VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming (23CS3PCOOJ)

Submitted by

Abhiram rayadurgam (1BM23CS011)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Sep-2024 to Jan-2025

B.M.S. College of Engineering,

Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "Object Oriented Java Programming (23CS3PCOOJ)" carried out by Abhiram Rayadurgam **(1BM23CS011),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

Geetha N Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

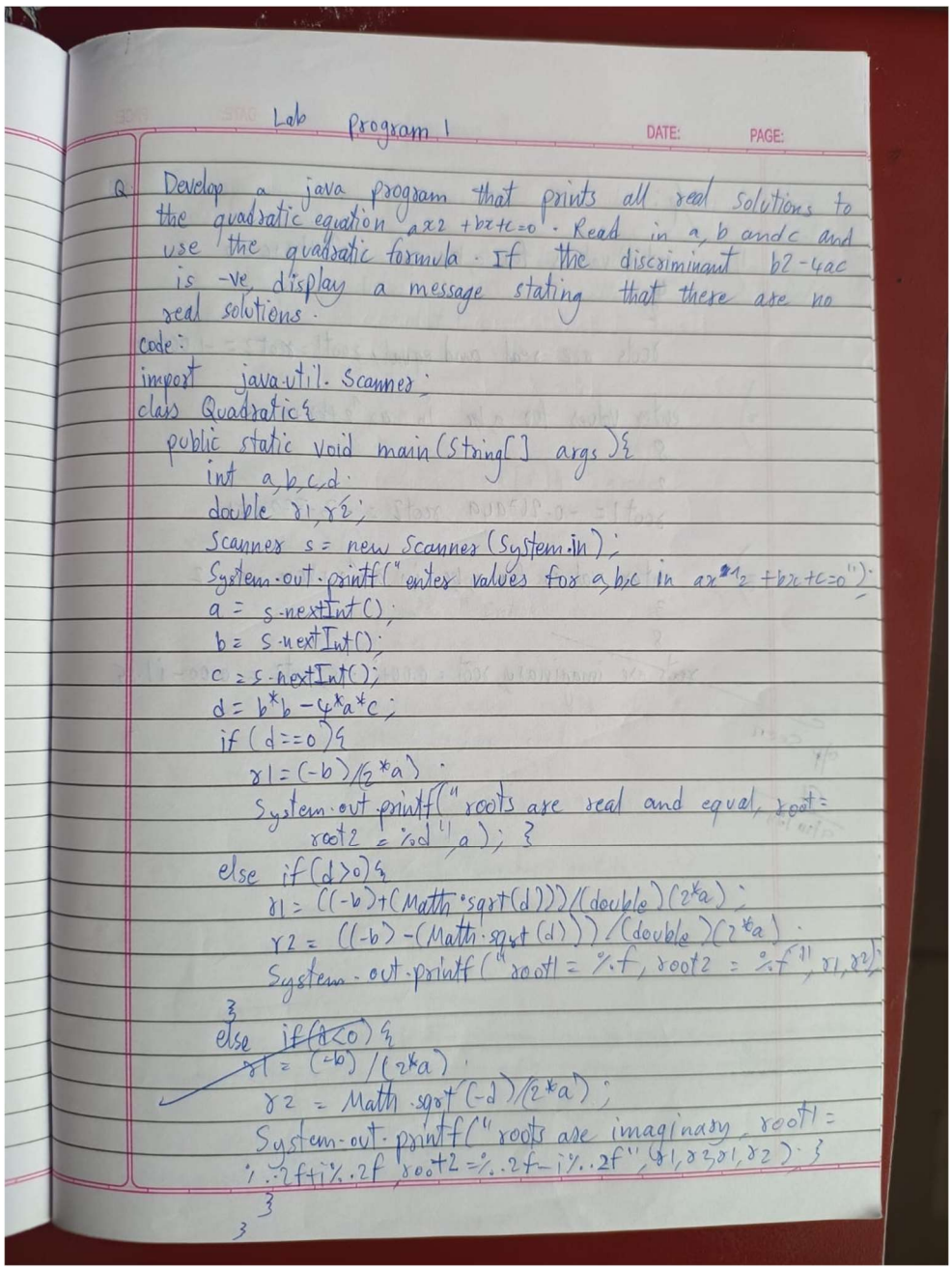
Index

Github Link:
https://github.com/Abhiram-Rayadurgam/Java/tree/main/record

Program 1

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

Observation:

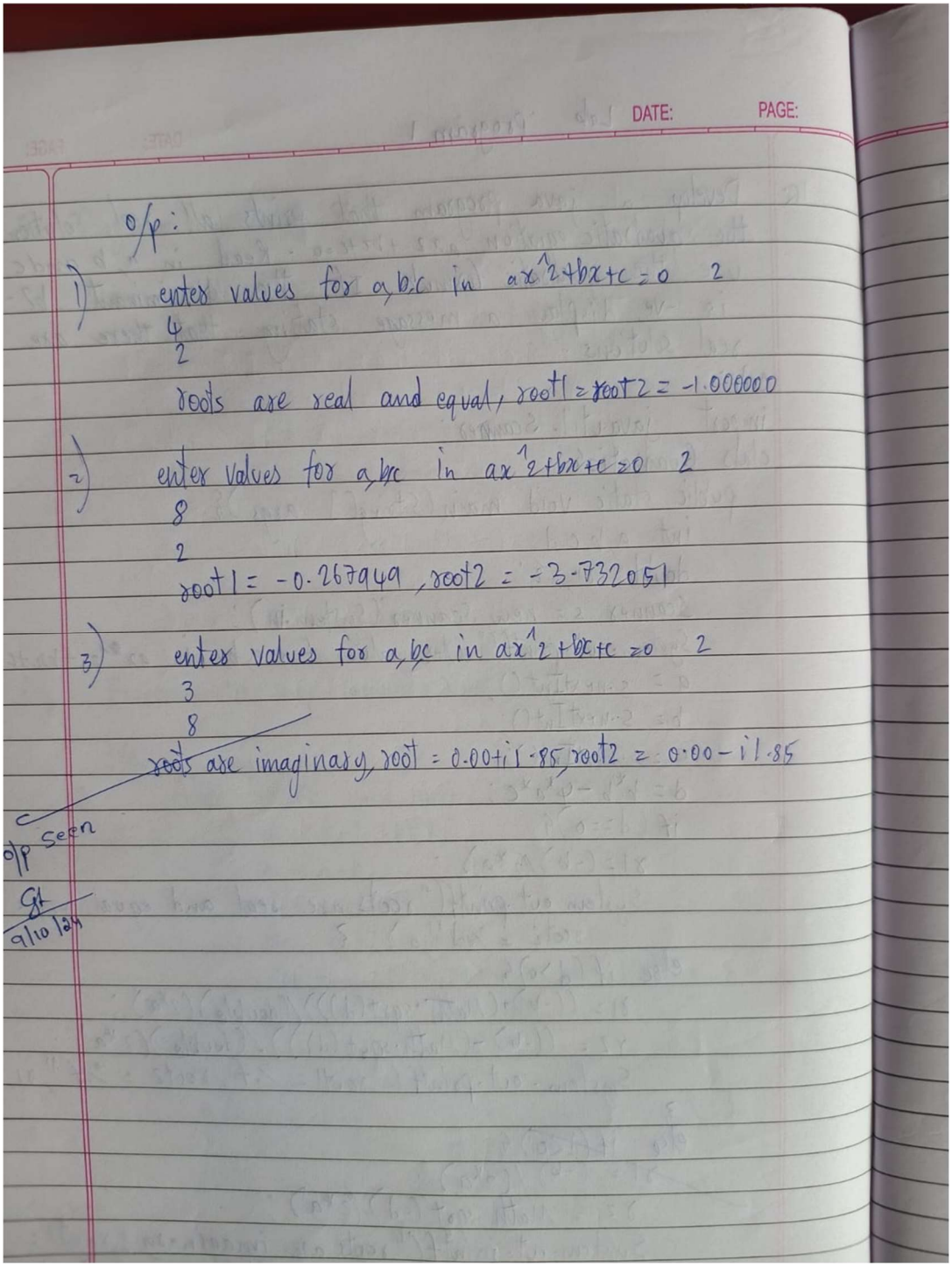


Lab Program 1 DATE: PAGE:

Q Develop a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b and c and use the quadratic formula. If the discriminant b^2-4ac is -ve, display a message stating that there are no real solutions.

code:

```
import java.util.Scanner;
class Quadratic{
  public static void main(String[] args){
    int a,b,c,d;
    double r1,r2;
    Scanner s = new Scanner(System.in);
    System.out.printf("enter values for a,b,c in ax**2+bx+c=0");
    a = s.nextInt();
    b = s.nextInt();
    c = s.nextInt();
    d = b*b-4*a*c;
    if(d==0){
      r1=(-b)/(2*a);
      System.out.printf("roots are real and equal, root=
        root2 = %d",a); }
    else if(d>0){
      r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
      r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
      System.out.printf("root1 = %.f, root2 = %f",r1,r2);
    }
    else if(d<0){
      r1 = (-b)/(2*a);
      r2 = Math.sqrt(-d)/(2*a);
      System.out.printf("roots are imaginary root1 =
      %.2f+i%.2f root2 = %.2f-i%.2f",r1,r2,r1,r2); }
  }
}
```


DATE:
PAGE:
o/p:
1) enter values for a,b,c in ax^2+bx+c=0 2
4
2
roots are real and equal, root1 = root2 = -1.000000
2) enter values for a,b,c in ax^2+bx+c=0 2
8
2
root1 = -0.267949, root2 = -3.732051
3) enter values for a,b,c in ax^2+bx+c=0 2
3
8
roots are imaginary, root1 = 0.00+i1.85, root2 = 0.00-i1.85
o/p seen
9/10/24

Code:

```
//USN:1BM23CS011
//name:Abhiram Rayadurgam
import java.util.Scanner;

class Quadratic{
 public static void main(String[] args){
 int a,b,c,d;
 double r1,r2;
 Scanner s = new Scanner(System.in);
 System.out.printf("enter values for a,b,c in ax^2+bx+c=0");
 a=s.nextInt();
 b=s.nextInt();
 c=s.nextInt();
 d=b*b-4*a*c;
 if(d==0){
 r1=(-b)/(2*a);
 System.out.printf("roots are real and equal, root1 = root2 = %f",r1);
 }
 else if(d>0){
 r1=((-b) + (Math.sqrt(d)))/(double)(2*a);
 r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
 System.out.printf("root1 = %f,root2 = %f",r1,r2);
 }
 else if(d<0){
 r1 = (-b)/(2*a);
 r2 = Math.sqrt(-d)/(2*a);
 System.out.printf("roots are imaginary, root1 = %.2f+i%.2f,root2 = %.2f-i%.2f",r1,r2,r1,r2);
 }
 }
```

