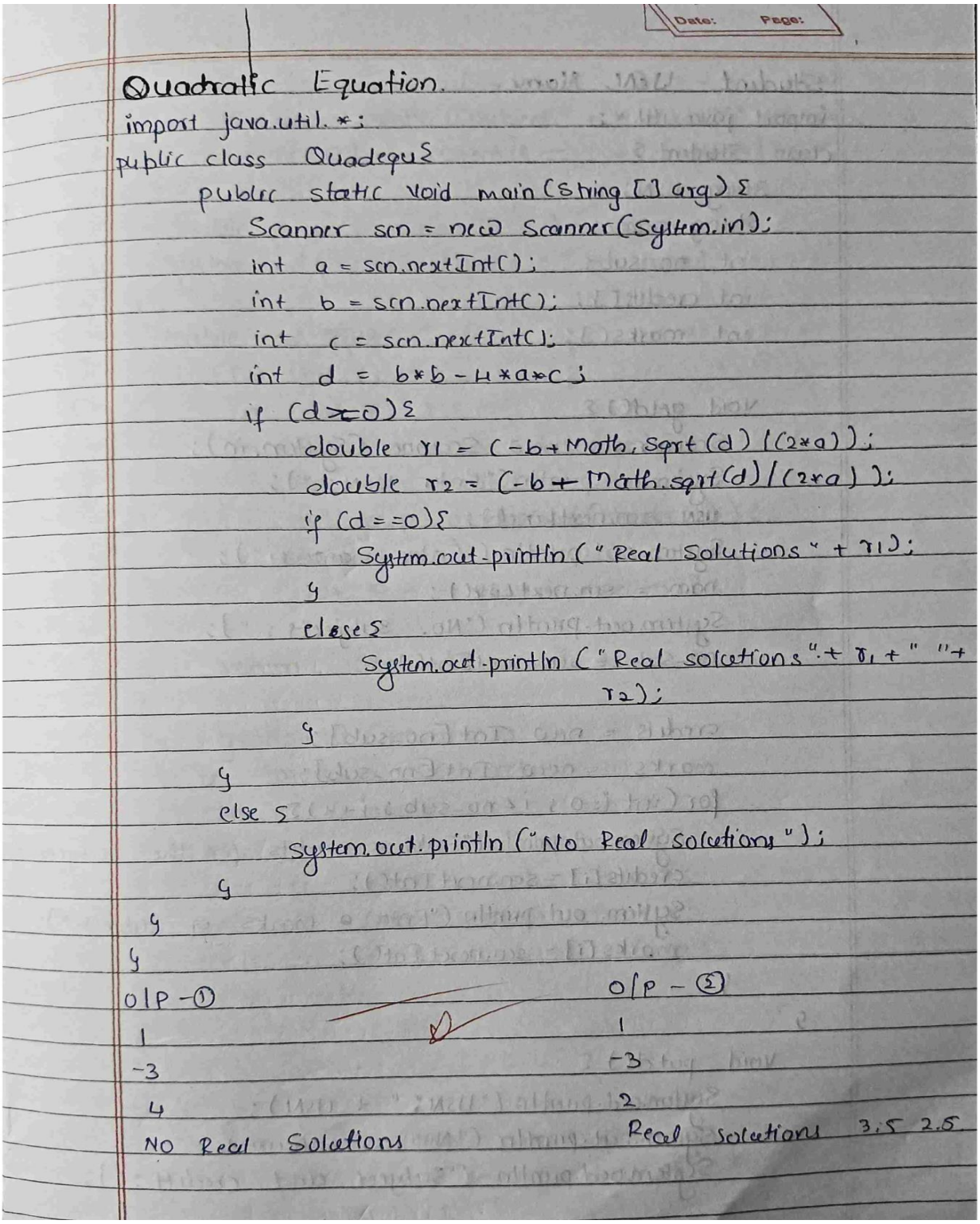
Dr. Kavitha Sooda

Professor and Head,
Department of CSE
BMSCE, Bengaluru

INDEX

LABORATORY PROGRAM - 1

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



Quadratic Equation.

```
import java.util.*;
public class Quadequ{
    public static void main(String [] arg){
        Scanner scn = new Scanner(System.in);
        int a = scn.nextInt();
        int b = scn.nextInt();
        int c = scn.nextInt();
        int d = b*b-4*a*c;
        if (d>=0){
            double r1 = (-b+Math.sqrt(d))/(2*a));
            double r2 = (-b+Math.sqrt(d)/(2*a));
            if (d==0){
                System.out.println("Real Solutions" + r1);
            }
            else{
                System.out.println("Real solutions".+ r1 + " "+
                                    r2);
            }
        }
        else {
            System.out.println("No Real solutions");
        }
    }
}
```

O/P - ①

```
1
-3
4
```

No Real Solutions

O/P - ②

```
1
-3
2
```

Real solutions 3.5 2.5

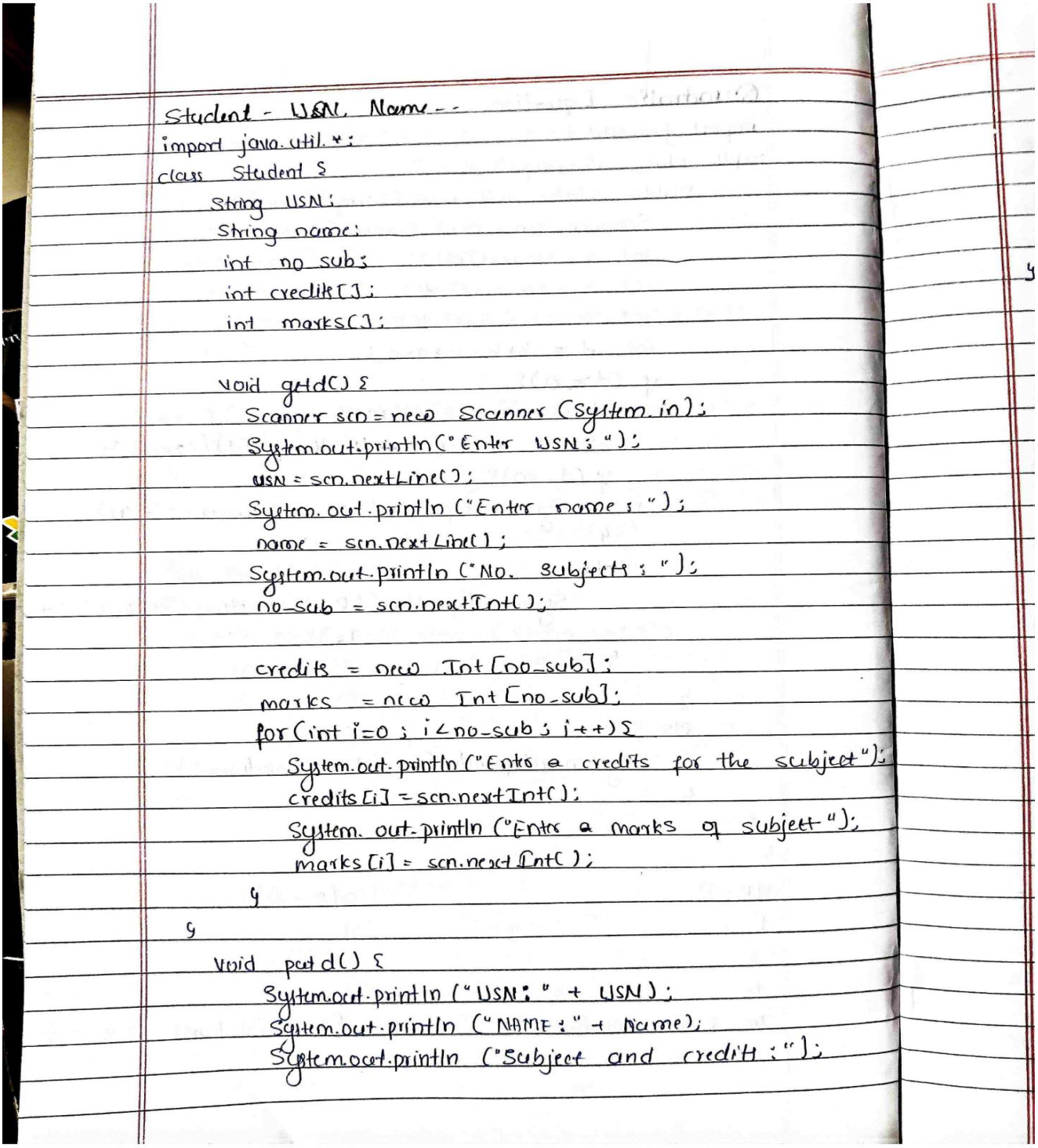
```
import java.util.*;
class QuadMain{
    public static void main(String[] args){
      Scanner scn=new Scanner(System.in);
      int a=scn.nextInt();
      int b=scn.nextInt();
      int c=scn.nextInt();
      int d=(b*b)-(4*a*c);
      if(d>=0){
        double r1=(-b+Math.sqrt(d)/(2*a));
        double r2=(-b-Math.sqrt(d)/(2*a));
        if(d==0){
            System.out.println("Real Solutions" +r1);
        }
        else{
          System.out.println("Real solutions: " +r1+" "+r2);
        }
      }
      else{
        System.out.println("No real solutions");
      }
   }
}
```

OUTPUT:

```
C:\1BM23CS341>javac QuadMain.java

C:\1BM23CS341>java QuadMain
1
2
3
No real solutions

C:\1BM23CS341>java QuadMain
1
-5
4
Real solutions: 6.5 3.5
```

LABORATORY PROGRAM - 2

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



```
Student - USN, Name--
import java.util.*;
class Student {
    String USN;
    String name;
    int no_sub;
    int credits[];
    int marks[];

    void getd() {
      Scanner scn = new Scanner (System.in);
      System.out.println("Enter USN : ");
      USN = scn.nextLine();
      System.out.println("Enter name : ");
      name = scn.nextLine();
      System.out.println("No. subjects : ");
      no_sub = scn.nextInt();

      credits = new Int[no_sub];
      marks = new Int[no_sub];
      for(int i=0 ; i<no_sub ; i++) {
        System.out.println("Enter a credits for the subject");
        credits[i] = scn.nextInt();
        System.out.println("Enter a marks of subject");
        marks[i] = scn.nextInt();
      }
}
    void putd() {
      System.out.println("USN: " + USN);
      System.out.println("NAME:" + Name);
      System.out.println("Subject and credits:");
```

```
for (int i=0; i<no_sub ; i++){
    System.out.println("subject " + (i+1) +":" "Marks = " +
    marks[i] + ", credits = " + credits[i]);
}
double sgpa = cale_SGPA();
System.out.println("SGPA :" + sgpa);
}
double cal_SGPA(){
    int tot_c =0;
    int tot_P =0;
    for (int i=0; i<no_sub; i++){
        tot_c+ = credits[i];
        int gr_p = gradepoint(marks[i]);
        tot_P + = gr_p * credits[i];
    }
    return (double) tot_p / tot_c ;
}
int gradepoint (int marks){
    if (marks >=90) return 10;
    else if (marks >=80) return 9;
    else if (marks >=70) return 8;
    else if (marks >=60) return 7;
    else if (marks >=50) return 6;
    else if (marks >=40) return 5;
    else return 0;
  }
}
public class Main {
    public static void main(String[] args){
        Student s1 = new Student();
        s1.getd();
        s1.putd();
```

o/p

Enter USN : 1BM23CS341

Enter Name : Srushti Sankad

Enter number of subjects:

2

Enter credits for subject : 4

Enter marks for subject: 90

Enter credits for subject : 3

Enter mark for subject : 98

USN : 1BM23CS341

Name : Srushti Sankad

Subjects and credits

Subject 1 : Marks = 90, Credits = 4

Subject 2 : Marks = 98, credits = 3

SGPA : 10.0.

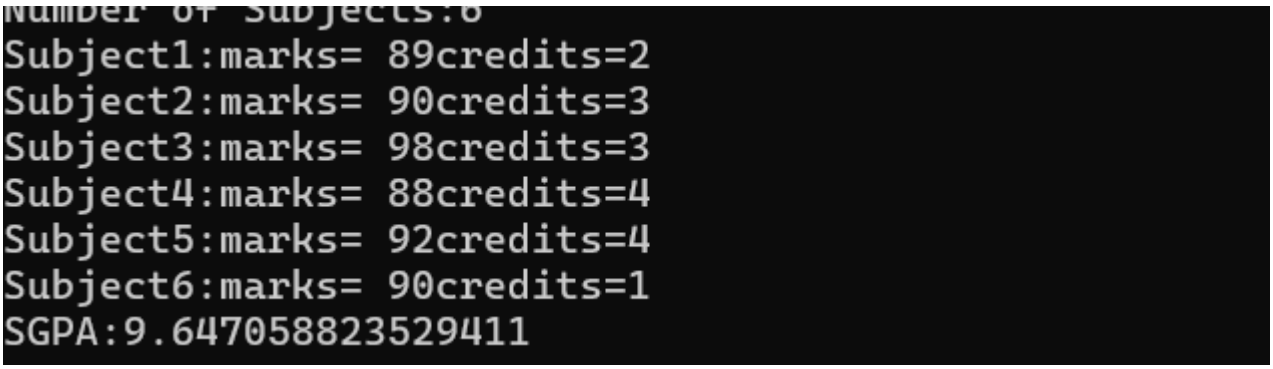
```
import java.util.*;
class Student{
    String USN;
    String name;
    int no_sub;
    int credits[];
    int marks[];

   void getd(){
     Scanner scn=new Scanner(System.in);
     System.out.println("Enter USN: ");
     USN=scn.nextLine();
     System.out.println("Enter name: ");
     name=scn.nextLine();
     System.out.println("Enter number of subjects: ");
     no_sub=scn.nextInt();

     credits=new int[no_sub];
     marks=new int[no_sub];
     for(int i=0;i<no_sub;i++){
        System.out.println("Enter credits: ");
        credits[i]=scn.nextInt();
        System.out.println("Enter Mraks: ");
        marks[i]=scn.nextInt();
    }
  }

  void putd(){
      System.out.println("USN:" + USN);
      System.out.println("Name:"+name);
      System.out.println("Number of Subjects:" + no_sub);
      for(int i=0;i<no_sub;i++){
          System.out.println("Subject"+(i+1)+ ":"+"marks= "+ marks[i]+ "credits="+credits[i]);
      }
      double sgpa=cal_sgpa();
      System.out.println("SGPA:" + sgpa);
  }
  double cal_sgpa(){
      int total_credits=0;
      int total_points=0;
      for(int i=0;i<no_sub;i++){
          total_credits+=credits[i];
          int grade_p=gradepoints(marks[i]);
          total_points+=grade_p *credits[i];
      }
      return (double) total_points/total_credits;
  }

 int gradepoints(int marks){
    if(marks>=90) return 10;
    else if(marks>=80) return 9;
```

```
    else if(marks>=70) return 8;
    else if(marks>=60) return 7;
    else if(marks>=50) return 6;
    else if(marks>=40) return 5;
    else return 0;
  }
}

class StudentMain{
    public static void main(String[] args){
      Scanner scn=new Scanner(System.in);
      System.out.println("Enter number of students:");
      int n=scn.nextInt();
      Student s1=new Student();
      for(int i=0;i<n;i++){
      s1.getd();
      s1.putd();
      }
   }
}
```

OUTPUT:

```
C:\1BM23CS341>javac StudentMain.java

C:\1BM23CS341>java StudentMain
Enter number of students:
2
Enter USN:
1BM23CS100
Enter name:
Priya
Enter number of subjects:
6
Enter credits:
3
Enter Mraks:
90
Enter credits:
3
Enter Mraks:
99
Enter credits:
4
Enter Mraks:
98
Enter credits:
4
Enter Mraks:
97
Enter credits:
2
Enter Mraks:
89
Enter credits:
1
Enter Mraks:
90
USN:1BM23CS100
Name:Priya
Number of Subjects:6
Subject1:marks= 90credits=3
Subject2:marks= 99credits=3
```

```
Subject3:marks= 98credits=4
Subject4:marks= 97credits=4
Subject5:marks= 89credits=2
Subject6:marks= 90credits=1
SGPA:9.882352941176471
Enter USN:
1BM23CS300
Enter name:
Shrichetan
Enter number of subjects:
6
Enter credits:
2
Enter Mraks:
89
Enter credits:
3
Enter Mraks:
90
Enter credits:
3
Enter Mraks:
98
Enter credits:
4
Enter Mraks:
88
Enter credits:
4
Enter Mraks:
92
Enter credits:
1
Enter Mraks:
90
USN:1BM23CS300
Name:Shrichetan
Number of Subjects:6
Subject1:marks= 89credits=2
Subject2:marks= 90credits=3
Subject3:marks= 98credits=3
```


Number of Subjects:6
Subject1:marks= 89credits=2
Subject2:marks= 90credits=3
Subject3:marks= 98credits=3
Subject4:marks= 88credits=4
Subject5:marks= 92credits=4
Subject6:marks= 90credits=1
SGPA:9.647058823529411

LABORATORY PROGRAM - 3

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

Q3 Create a class Book which contains four members : name, author, price, num_pages. Include a constructor to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a java program to create a n book objects.

=>

```
class Book
{ String name;
  String author;
  int price;
  int num_pages;

  Scanner scn = new Scanner (System.in);

  Book() { }
  Book (String name, String author, int price, int
        num_pages){
      this.name = name;
      this.stringauthor = author;
      this.price = price;
      this.num_pages = num_pages;
  }
  void getd()
  {
    System.out.println ("Enter Book name : ");
    name = scn.nextLine();
    System.out.println ("Enter author name : ");
    author = scn.nextLine();
    System.out.println ("Enter price of Book : ");
    price = scn.nextInt();
    System.out.println ("Enter number of pages : ");
    num_pages = scn.nextInt();
  }
```

```
void putd(){
    System.out.println("Book name :" + name);
    System.out.println("Authour Name :" + author);
    System.out.println("Book Price :" + price);
    System.out.println("Number of pages :" + num_pages);
}

public String toString(){
    return "Book Name = " + name +
           ", Author Name = " + author +
           ", Book Price = " + price +
           ", Number of pages = " + num_pages +
           "}"
}
}
class BookMain{
public static void main(String[] args){
    Scanner scn = new Scanner(System.in);
    Book b = new Book("Wings of Fire", "APJ Abdul
                       Kalam", 250, 500);
    b.putd();
    S.O.P("Enter number of Books");
    int n = scn.nextInt();
    Book b1[] = new Book[n];
    for(int i=0; i<n; i++){
        b1[i].getd();
        b1[i].putd();
    }
    System.out.println("All Details of Books");
    for(int i=0; i<n; i++){
        System.out.println(b1[i].toString());
    }
}
}
```

O/P

Book name : Wings of Fire
Author name : APJ Abdul Kalam.
Book Price : 250
Number of pages : 500

Enter number of Books : 2
Enter name of book : xyz
Enter author name : pqr
Enter price : 300
Enter number of pages : 700

Book name : xyz
Author name : pqr
Book Price : 300
Number of pages : 700

Enter name of Book : ABC
Enter author name : abc
Enter price : 450
Enter number of pages : 999