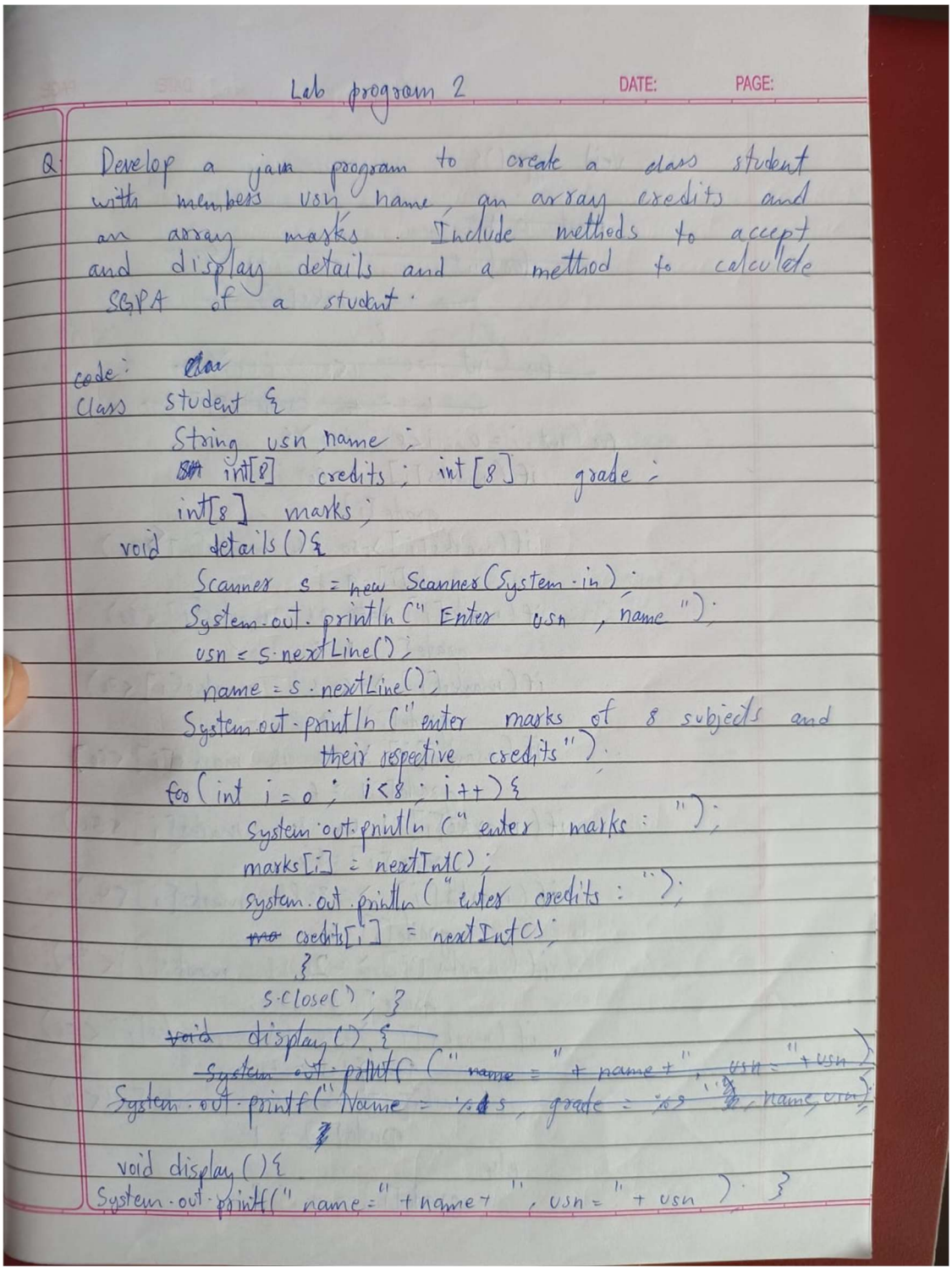
Output:

```
D:\>cd 1bm23cs011

D:\1BM23CS011>javac Quadratic.java

D:\1BM23CS011>java Quadratic
enter values for a,b,c in ax^2+bx+c=02
4
2
roots are real and equal, root1 = root2 = -1.000000
D:\1BM23CS011>java Quadratic
enter values for a,b,c in ax^2+bx+c=02
8
2
root1 = -0.267949,root2 = -3.732051
D:\1BM23CS011>java Quadratic
enter values for a,b,c in ax^2+bx+c=02
3
8
roots are imaginary, root1 = 0.00+i1.85,root2 = 0.00-i1.85
D:\1BM23CS011> name: Abhiram Rayadurgam
```

Program 2

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.

Observation:



Lab program 2 DATE: PAGE:

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.

code:

```
Class student {
    String usn, name;
    int[8] credits; int[8] grade;
    int[8] marks;
void details(){
    Scanner s = new Scanner(System.in);
    System.out.println("Enter usn, name");
    usn = s.nextLine();
    name = s.nextLine();
    System.out.println("enter marks of 8 subjects and
            their respective credits");
    for(int i=0; i<8; i++){
        System.out.println("enter marks:");
        marks[i] = nextInt();
        System.out.println("enter credits:");
        credits[i] = nextInt();
        }
        s.close(); }
void display(){
    System.out.printf("name = " + name + ", usn = " + usn);
System.out.printf("Name = %s, grade = %s", name, usn);
}
void display(){
System.out.printf("name=" + name + ", usn=" + usn); }
```

DATE: PAGE:

```
void sgpa(){
float a,b;
for(int i=0; i<8 ;i++){
    a += marks[i] ;
}
for(int i=0; i<8; i++){
    b += credits[i];
for(int i=0; i<8 ; i++){
    if(marks[i] >=90)
        grade[i] = 10 ;
    if(marks[i]>=80 && marks[i]<90)
        grade[i]=9 ;
    if(marks[i] >=70 && marks[i] <80)
        grade[i]=8 ;
    if(marks[i] >=60 && marks[i] <70)
        grade[i] =7 ;
    if(marks[i] >=50 && marks[i] <60)
        grade[i] = 6;
    if(marks[i] >=40 && marks[i] <50)
        grade[i] =5;
    if(marks[i] >=30 && marks[i] <40)
        grade[i]=4;
    if(marks[i] >=20 && marks[i] <30)
        grade[i] = 3 ;
    if(marks[i] >=10 && marks[i] <20)
        grade[i] = 2 ;
    if(marks[i] >=0 && marks[i] <10)
        grade[i] = 1;
    else
        grade[i] =0 ;}
```