Book name : ABC
Author name : abc
Book price : 450
Number of Pages : 999

Book 1 Book Name = xyz, Author Name = pqr, Book Price = 300
Number of pages = 700 4.
Book 2 Book Name = ABC, Author Name = abc, Book Price = 450
Number of pages = 999 4

```
import java.util.*;
class Book{
   String name;
   String author;
   int price;
   int no_pages;

   Book(){}

   Scanner scn=new Scanner(System.in);
   Book(String name, String author,int price, int no_pages){
         this.name=name;
         this.author=author;
         this.price=price;
         this.no_pages=no_pages;
  }
  void getd(){
     System.out.println("Enter book name:");
     name=scn.nextLine();
     System.out.println("Enter author name:");
    author=scn.nextLine();
    System.out.println("Enter price:");
    price=scn.nextInt();
    System.out.println("Enter number of pages:");
    no_pages=scn.nextInt();
 }
 void putd(){
     System.out.println("Book name:" + name);
     System.out.println("Author name:" + author);
     System.out.println("Book price:" + price);
     System.out.println("Number of pages:" + no_pages);
 }

 public String toString(){
    return "Book{" + "Name="+name+",Author="+author+",Price="+price+",Number of
pages="+no_pages+"}";
 }
}

class BookMain{
    public static void main(String[] args){
      Scanner scn=new Scanner(System.in);
      Book b1=new Book("BalaswamyBook","Balaswamy", 499, 900);
      b1.putd();
      System.out.println("Enter number of books");
      int n=scn.nextInt();
      Book[] b=new Book[n];
      for(int i=0;i<n;i++){
          b[i]=new Book();
          b[i].getd();
          b[i].putd();
```

```
    }
    System.out.println("All book details:");
    for(int i=0;i<n;i++){
        System.out.println(b[i].toString());
    }
  }
}
```

OUTPUT:

```
C:\1BM23CS341>javac BookMain.java

C:\1BM23CS341>java BookMain
Book name:BalaswamyBook
Author name:Balaswamy
Book price:499
Number of pages:900
Enter number of books
2
Enter book name:
XYZ
Enter author name:
ABC
Enter price:
900
Enter number of pages:
1079
Book name:XYZ
Author name:ABC
Book price:900
```

LABORATORY PROGRAM - 4

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

Q4. Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea().

⇒
```
abstract class shape {
  int dim1, dim2;
  Shape (int x, int y) {
    dim1 = x;
    dim2 = y;
    abstract double printArea();
  }

  class Rectangle extends shape {
  Rectangle (int a, int b) {
    super (a, b);
  }
  double printArea () {
  return dim1 * dim2;
  }
  }

  class Triangle extends Shape {
  Triangle (int a, int b) {
    super (a,b);
  }
  double printArea() {
    return 0.5 * dim1 * dim2;
  }
  }

  class circle extends Shape {
  circle (int a, int b) {
      super (a,b);
  }
```

```
double printArea(){
    return 3.14*dim1*dim1;
}
}

class AbstractDemo{
    public static void main(String ss[]){
        Rectangle r = new Rectangle(100, 240);
        Triangle t = new Triangle(10, 20);
        Circle c = new circle(10.0);

        System.out.println("Area of Rectangle: "+r.printArea());
        System.out.println("Area of Triangle: "+t.printArea());
        System.out.println("Area of Circle: "+c.printArea());
    }
}
```

O/P

Area of Rectangle : 24000.0

Area of Triangle : 100.0

Area of Circle : 314.0

```
abstract class Shape
{
   int dim1,dim2;
   Shape(int x,int y){
      dim1=x;
      dim2=y;
   }
   abstract double PrintArea();
}

class Rectangle extends Shape
{
  Rectangle(int a,int b){
     super(a,b);
  }
  double PrintArea(){
     return dim1*dim2;
  }
}

class Triangle extends Shape
{
  Triangle(int a,int b){
     super(a,b);
  }
  double PrintArea(){
     return 0.5*dim1*dim2;
  }
}

class Circle extends Shape
{
  Circle(int a,int b){
     super(a,b);
  }
  double PrintArea(){
     return 3.14*dim1*dim1;
  }
}
class AbstractMain
{
   public static void main(String[] args){
     Rectangle r=new Rectangle(100,240);
     Triangle t=new Triangle(10,20);
     Circle c=new Circle(10,0);

     System.out.println("Area of rectangle:"+r.PrintArea());
     System.out.println("Area of Triangle:"+t.PrintArea());
     System.out.println("Area of Circle:"+c.PrintArea());
  }
}
```

OUTPUT:

```
C:\1BM23CS341>javac AbstractMain.java

C:\1BM23CS341>java AbstractMain
Area of rectangle:24000.0
Area of Triangle:100.0
Area of Circle:314.0
```

LABORATORY PROGRAM - 5

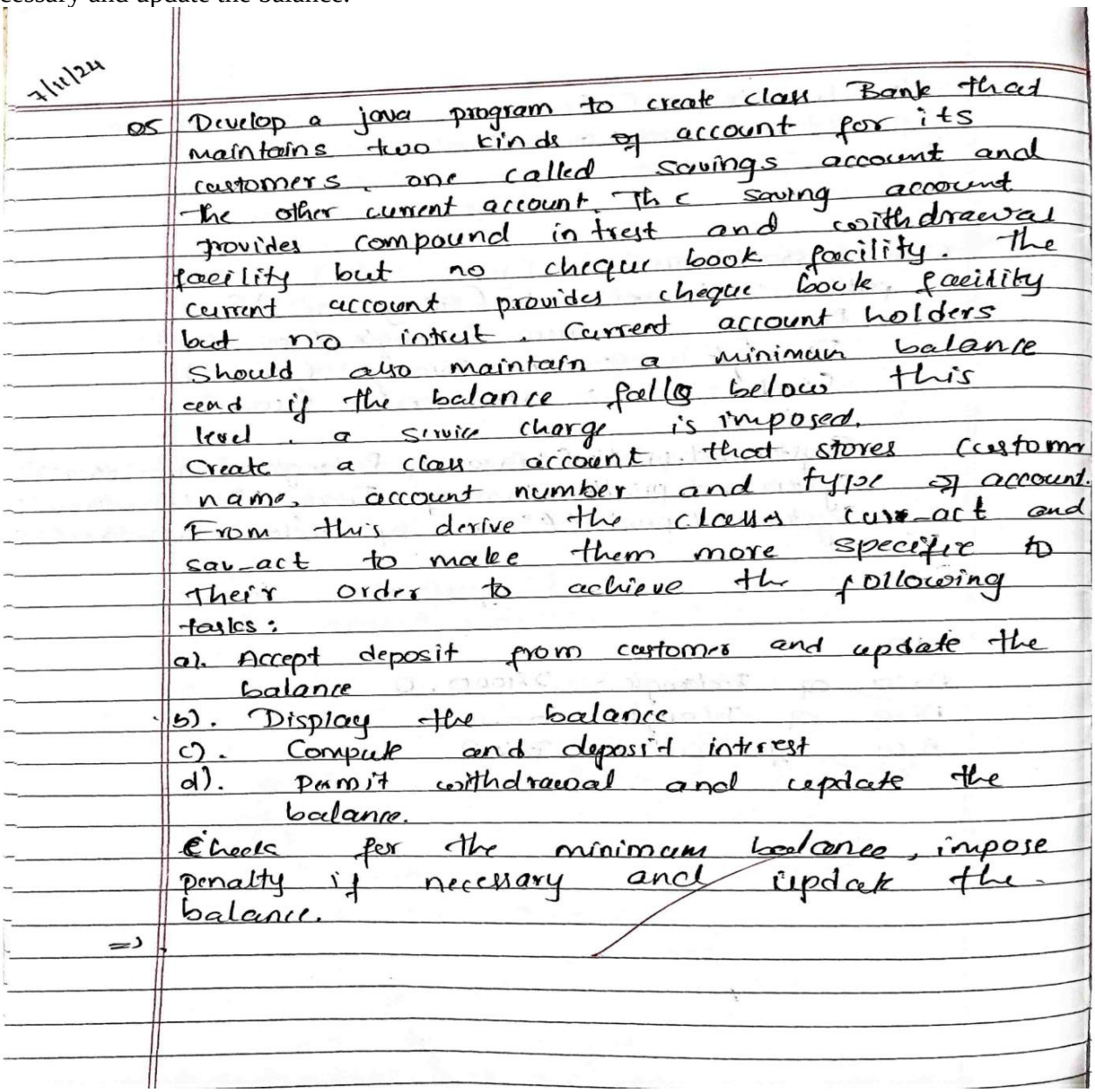
Q).Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called a savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update
the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance
Check for the minimum balance, impose penalty if
necessary and update the balance.



```
import java.util.*;
abstract class Account
{
    String cust_name, Acc_num;
    double balance;
    Account (String cust_name, String Acc_num,
        double inti_balance){
        this.cust_name = cust_name;
        this.Acc_num = Acc_num;
        this.balance = inti_balance;
    }
    abstract void deposit (double amt);
    abstract void displayBalance();
    abstract void withdraw (double amt);
}

class Sav_Acct extends Account{
    double interestRate;
    Sav_Acct (String cust_name, String Acc_num,
        double inti_balance ){
        super(cust_name, Acc_num, intia_balance);
        this.interstRate = interestRate;
    }
    void deposit(double amt){
        balance += amt;
    }
    void displayBalance(){
        System.out.println("Saving Balance :"+balance);
    }
    void withdraw(double amt){
        if (amt <= balance){
            balance -= amt;
        }
    }
```

```
    void computAndDepositInterest(){
        balance += balance * interestRate/100;
    }
}

class Cur_Acct extends Account {
    static final double MIN_BALANCE=1000, SERVICE_
        CHARGE = 50;
    Cur_Acct(String Cust_name, String Acc_num,
            double inti_balance) {
        super(Cust_name, Acc_num, inti_balance);
    }
    void deposit(double amt) {
        balance += amt;
    }
    void displayBalance() {
        System.out.println("Saving Balance: "+balance);
    }
    void withdraw(double amt) {
        if (amt <= balance) {
            balance -= amt;
            if (balance < MIN_BALANCE) {
                balance -= SERVICE_CHARGE;
            }
        }
    }
}
class Bank
{
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
```

```
System.out.println("Enter account type(savings/current):");
String type = scn.nextLine();

System.out.println("Enter account name:");
String name = scn.nextLine();

System.out.println("Enter account number:");
String num = scn.nextLine();

Account account;
if(type.equals("savings")){
    System.out.println("Initial balance and
                        interest rate:");
    account=new Sav_Acct(name, number,
                         scn.nextDouble());
}
else{
    System.out.println("1.Deposit 2.Display Balance.
                        3.Withdraw 4.Interest 5.Exit");
    System.out.println("Initial balance");
    account = new Cur_Acct(name, number,
                           scn.nextDouble());
}
while(true){
    System.out.println("1.Deposit 2.Display Balance
                        3.Withdraw 4.Interest.
                        5..Exit");
    int choice = scn.nextInt();
    switch(choice){
        case1 : account.deposit(scn.nextDouble());
                break;
        case2 : account.displayBalance();
                break;
```

```
case 3 : account.withdraw(scn.nextDouble());
         break;
case 4 : if(account instanceof Sav_Acct){
          ((Sav_Acct)account).ComputeAnd
          DepositInterest();
         }
         break;
case 5 : return;
}
}
}
}
```

O/P

Enter account type (Savings/current): Savings
Enter account name : ~~John~~ Srishti
Enter account number : 1A0010321
Initial balance : 650
1. Deposit 2. Display Balance 3. Withdraw
4. Interest 5. Exit.
2.
Saving Balance = 650.0
1
1000
2
Saving Balance = 1650.0
3
500
2
Saving Balance = 1150.0
5.

```
import java.util.Scanner;

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

  Account(String customerName, String account Number, double initialBalance) {
    this.customerName = customerName;
    this.accountNumber = accountNumber:
    this.balance initial Balance;
  }
  abstract void deposit(double amount);
  abstract void displayBalance();
  abstract void withdraw(double amount);
}

class SavAcct extends Account {
  double interestRate:

  SavAcct(String customerName, String accountNumber, double initialBalance, double interestRate){
    super(customerName, accountNumber, initialBalance);
    this.interest Rate = interestRate:
  }
  void deposit(double amount) {
   balance + amount:
  }
  void displayBalance(){
    System.out.println("Savings Balance: " + balance);
  }
  void withdraw(double amount){
    if (amount <= balance) balance -- amount;
  }
  void computeAndDepositInterest(){
    balance + balance interest Rate / 100;
  }
}

class CurAcct extends Account{
  static final double MIN BALANCE = 1000, SERVICE CHARGE = 50;

  CurAcct(String customerName, String accountNumber, double initialBalance){
    super(customerName, account Number, initialBalance);
  }
  void deposit(double amount) {
    balance + amount;
  }
  void displayBalance(){
    System.out.println("Current Balance: " + balance);
  }
  void withdraw(double amount) {
    if (amount <= balance) {
```

```
        balance - amount:if (balance < MIN BALANCE) balance - SERVICE CHARGE;
       }
    }
}

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

    System.out.println("Enter account type (savings/current): ");
    String type = scanner.nextLine();

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

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

    Account account;

   if(type.equals("savings")) {
      System.out.println("Initial balance and interest rate: ");
      account = new SavAcct(name, number, scanner.nextDouble(), scanner.nextDouble());
   }
   else{
     System.out.println("Initial balance: ");
     account = new CurAcct(name, number, scanner.nextDouble());
   }
   while (true) {
     System.out.println("\n1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit");
     int choice scanner.nextInt();

     switch (choice) {
       case 1: account.deposit(scanner.nextDouble());
               break;
       case 2: account.displayBalance();
               break;
       case 3: account.withdraw(scanner.nextDouble());
               break;
       case 4: if (account instanceof SavAcct) ((SavAcct)account).computeAndDepositInterest();
               break;

       case 5:return;
     }
   }
 }
}
```

OUTPUT:

```
D:\24BMSCE>javac Bank.java

D:\24BMSCE>java Bank
Enter account type (savings/current):
savings
Enter customer name:
anu rai
Enter account number:
123786645087301
Initial balance and interest rate:
5000
50

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
1
200

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
2
Savings Balance: 5200.0

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
3
100

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
4

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
2
Savings Balance: 7650.0

1. Deposit 2. Display Balance 3. Withdraw 4. Interest 5. Exit
5
```

LABORATORY PROGRAM - 6

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

14/11/24

Bafna Gold
Date: Page:

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

```
Package CIE;
import java.util.*;
public class Student
{
    public String USN;
    public string Name;
    public int sem;

    Scanner scn = new Scanner (System.in);

public void getd()
    System.out.println();
    scn.nextLine();
    System.out.println ("Enter USN :");
    USN = scn.nextLine();
    System.out.println ("Enter Name:");
    Name = scn.nextline();
    System.out.println ("Enter Sem");
    sem = scn.nextInt();
}
```

```
    public void putd(){
        System.out.println("USN: "+ USN);
        System.out.println("Name: "+ Name);
        System.out.println("Sem: "+ sem);
    }
}

Package CIE;
import java.util.*;
public class Internals
{
    Scanner scn = new Scanner(System.in);
    public int Cmarks[] = new int[5];
    public void getd(){
        System.out.println("Enter CIE marks:");
        for(int i=0;i<5;i++){
            Cmarks[i] = scn.nextInt();
        }
    }
    public void putd(){
        for(int i=0;i<5;i++){
            System.out.print(Cmarks[i]+ " ");
        }
    }
}

Package SEE;
import java.util.*;
Import ~ CIE.Student;
import java.util.*;
```

```
public class External extends Students
{
    Scanner scn = new Scanner (System.in);
    public int Smarks [] = new int [5];
    public void getd( ){
        System.out.println ("Enter SEE marks:");
        for (int i=0; i<5; i++){
            Smarks [i] = scn.nextInt();
        }
    }
    public void putd(){
        for (int i=0; i<5; i++){
            System.out.print (Smarks[i] + " ");
        }
    }
    public void final (){
    public int getfinalmarks (int Idx, int Cmarks){
        return Cmarks + (Smarks [Idx] / 2);
    }
}
```

```
pack
import java.util.*;
import CIE.*;
import SEE.*;

class C_SMain
{
    public static void main (String [] args){
        Scanner scn = new Scanner (System.in);
        System.out.println ("Enter number of students");
        int n = scn.nextInt();
```

```
Student s = new Student();

Internals I[] = new Internals[n];
External E[] = new External[n];

for (int i=0 ; i<n ;i++){
    s.getId();
    s.putId();

    I[i] = new Internals();
    I[i].getId();
    I[i].putId();

    E[i] = new External();
    E[i].getId();
    E[i].putId();

    System.out.println("Final Marks");
    for (int j=0;j<5;j++){
        int finalmarks = E[i].getfinalmarks(j, I[i].Cmarks[j]);
        System.out.println(finalmarks);
    }
}
}
}
```

O/P

```
Enter number of Students : 2
Enter USN : 1BM23CS341
Enter Name : Sruthi
Enter Sem : 3
USN : 1BM23CS341
Name : Sruthi
Sem : 3
```

Bafna Gold
Date: Page:

Enter CIE marks: 41 45 46 47 48
41 45 46 47 48
Enter SEE marks: 90 91 95 94 92
90 99 95 94 92
Final Marks: 85 97 95 91 91

Enter USN: 1BM23CS222
Enter Name: Spoorti
Enter Sem: 3
USN = 1BM23CS222
Name: Spoorti
Sem: 3
Enter CIE marks: 44 45 43 42 41
44 45 43 42 41
Enter SEE marks: 90 99 98 94 91
90 99 98 94 91
Final Marks
89 94 92 89 86.

```
//CIE/Student
package CIE;
import java.util.*;


public class Student
{
  public String USN;
  public String Name;
  public int sem;

  Scanner scn=new Scanner(System.in);

  public void getd(){
     System.out.println();
     scn.nextLine();
     System.out.println("Enter USN:");
     USN=scn.nextLine();
     System.out.println("Enetr Name:");
     Name=scn.nextLine();
     System.out.println("Enetr sem:");
     sem=scn.nextInt();
  }

  public void putd(){
    System.out.println("USN:"+USN);
    System.out.println("Name:"+Name);
    System.out.println("Sem:"+sem);
}
}

//CIE/Internals
package CIE;
import java.util.*;

public class Internals
{
  Scanner scn=new Scanner(System.in);

  public int Cmarks[]=new int[5];

  public void getd(){
    System.out.println("Enter CIE marks:");
    for(int i=0;i<5;i++){
      Cmarks[i]=scn.nextInt();
    }
  }

 public void putd(){
    for(int i=0;i<5;i++){
```

```
      System.out.print(Cmarks[i]+" ");
    }
  }
}

//SEE/Externals
package SEE;
import CIE.Student;
import java.util.*;


public class External extends Student
{
  Scanner scn=new Scanner(System.in);

    public int Smarks[]=new int[5];

  public void getd(){
    System.out.println();
    System.out.println("Enter SEE marks:");
    for(int i=0;i<5;i++){
      Smarks[i]=scn.nextInt();
    }
  }

  public void putd(){
     for(int i=0;i<5;i++){
      System.out.print(Smarks[i]+" ");
     }
  }

  public int getFinalMark(int Idx, int internalMark) {
      return internalMark + (Smarks[Idx] / 2);
   }
}

//C_SMain
import java.util.*;
import CIE.*;
import SEE.*;

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

  System.out.println("Enter number of Students:");
  int n=scn.nextInt();

  Student s=new Student();
```

```
  Internals I[]=new Internals[n];
  External E[]=new External[n];

  for(int i=0;i<n;i++){
     s.getd();
     s.putd();

     I[i]=new Internals();
     I[i].getd();
     I[i].putd();

     E[i]=new External();
     E[i].getd();
     E[i].putd();

   System.out.println("Final Marks:");
         for (int j = 0; j < 5; j++) {
            int finalMark = E[i].getFinalMark(j, I[i].Cmarks[j]);
            System.out.println(finalMark);
         }
  }

   }
}
```