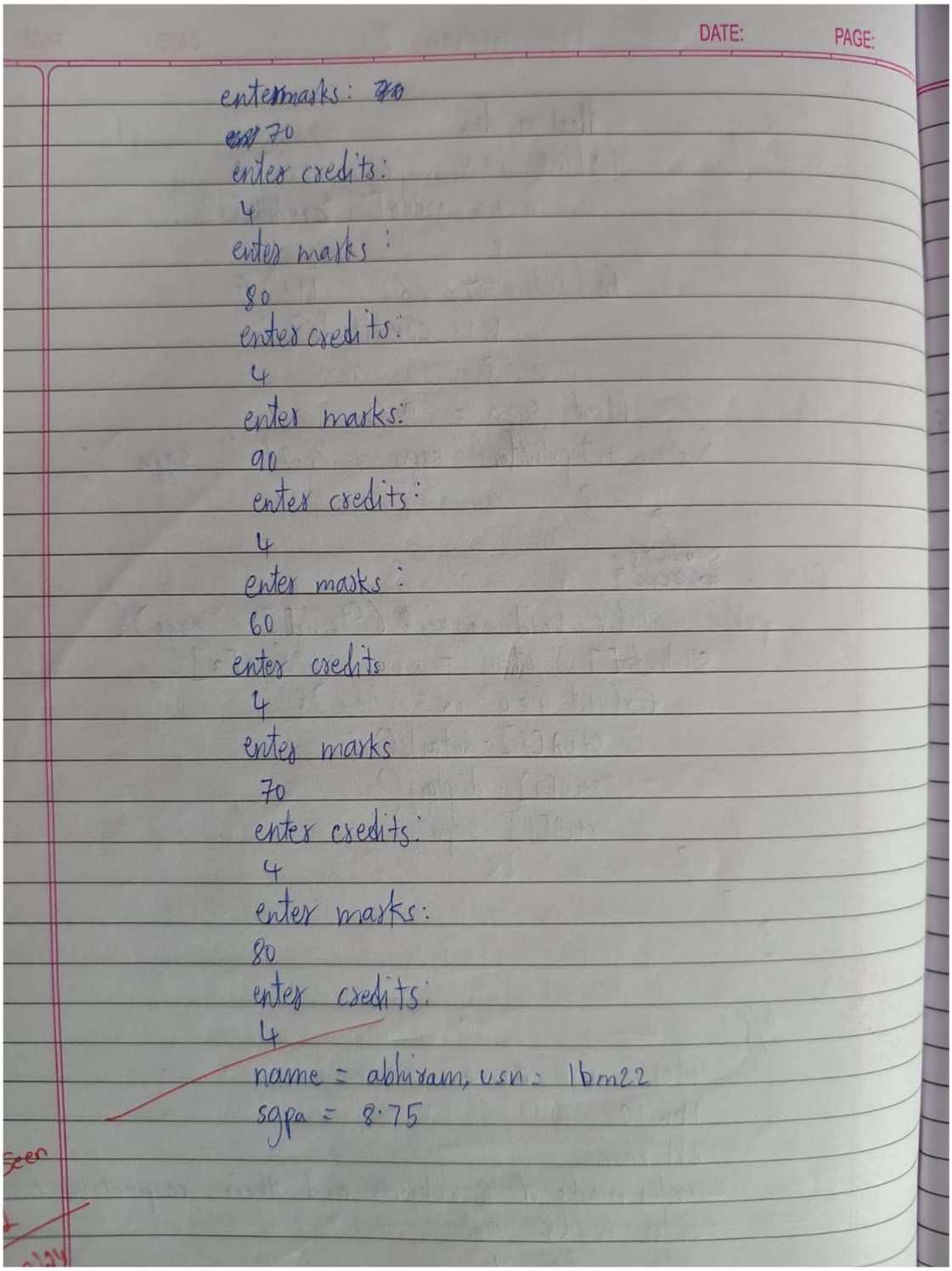
DATE: PAGE:

```
        float a,b;
        for(int i=0; i<8; i++){
            a += grade[i] * credits[i];
        }
        for(int i=0; i<8; i++){
            b += credits[i];
        }
        float sgpa = a/b;
        System.out.printf("sgpa = %.2f", sgpa);
    }
}
Class students{
    public static void main(String[] args){
        Student[] stud = new student[3];
        for(int i=0; i<3; i++){
            stud[i].details();
            stud[i].display();
            stud[i].sgpa();
        }
    }
}
```

Seen
execute

O/P:

```
enter usn, name:
1bm22
abhiram
enter marks of 8 subjects and their respective credits:
enter marks: 90
enter credits: 4
enter marks: 80
enter credits: 4
```


DATE:
PAGE:
enter marks: 70
70
enter credits:
4
enter marks:
80
enter credits:
4
enter marks:
90
enter credits:
4
enter marks:
60
enter credits
4
enter marks
70
enter credits:
4
enter marks:
80
enter credits:
4
name = abhiram, usn = 1bm22
sgpa = 8.75

Code:

```
//USN:1BM23CS011
//name:Abhiram Rayadurgam

import java.util.Scanner;

class Student {
 String usn, name;
 int[] credits = new int[8];
 int[] grade = new int[8];
 int[] marks = new int[8];

 void details(Scanner s) {
 System.out.println("enter name, usn:");
 usn = s.next();
 name = s.next();
 System.out.println("enter marks of 8 subjects and their respective credits:");
 for (int i = 0; i < 8; i++) {
 System.out.println("enter marks:");
 marks[i] = s.nextInt();
 System.out.println("enter credits:");
 credits[i] = s.nextInt();
 }
 }

 void display() {
 System.out.println("name = " + name + ", usn = " + usn);
 }

 void grades() {
 for (int i = 0; i < 8; i++) {
 if (marks[i] >= 90){
 grade[i] = 10;}
 else if (marks[i] >= 80 && marks[i] < 90){
 grade[i] = 9;}
 else if (marks[i] >= 70 && marks[i] < 80){
 grade[i] = 8;}
 else if (marks[i] >= 60 && marks[i] < 70){
 grade[i] = 7;}
 else if (marks[i] >= 50 && marks[i] < 60){
 grade[i] = 6;}
 else if (marks[i] >= 40 && marks[i] < 50){
 grade[i] = 5;}
 else if (marks[i] >= 30 && marks[i] < 40){
 grade[i] = 4;}
 else if (marks[i] >= 20 && marks[i] < 30){
 grade[i] = 3;}
 else if (marks[i] >= 10 && marks[i] < 20){
 grade[i] = 2;}
 else if (marks[i] >= 0 && marks[i] < 10){
 grade[i] = 1;}
 else{
 grade[i] = 0;}
 }
```

```
 }

 void sgpa() {

 float a = 0, b = 0;
 for (int i = 0; i < 8; i++) {
 a = a + grade[i] * credits[i];
 }
 for (int i = 0; i < 8; i++) {
 b = b + credits[i];
 }
 float sgpa = a / b;
 System.out.printf("sgpa = %.2f", sgpa);
 System.out.println(" ");
 }
}

class Students {
 public static void main(String[] args) {
 Scanner s = new Scanner(System.in);
 Student[] stud = new Student[3];
 for (int i = 0; i < 3; i++) {
 stud[i] = new Student();
 stud[i].details(s);
 stud[i].display();
 stud[i].grades();
 stud[i].sgpa();
 }
 }
}
```

Output:

```
D:\1BM23CS011>javac Students.java

D:\1BM23CS011>java Students
enter name, usn:
1bm23
abhi
enter marks of 8 subjects and their respective credits:
enter marks:
90
enter credits:
4
enter marks:
80
enter credits:
4
enter marks:
70
enter credits:
4
enter marks:
90
enter credits:
4
enter marks:
66
enter credits:
4
enter marks:
88
enter credits:
4
enter marks:
77
```

```
enter credits:
4
enter marks:
99
enter credits:
4
name = abhi, usn = 1bm23
sgpa = 8.88
enter name, usn:
1bm22
ariz
enter marks of 8 subjects and their respective credits:
enter marks:
90
enter credits:
4
enter marks:
99
enter credits:
4
enter marks:
80
enter credits:
4
enter marks:
100
enter credits:
4
enter marks:
99
enter credits:
4
enter marks:
90
enter credits:
4
enter marks:
98
enter credits:
```

```
4
enter marks:
89
enter credits:
4
name = ariz, usn = 1bm22
sgpa = 9.75
enter name, usn:
1bm21
arnav
enter marks of 8 subjects and their respective credits:
enter marks:
90
enter credits:
4
enter marks:
80
enter credits:
4
enter marks:
70
enter credits:
4
enter marks:
85
enter credits:
4
enter marks:
90
enter credits:
4
enter marks:
99
enter credits:
4
enter marks:
90
enter credits:
4
```

```
enter marks:
70
enter credits:
4
name = arnav, usn = 1bm21
sgpa = 9.25

D:\1BM23CS011>Name: Abhiram, USN :1BM23CS011
```