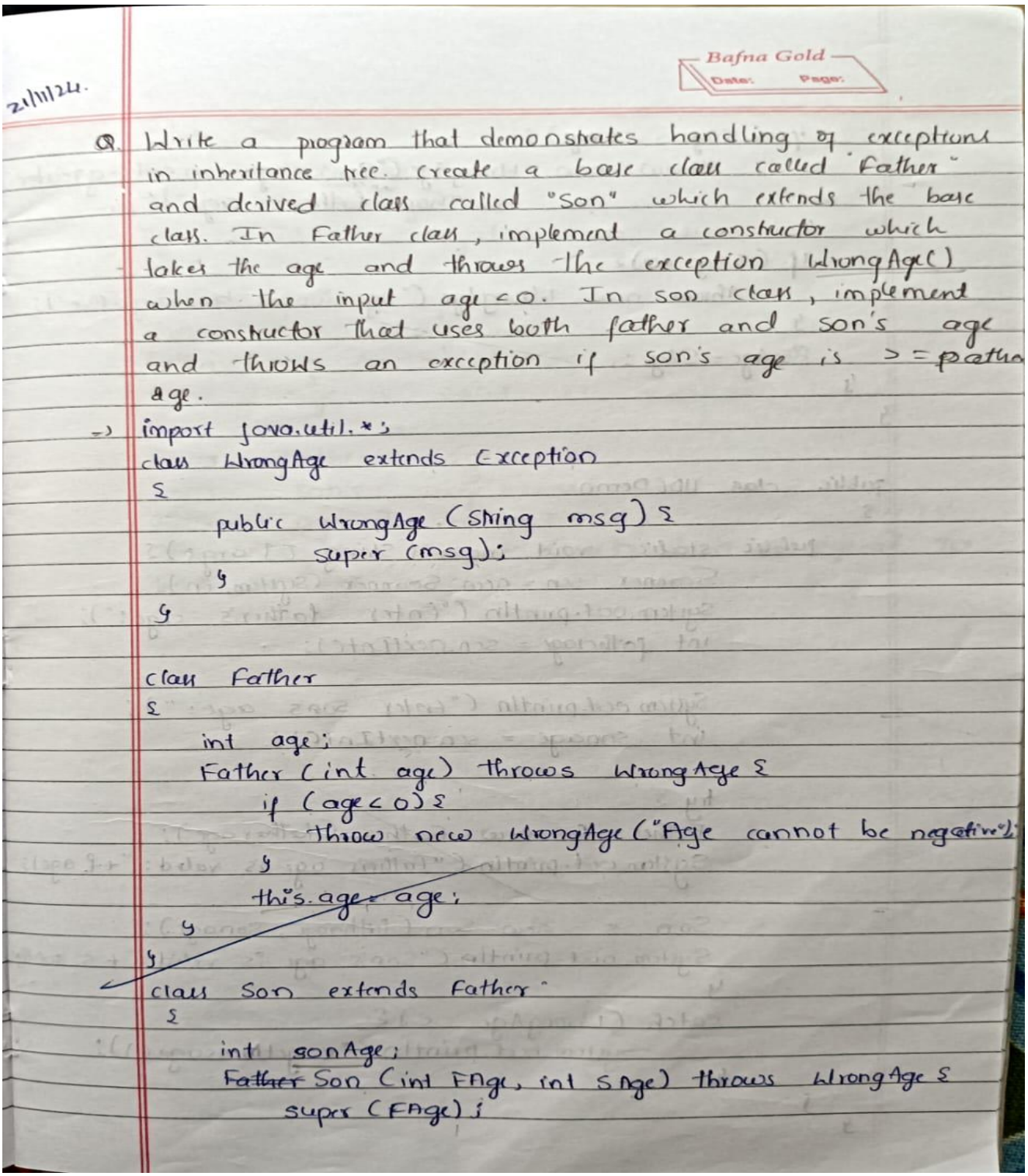
OUTPUT:

```
C:\StudentMarks>javac -d . CIE/Student.java CIE/Internals.java SEE/External.java C_SMain.java

C:\StudentMarks>java C_SMain
Enter number of Students:
2


Enter USN:
1BM23CS100
Enetr Name:
Vishal
Enetr sem:
3
USN:1BM23CS100
Name:Vishal
Sem:3
Enter CIE marks:
45
43
44
42
41
45 43 44 42 41
Enter SEE marks:
8
88
89
90
99
8 88 89 90 99 Final Marks:
49
87
88
87
90
```

```
Enter USN:
1BM23CS200
Enetr Name:
Dhanushree
Enetr sem:
3
USN:1BM23CS200
Name:Dhanushree
Sem:3
Enter CIE marks:
44
45
46
43
42
44 45 46 43 42
Enter SEE marks:
90
95
67
85
74
90 95 67 85 74 Final Marks:
89
92
79
85
79
```

LABORATORY PROGRAM - 7

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



21/11/24.

Bafna Gold

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

=>
```
import java.util.*;
class WrongAge extends Exception
{
    public WrongAge (String msg) {
        super (msg);
    }
}

class Father
{
    int age;
    Father (int age) throws WrongAge {
        if (age < 0) {
            throw new WrongAge ("Age cannot be negative");
        }
        this.age = age;
    }
}
class Son extends Father
{
    int sonAge;
    Father Son (int FAge, int SAge) throws WrongAge {
        super (FAge);
```

```
        if (SAge >= FAge) {
            throw new WrongAge ("Son's age can not greater
                                than or equal father's age");
        }
        if (SAge < 0) {
            throw new WrongAge ("Age cannot be negative");
        }
        SonAge = SAge;
    }
}

public class UDEDemo
{
    public static void main (String [] args) {
        Scanner scn = new Scanner (System.in);
        System.out.println ("Enter father's age:");
        int fatherage = scn.nextInt();

        System.out.println ("Enter son's age:");
        int sonage = scn.nextInt();

        try {
        Father f = new Father (fatherage);
        System.out.println ("Father's age is valid:" +f.age);

        Son s = new Son (fathrage, sonage);
        System.out.println ("Son's age is valid:" + s.sonAg
        }
        catch (WrongAge e) {
            System.out.println (e.getMessage());
        }
    }
}
```

O/P

① Enter father's age : -1
Enter Son's age : 30
Age cannot be negative.

② Enter father age : 45
Enter Son's age : 25
Father's age is valid : 45
Son's age is valid : 25

③ Enter father's age : 45
Enter son's age : 45
Father's age is valid : 45
Son's age ~~no~~ cannot greater than or equal to Father's ~~age~~.

```
import java.util.*;
class WrongAge extends Exception
{
    public WrongAge(String msg){
       super(msg);
    }
}
class Father
{
   int age;
   Father(int age) throws WrongAge{
      if(age<0){
         throw new WrongAge("Age can not be negative");
      }
      this.age=age;
   }
}

class Son extends Father
{
    int SonAge;
    Son(int Fage,int Sage) throws WrongAge{
        super(Fage);
        if(Sage>=Fage){
           throw new WrongAge("Son's age cannot be greater than or equal to Father's age");
        }
        if(Sage<0){
            throw new WrongAge("Age can not be negative");
        }
        SonAge=Sage;
    }
}

public class UDEDemo
{
   public static void main(String [] args){
      Scanner scn=new Scanner(System.in);
      System.out.println("Enter father's age:");
      int FatherAge=scn.nextInt();

      System.out.println("Enter son's age:");
      int SonAge=scn.nextInt();

      try
      {
         Father f=new Father(FatherAge);
         System.out.println("Father's Age is valid:"+ f.age);

         Son s=new Son(FatherAge,SonAge);
         System.out.println("Son's age is valid:"+ s.SonAge);
      }
```

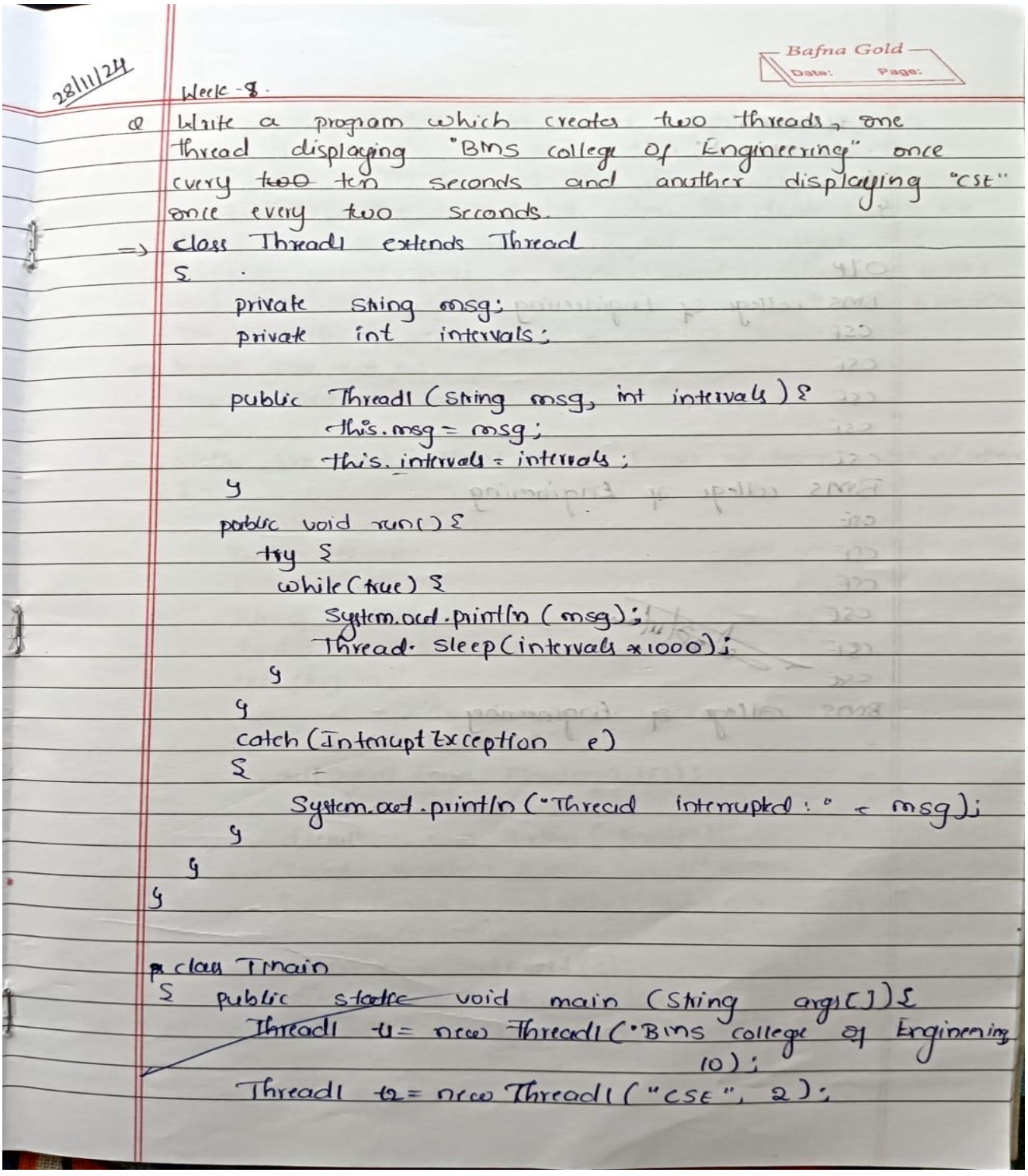
```
    catch(WrongAge e){
        System.out.println(e.getMessage());
    }
  }
}
```

OUTPUT:

```
C:\1BM23CS341>javac UDEDemo.java

C:\1BM23CS341>java UDEDemo
Enter father's age:
55
Enter son's age:
40
Father's Age is valid:55
Son's age is valid:40

C:\1BM23CS341>java UDEDemo
Enter father's age:
50
Enter son's age:
55
Father's Age is valid:50
Son's age cannot be greater than or equal to Father's age

C:\1BM23CS341>java UDEDemo
Enter father's age:
-55
Enter son's age:
45
Age can not be negative

C:\1BM23CS341>java UDEDemo
Enter father's age:
-55
Enter son's age:
-45
Age can not be negative
```

LABORATORY PROGRAM - 8

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



28/11/24

Week - 8.

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

```
=> class Thread1 extends Thread
{
    private String msg;
    private int intervals;

    public Thread1 (String msg, int intervals) {
        this.msg = msg;
        this.intervals = intervals;
    }
    public void run() {
        try {
            while(true) {
                System.out.println (msg);
                Thread.sleep(intervals *1000);
            }
        }
        catch (InterruptException e)
        {
            System.out.println ("Thread interrupted :" + msg);
        }
    }
}

class TMain
{ public static void main (String args[]){
    Thread1 t1 = new Thread1("BMS college of Engineering
                                10);
    Thread1 t2 = new Thread1("CSE", 2);
```

```
        t1.start();
        t2.start();
    }
}
```

O/P

BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
-
-
-

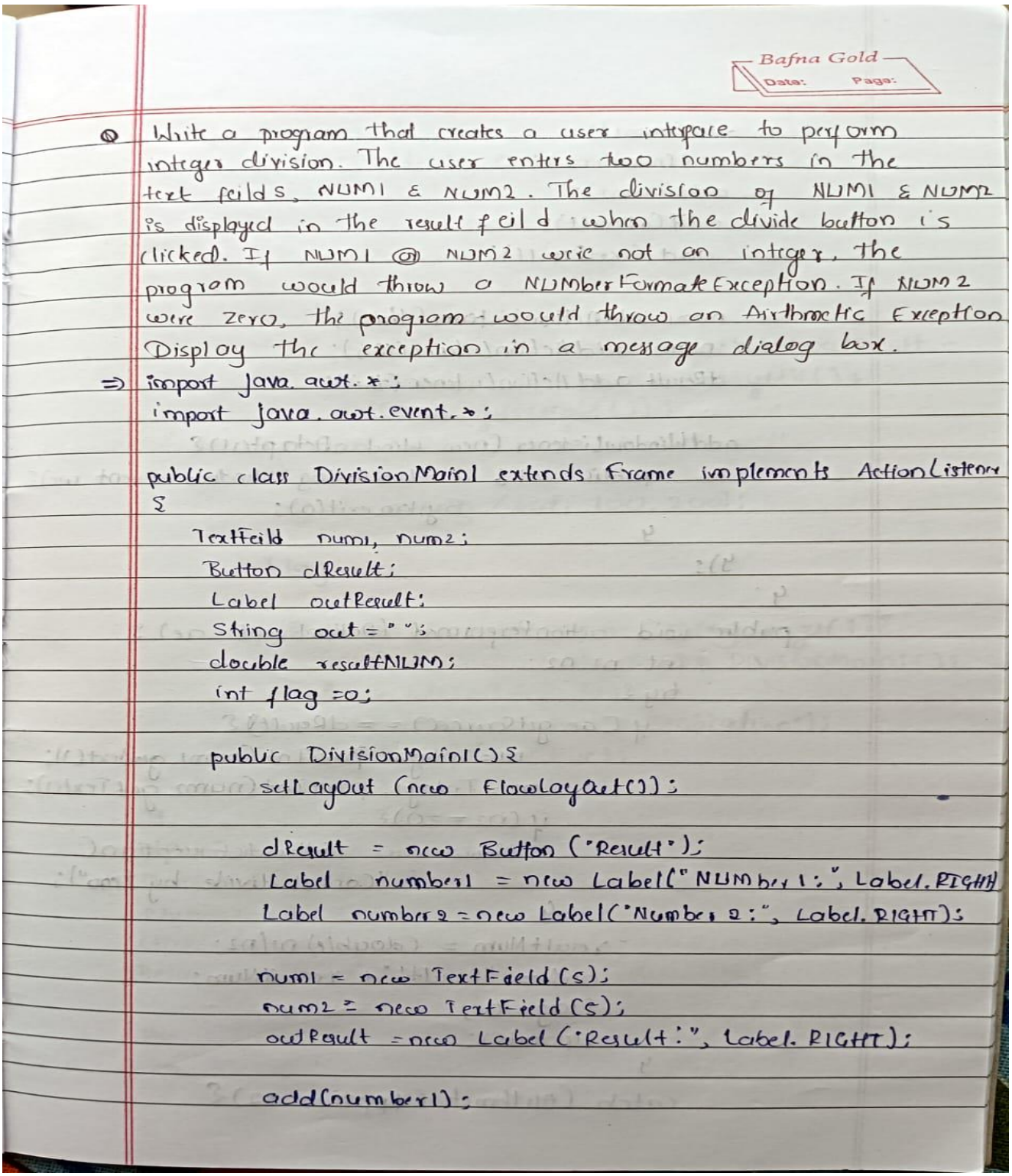
```
class Thread1 extends Thread
{
  private String msg;
  private int intervals;

  Thread1(String msg,int intervals){
     this.msg=msg;
     this.intervals=intervals;
  }

  public void run(){
     try{
        while(true){
           System.out.println(msg);
           Thread1.sleep(intervals*1000);
        }
     }
     catch(InterruptedException e){
        System.out.println("Interrupt Mag:"+msg);
     }
  }
}

class TMain
{
   public static void main(String [] args){
      Thread1 t1=new Thread1("BMS College of Enginneering",10);
      Thread1 t2=new Thread1("CSE",2);

      t1.start();
      t2.start();
   }
}
```

OUTPUT:

```
C:\1BM23CS341>javac TMain.java

C:\1BM23CS341>java TMain
CSE
BMS College of Enginneering
CSE
CSE
CSE
CSE
BMS College of Enginneering
CSE
CSE
CSE
CSE
CSE
BMS College of Enginneering
CSE
CSE
CSE
CSE
CSE
BMS College of Enginneering
CSE
CSE
CSE
CSE
CSE
BMS College of Enginneering
CSE
CSE
CSE
CSE
CSE
BMS College of Enginneering
CSE
CSE
CSE
CSE
CSE
BMS College of Enginneering
```

LABORATORY PROGRAM - 9

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



Bafna Gold
Date: Page:

Q. Write a program that creates a user interface to perform integer division. The user enters two numbers in the text fields, NUM1 & NUM2. The division of NUM1 & NUM2 is displayed in the result field when the divide button is clicked. If NUM1 @ NUM2 were not an integer, the program would throw a NUMberFormatException. If NUM2 were zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

⇒
```
import java.awt.*;
import java.awt.event.*;

public class DivisionMain1 extends Frame implements ActionListener
{
    TextField num1, num2;
    Button dResult;
    Label outResult;
    String out = " ";
    double resultNUM;
    int flag = 0;

    public DivisionMain1()
    {
        setLayOut (new FlowLayout());

        dResult = new Button ("Result");
        Label number1 = new Label("NUMber 1:", Label.RIGHT
        Label number2 = new Label("Number 2:", Label.RIGHT);

        num1 = new TextField(5);
        num2 = new TextField(5);
        outResult = new Label ("Result:", Label.RIGHT);

        add(number1);
```

```
add(num1);
add(number2);
add(num2);
add(dResult);
add(addResult);

num1.addActionListener(this);
num2.addActionListener(this);
dResult.addActionListener(this);

addWindowListener(new WindowAdapter(){
        public void windowClosing(WindowEvent we){
                System.exit(0);
        }
});
}
public void actionPerformed(ActionEvent ae){
        int n1,n2;
        try{
          if(ae.getSource()==dResult){
                n1=Integer.parseInt(num1.getText());
                n2=Integer.parseInt(num2.getText());
                if(n2==0){
                throw new ArithmeticException(
                        "Cannot divide by zero");
                }
          resultNum = (double)n1/n2;
          add="Result:"+resultNum;
          flag=0;
          repaint();
        }
        catch(ArithmeticException e2){
```

```
        flag = 1;
        out = "Divide by 0 Exception!" + e2;
        repaint();
      }
    }
    public void paint(Graphics g){
      if (flag==0){
        d.drawString(out, outResult.getX() + outResult.-
            getWidth(), outResult.getY()+
          outResult.getHeight());
      }
      else{
        g.drawString(out, 100, 200);
      }
      flag = 0;

      public static void main(String args[]){
        DivisionMain1 frame = new DivisionMain1();
        frame.setSize(400, 300);
        frame.setTitle("Division Application");
        frame.setVisible(true);
      }
    }
```

O/P.