Program 3

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

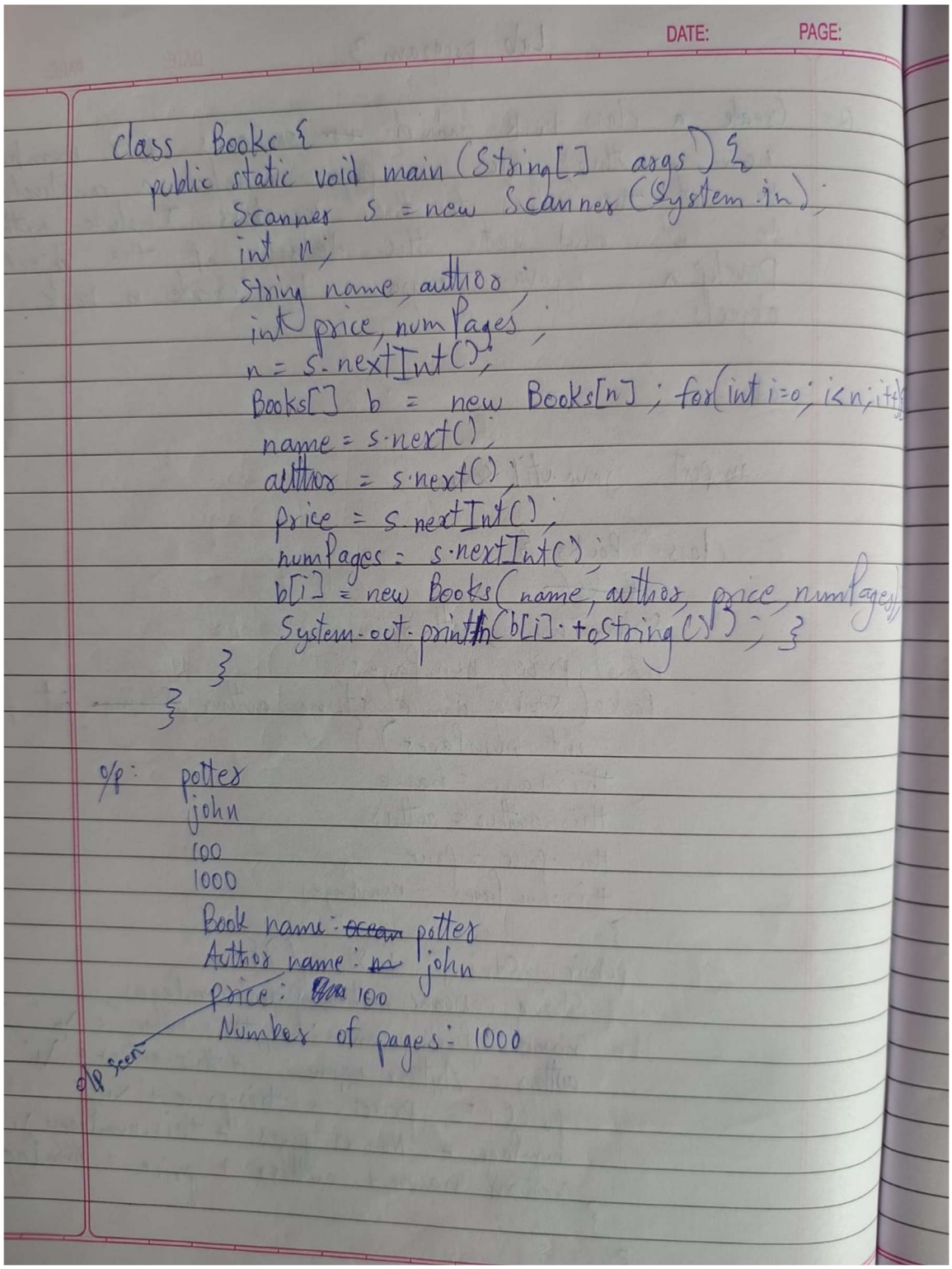
Observation:

Lab program 3 DATE: PAGE:

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. Develop a java program to create n book objects.

code:

```
import java.util.Scanner;

class Books {
    String name, author;
    int price, numPages;
    Books(String name, String author, int price,
          int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString() {
        String name, author, price, numPages;
        name = "Book name: " + this.name + "\n";
        author = "Author name: " + this.author + "\n";
        price = "price: " + this.price + "\n";
        numPages = "No. of pages: " + this.numPages + "\n";
        return name + author + price + numPages;
    }
}
```



```
class Books {
    public static void main (String[] args) {
        Scanner s = new Scanner (System.in);
        int n;
        String name, author;
        int price, numPages;
        n = s.nextInt();
        Books[] b = new Books[n]; for(int i=0; i<n; i++){
        name = s.next();
        author = s.next();
        price = s.nextInt();
        numPages = s.nextInt();
        b[i] = new Books(name, author, price, numPages);
        System.out.println(b[i].toString()); }
    }
}
```

o/p: potter
john
100
1000
Book name: ~~ocean~~ potter
Author name: ~~no~~ john
~~price~~: ~~0~~ 100
Number of pages: 1000

o/p seen

Code:

```
//USN:1BM23CS011
//name:Abhiram Rayadurgam
import java.util.Scanner;

class Books {
 String name;
 String author;
 int price;
 int numPages;

 Books(String name, String author, int price, int numPages) {
 this.name = name;
 this.author = author;
 this.price = price;
 this.numPages = numPages;

 }

 public String toString() {
 String name, author, price, numPages;
 name = "Book name: " + this.name + "\n";
 author = "Author name: " + this.author + "\n";
 price = "Price: " + this.price + "\n";
 numPages = "Number of pages: " + this.numPages + "\n";
 return name + author + price + numPages;
 }

}

class Bookc {
 public static void main(String[] args) {
 Scanner s = new Scanner(System.in);
 int n, price, numPages;
 String author, name;
 n = s.nextInt();
 Books[] b = new Books[n];
 for (int i = 0; i < n; i++) {
 name = s.next();
 author = s.next();
 price = s.nextInt();
 numPages = s.nextInt();
 b[i] = new Books(name, author, price, numPages);
 System.out.println(b[i].toString());
 }
 s.close();

 }
}
```