1) Number 1 : 10
Number 2 : 5
Click "Result"
Result : 2.0 (Displayed on the GUI)

Ans.

2. Number 1: 10
Number 2: 0
Click "Result"
Divide by 0 Exception!
java.lang.ArithmeticException: Cannot divide by zero

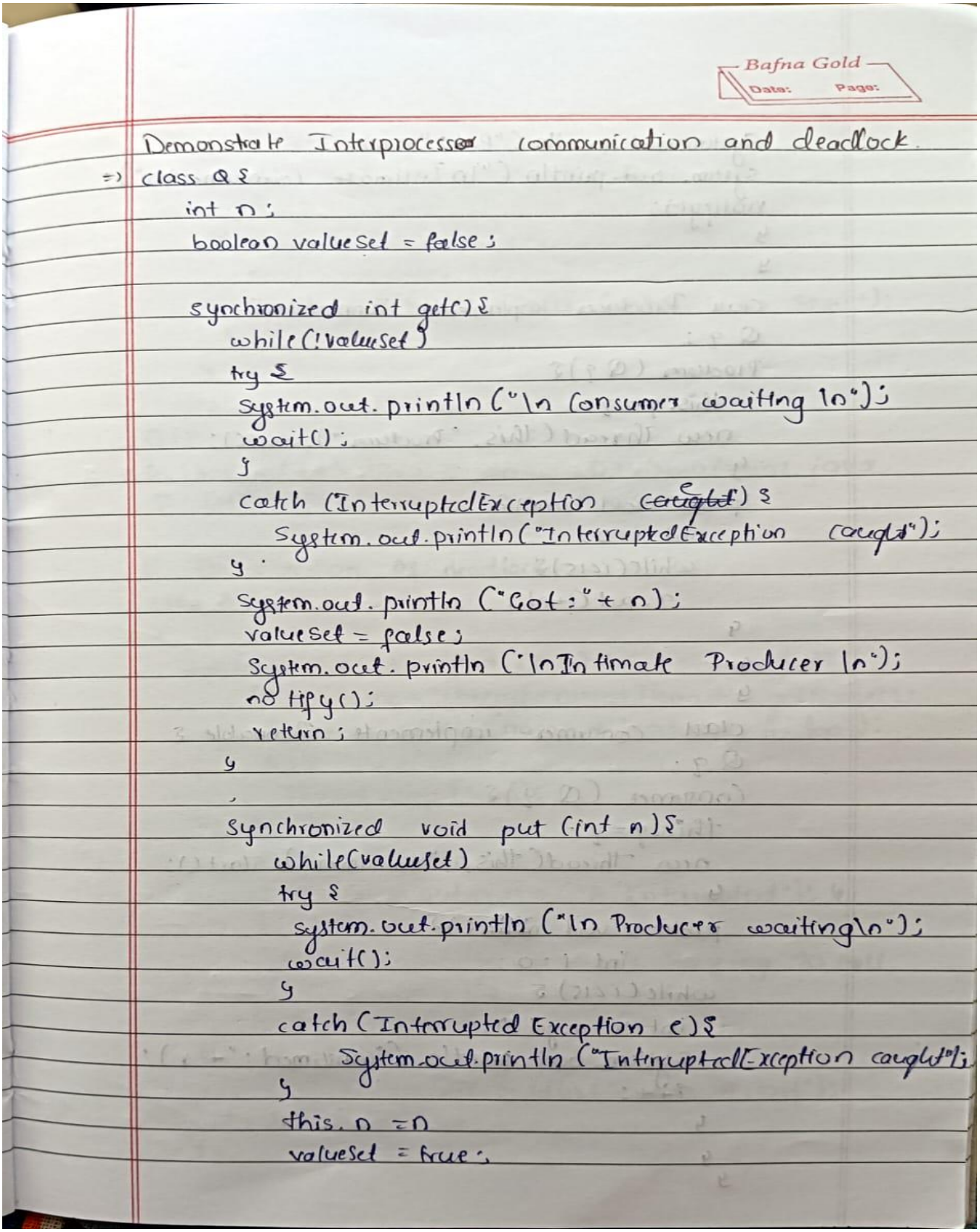
3. Number 1: abc
Number 2: 5
Click "Result"
Number Format Exception!
java.lang.NumberFormatException: For input string: abc

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

public class DivisionMain1 extends Frame implements ActionListener
{
        TextField num1,num2;
        Button dResult;
        Label outResult;
        String out="";
        double
        resultNum; int
        flag=0;

        public DivisionMain1()
        {
                setLayout(new FlowLayout());

                dResult = new Button("RESULT");
                Label number1 = new Label("Number
                1:",Label.RIGHT); Label number2 = new
                Label("Number 2:",Label.RIGHT); num1=new
                TextField(5);
                num2=new TextField(5);
                outResult = new Label("Result:",Label.RIGHT);

                add(number1);
                add(num1);
                add(number2);
                add(num2);
                add(dResult);
                add(outResult);

                num1.addActionListener(this);
                num2.addActionListener(this);
                dResult.addActionListener(this);
                addWindowListener(new WindowAdapter()
                {
                        public void windowClosing(WindowEvent we)
                        {
                                System.exit(0);
                        }
                });
        }
```

```
public void actionPerformed(ActionEvent ae)
{
	int n1,n2;
	try
	{
		if (ae.getSource() == dResult)
		{
			n1=Integer.parseInt(num1.getText());
			n2=Integer.parseInt(num2.getText());

			/*if(n2==0)
				throw new ArithmeticException();*/
			out=n1+" "+n2+" ";
			resultNum=n1/n2;
			out+=String.valueOf(resultNum);
			repaint();

		}
	}
	catch(NumberFormatException e1)
	{
		flag=1;
		out="Number Format Exception! "+e1;
		repaint();
	}
	catch(ArithmeticException e2)
	{
		flag=1;
		out="Divide by 0 Exception! "+e2;
		repaint();
	}

}
public void paint(Graphics g)
{
	if(flag==0)
	g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+ou
	tResult. getHeight()-8);
	else
	g.drawString(out,100,20
	0); flag=0;
}
```

LABORATORY PROGRAM - 9

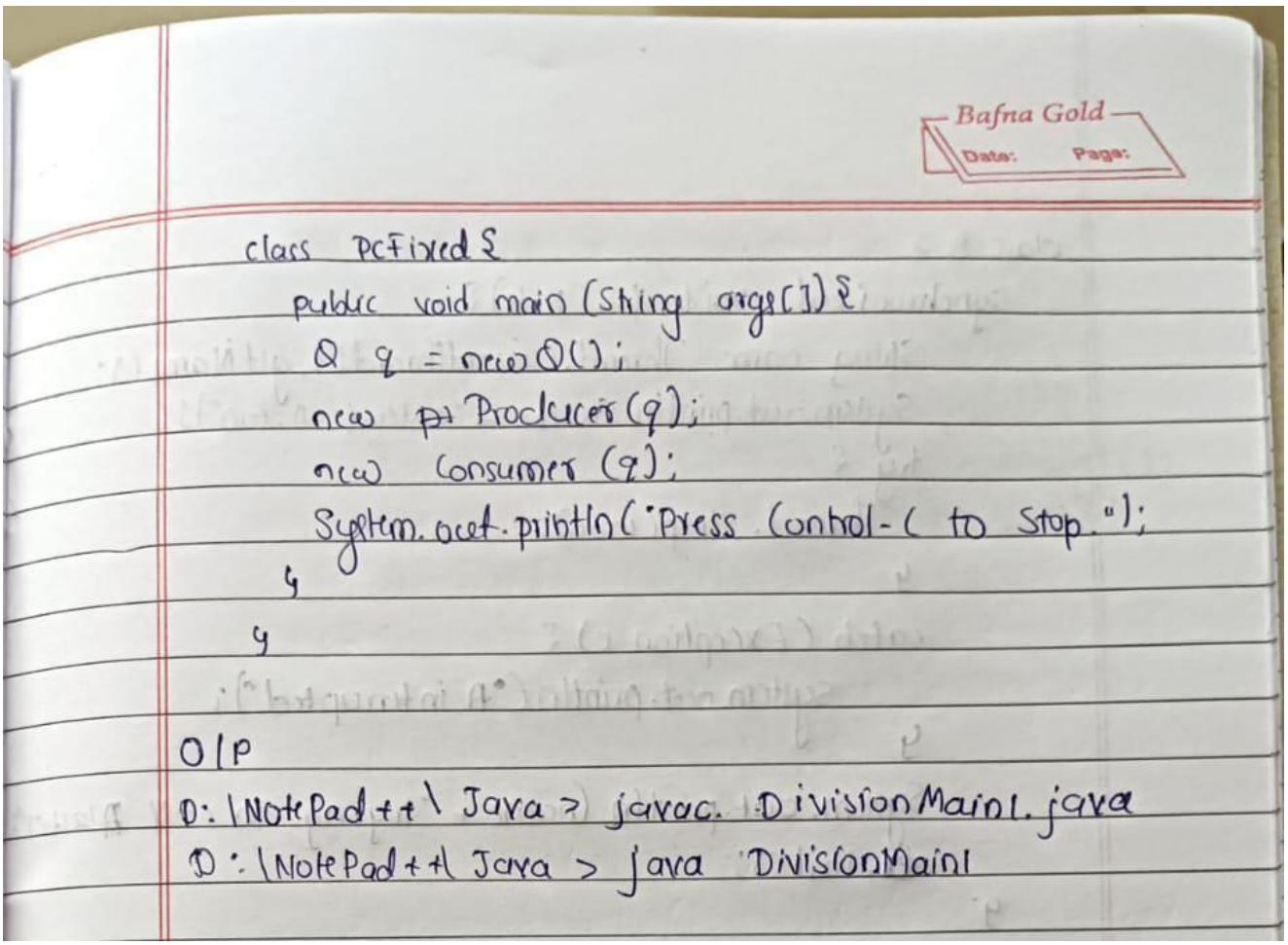
Q).Demonstrate Interprocess communication and deadlock



Demonstrate Interprocesser communication and deadlock.

=>
```
class Q {
    int n;
    boolean valueSet = false;