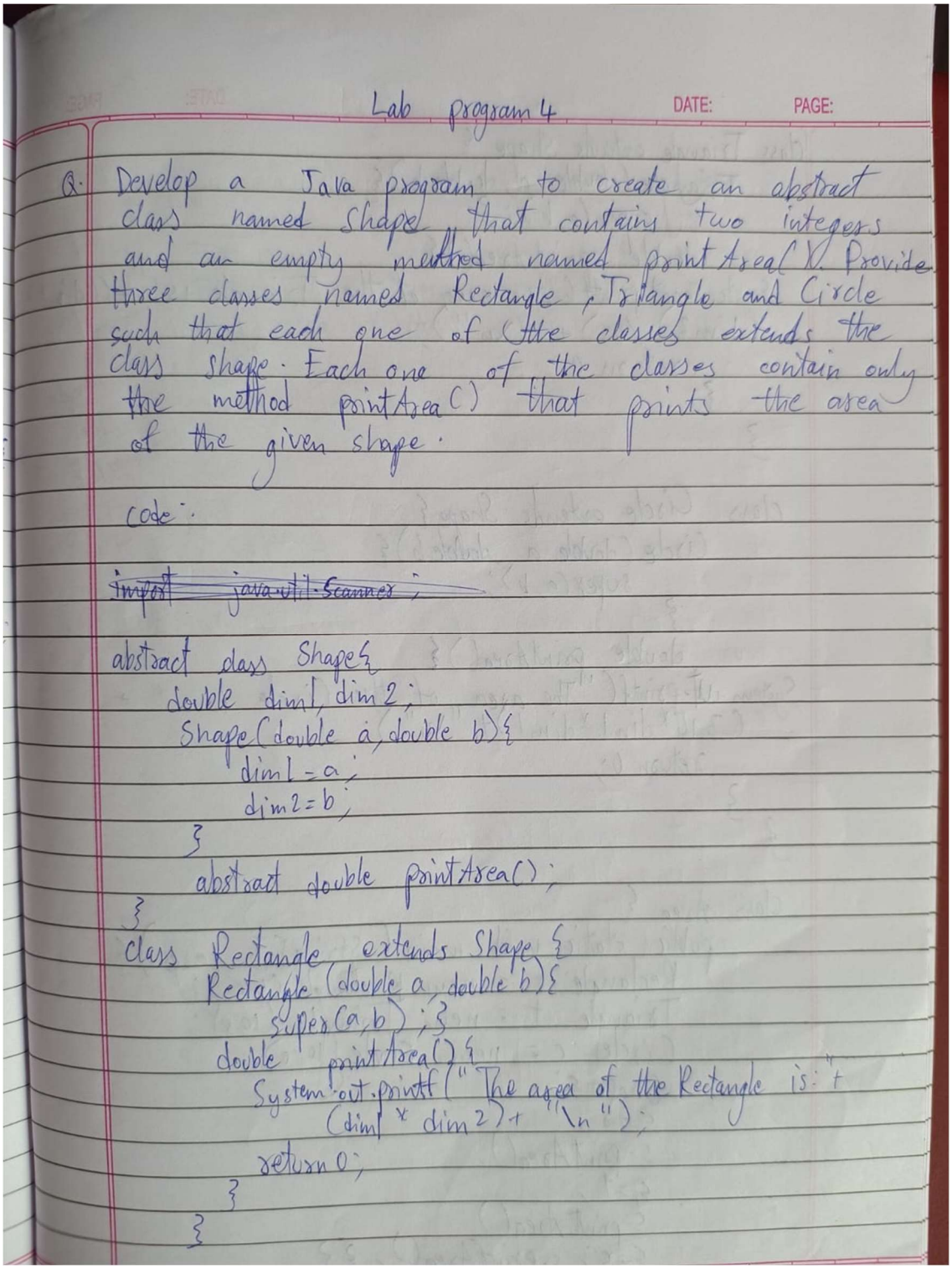
Output:

```
D:\1BM23CS011>java Bookc
3
potter
john
100
1000
Book name: potter
Author name: john
Price: 100
Number of pages: 1000

ocean
mary
50
500
Book name: ocean
Author name: mary
Price: 50
Number of pages: 500

realism
joseph
300
5000
Book name: realism
Author name: joseph
Price: 300
Number of pages: 5000


D:\1BM23CS011>Name: Abhiram Rayadurgam, USN: 1BM23CS011
```

Program 4

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

Observation:



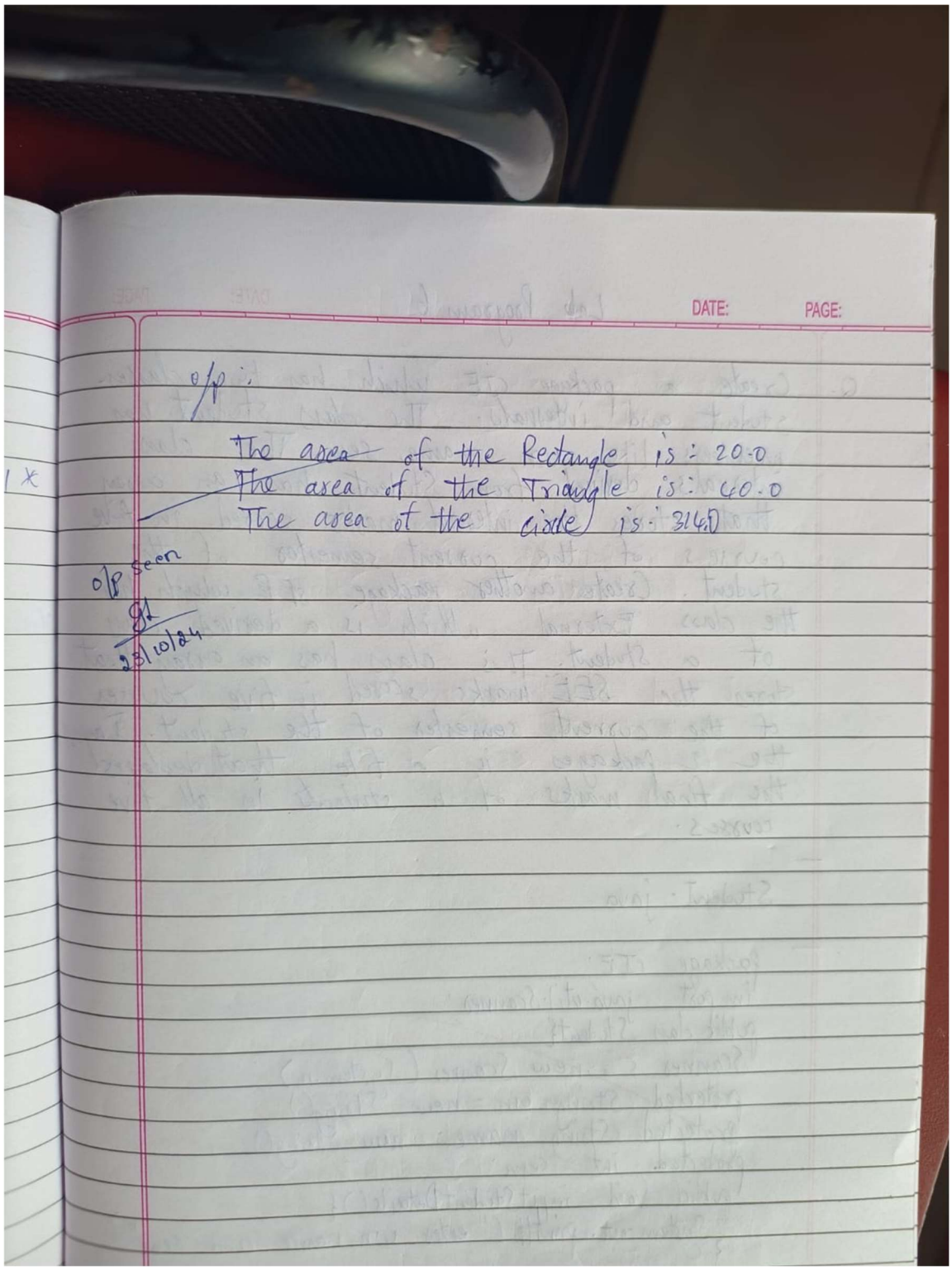
Lab program 4 DATE: PAGE:

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.

Code:

```
~~import java.util.Scanner;~~

abstract class Shape{
    double dim1, dim2;
    Shape(double a, double b){
        dim1 = a;
        dim2 = b;
    }
    abstract double printArea();
}
class Rectangle extends Shape {
    Rectangle (double a, double b){
        super(a,b);}
    double printArea(){
        System.out.printf("The area of the Rectangle is: "+
            (dim1 * dim2)+ "\n");
        return 0;
    }
}
```

```
class Triangle extends Shape {
    Triangle (double a, double b) {
        super(a,b); }
        double printArea() {
System.out.printf("The area of the Triangle is: " + ((dim1 *
dim2) /2) + "\n");
        return 0;
    }
}

class Circle extends Shape {
    Circle (double a, double b) {
        super(a,b);
    }
    double printArea() {
System.out.printf("The area of the Circle is: " +
(3.14 * dim1 * dim1) + "\n");
    return 0;
    }
}

class Area {
    public static void main (String[] args) {
        Rectangle r = new Rectangle (4,5);
        Triangle t = new Triangle (10,8);
        Circle c = new Circle (10,0);
        Shape s;
        s = r;
        s.printArea();
        s = t;
        s.printArea();
        s = c; s.printArea(); }}
```

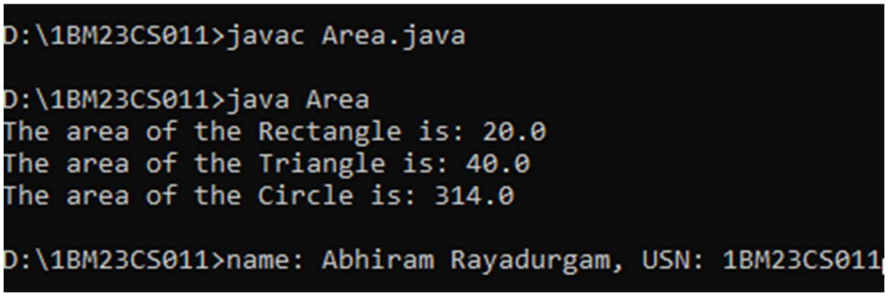

DATE:
PAGE:
o/p:
The area of the Rectangle is: 20.0
The area of the Triangle is: 40.0
The area of the circle is: 314.0
o/p seen
28/10/24

Code:

```
//USN:1BM23CS011
//name:Abhiram Rayadurgam
import java.util.Scanner;

abstract class Shape {
 double dim1, dim2;

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

 abstract double printArea();
}

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

 double printArea() {
 System.out.printf("The area of the Rectangle is: " + (dim1 * dim2) + "\n");
 return 0;
 }

}

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

 double printArea() {
 System.out.printf("The area of the Triangle is: " + ((dim1 * dim2) / 2) + "\n");
 return 0;
 }
}

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

 double printArea() {
 System.out.printf("The area of the Circle is: " + (3.14 * dim1 * dim1) + "\n");
 return 0;
 }
}

class Area {
```

```
 public static void main(String[] args) {
 Rectangle r = new Rectangle(4, 5);
 Triangle t = new Triangle(10, 8);
 Circle c = new Circle(10, 10);
 Shape s;
 s = r;
 s.printArea();
 s = t;
 s.printArea();
 s = c;
 s.printArea();
 }
}
```

Output:



Program 5

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

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

a)Accept deposit from customer and update the balance.
b)Display the balance.
c)Compute and deposit interest
d)Permit withdrawal and update the balance
e) Check for the minimum balance, impose penalty if necessary and update the balance.

Observation:

Lab Program 5

DATE: PAGE:

Q: Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no ~~check~~ 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.