    synchronized int get() {
      while(!valueset)
      try {
       System.out.println("\n Consumer waiting \n");
       wait();
      }
      catch (InterruptedException e) {
         System.out.println("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("\n Producer waiting\n");
        wait();
       }
       catch(InterruptedException e) {
           System.out.println("InterruptedException caught");
       }
       this.n = n
       valueSet = true;
```

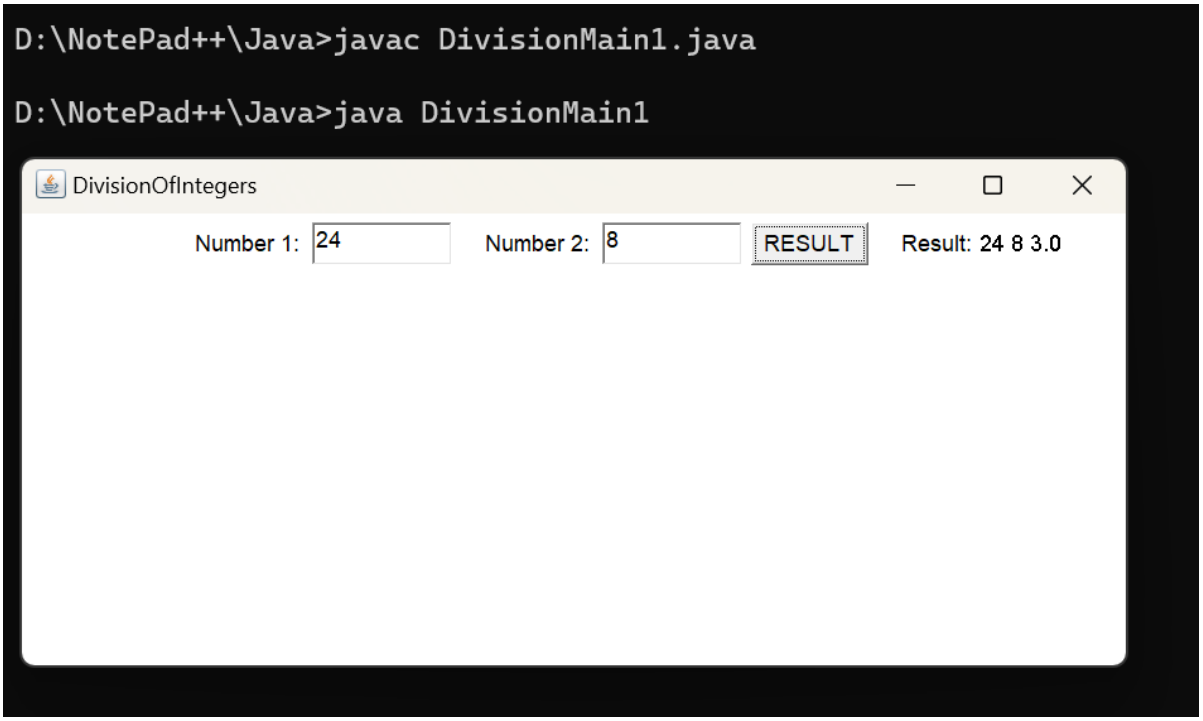
```
System.out.println("Put: "+n);
System.out.println("\n Intimate Consumer\n");
notify();
}
}

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++;
 }
}
}
```



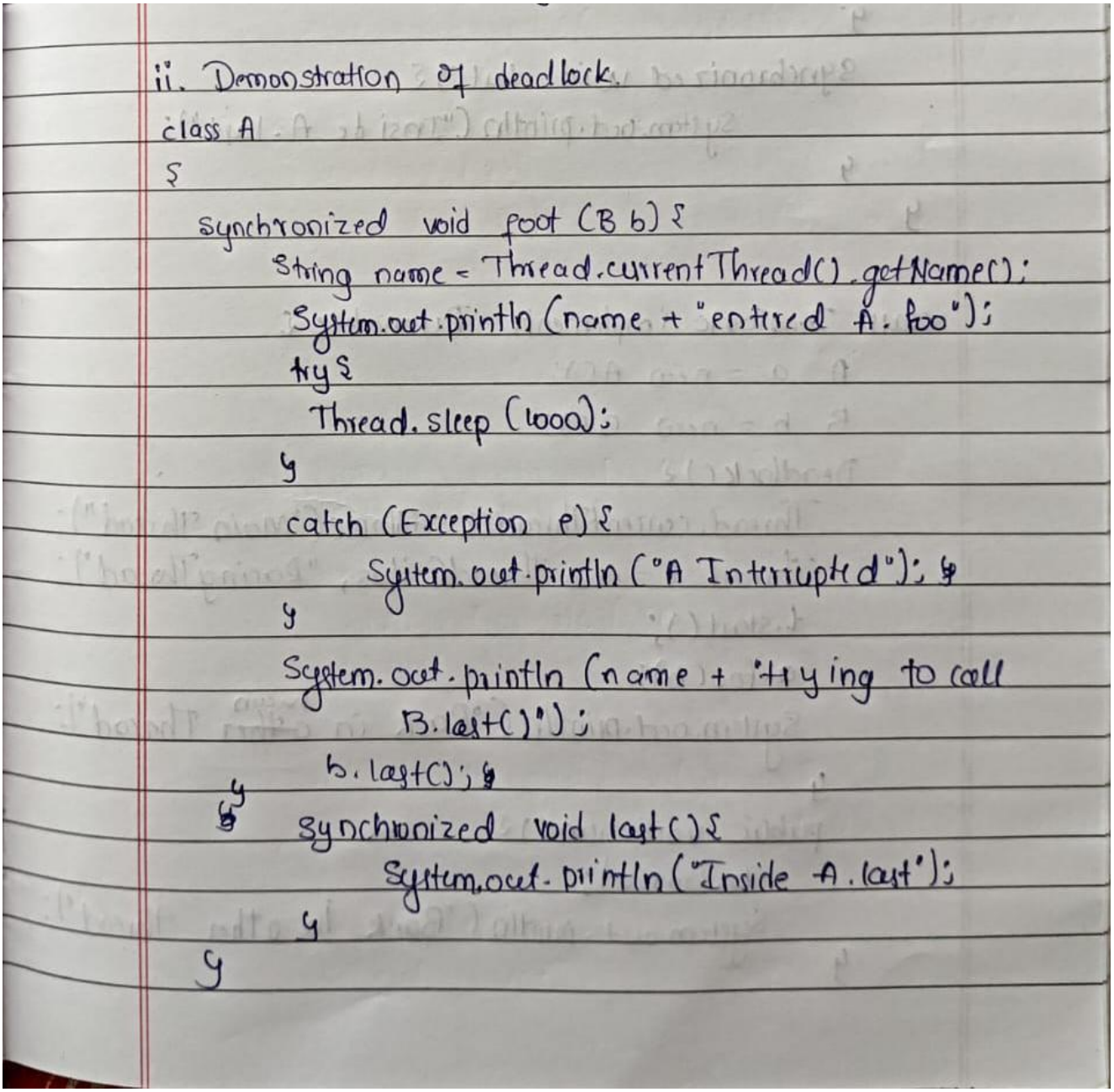

```
class Q {
int n;
boolean valueSet = false;

synchronized int get() {
while(!valueSet)
try {
System.out.println("\nConsumer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
System.out.println("Got: " + n);
valueSet = false;
System.out.println("\nIntimate Producer\n");
notify();
return n;
}

synchronized void put(int n) {
while(valueSet)
try {
System.out.println("\nProducer waiting\n");
```

```
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}

class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}
}

class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
}
public void run() {
        int i=0;
while(i<15) {
int r=q.get();
System.out.println("consumed:"+r);
i++;
}
}
}

class PCFixed {
public static void main(String args[]) {
Q q = new Q();
new Producer(q);
new Consumer(q);
System.out.println("Press Control-C to stop.");
}
}
```

OUTPUT:


D:\NotePad++\Java>javac DivisionMain1.java
D:\NotePad++\Java>java DivisionMain1
DivisionOfIntegers
Number 1:
24
Number 2:
8
RESULT
Result: 24 8 3.0

ii. Demonstration of deadlock



```
ii. Demonstration of deadlock.
class A
{
  synchronized void foo (B b) {
      String name = Thread.currentThread().getName();
      System.out.println (name + "entered A.foo");
      try {
       Thread.sleep (1000);
      }
      catch (Exception e) {
          System.out.println ("A Interrupted"); }
      }
      System.out.println (name + "trying to call
          B.last()");
       b.last(); }
  }
  synchronized void last() {
      System.out.println ("Inside A.last");
  }
}
```

```
class B {
  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 A.last()");
      a.last();
  }
  synchronized void last() {
          System.out.println("Inside A.last");
  }
}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();
    Deadlock() {
      Thread.currentThread().setName("Main Thread");
      Thread t = new Thread(this, "RacingThread");
      t.start();
      a.foo(b);
      System.out.println("Back in main Thread");
    }
    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }
```

```
public static void main(String args[]){
        new Deadlock();
}
public static void main(String args[]){
        DivisionMain1 dm = new DivisionMain1();
        dm.setSize(new Dimension(800,400));
        dm.setTitle("DivisionOfIntegers");
        dm.setVisible(true);
}
}
```

```
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(); }
     synchronized 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(); }
synchronized 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(); }
}
  public static void main(String[] args)
  {
      DivisionMain1 dm=new DivisionMain1();
      dm.setSize(new Dimension(800,400));
      dm.setTitle("DivisionOfIntegers");
      dm.setVisible(true);
   }
}
```