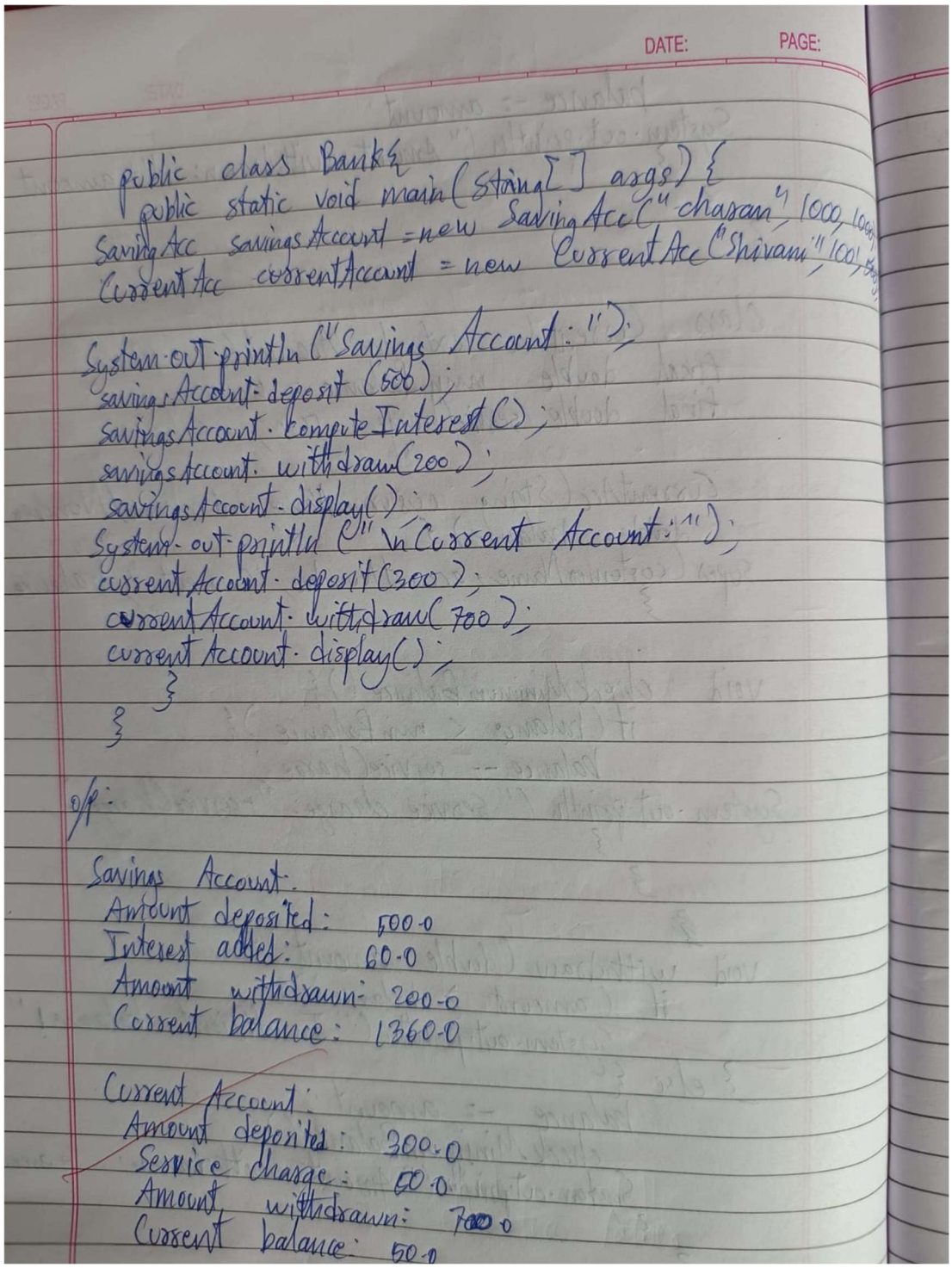
```
class Account {
    String customerName;
    int accNumber;
    String accountType;
    double balance;

    Account(String customerName, int accNumber,
    String accountType, double balance){
        this.customerName = customerName;
        this.accNumber = accNumber;
        this.accountType = accountType;
        this.balance = balance;
    }
```

DATE: PAGE:

```
Void deposit(double amount){
    if(amount >0){
    balance += amount;
    System.out.println("Amount deposited: " + amount);
    } else {
    System.out.println("Invalid deposit amount.");
    }
}

Void display(){
    System.out.println("current balance: " + balance);
    }
}


class SavingAcc extends Account {
    final double interestRate = 0.04;
    SavingAcc(String customerName, int accountNumber,
              double balance){
super(customerName, accountNumber, "Savings", balance);
    }

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

Void withdraw(double amount){
    if(amount > balance){
    System.out.println("Insufficient balance!");} 
    else {
```

DATE: PAGE:

```
        balance -= amount;
        System.out.println("Amount withdrawn: " + amount);
    }
}

class CurrentAcc extends Account {
    final double minBalance = 500.0;
    final double serviceCharge = 50.0;

    CurrentAcc(String customerName, int accountNumber,
               double balance) {
        super(customerName, accountNumber, "Current", balance);
    }

    void checkMinimumBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge: " + serviceCharge);
        }
    }

    void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient balance!");
        } else {
            balance -= amount;
            checkMinimumBalance();
            System.out.println("Amount withdrawn: " + amount);
        }
    }
}
```



Code:

```
//USN:1BM23CS011
//name:Abhiram Rayadurgam
class Account {
    String customerName;
```

```
    int accNumber;
    String accountType;
    double balance;

    Account(String customerName, int accNumber, String accountType, double balance) {
        this.customerName = customerName;
        this.accNumber = accNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Amount deposited: " + amount);
        } else {
            System.out.println("Invalid deposit amount.");
        }
    }

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

class SavingAcc extends Account {
    final double interestRate = 0.04;

    SavingAcc(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Savings", balance);
    }

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

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

class CurrentAcc extends Account {
    final double minBalance = 500.0;
```

```
    final double serviceCharge = 50.0;

    CurrentAcc(String customerName, int accountNumber, double balance) {
        super(customerName, accountNumber, "Current", balance);
    }

    void checkMinimumBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge: " + serviceCharge);
        }
    }

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

public class Bank {
    public static void main(String[] args) {
        SavingAcc savingsAccount = new SavingAcc("Charan", 1000, 1000);
        CurrentAcc currentAccount = new CurrentAcc("Shivani", 1001, 500);

        System.out.println("Savings Account:");
        savingsAccount.deposit(500);
        savingsAccount.computeInterest();
        savingsAccount.withdraw(200);
        savingsAccount.display();

        System.out.println("\nCurrent Account:");
        currentAccount.deposit(300);
        currentAccount.withdraw(700);
        currentAccount.display();
    }
}
```

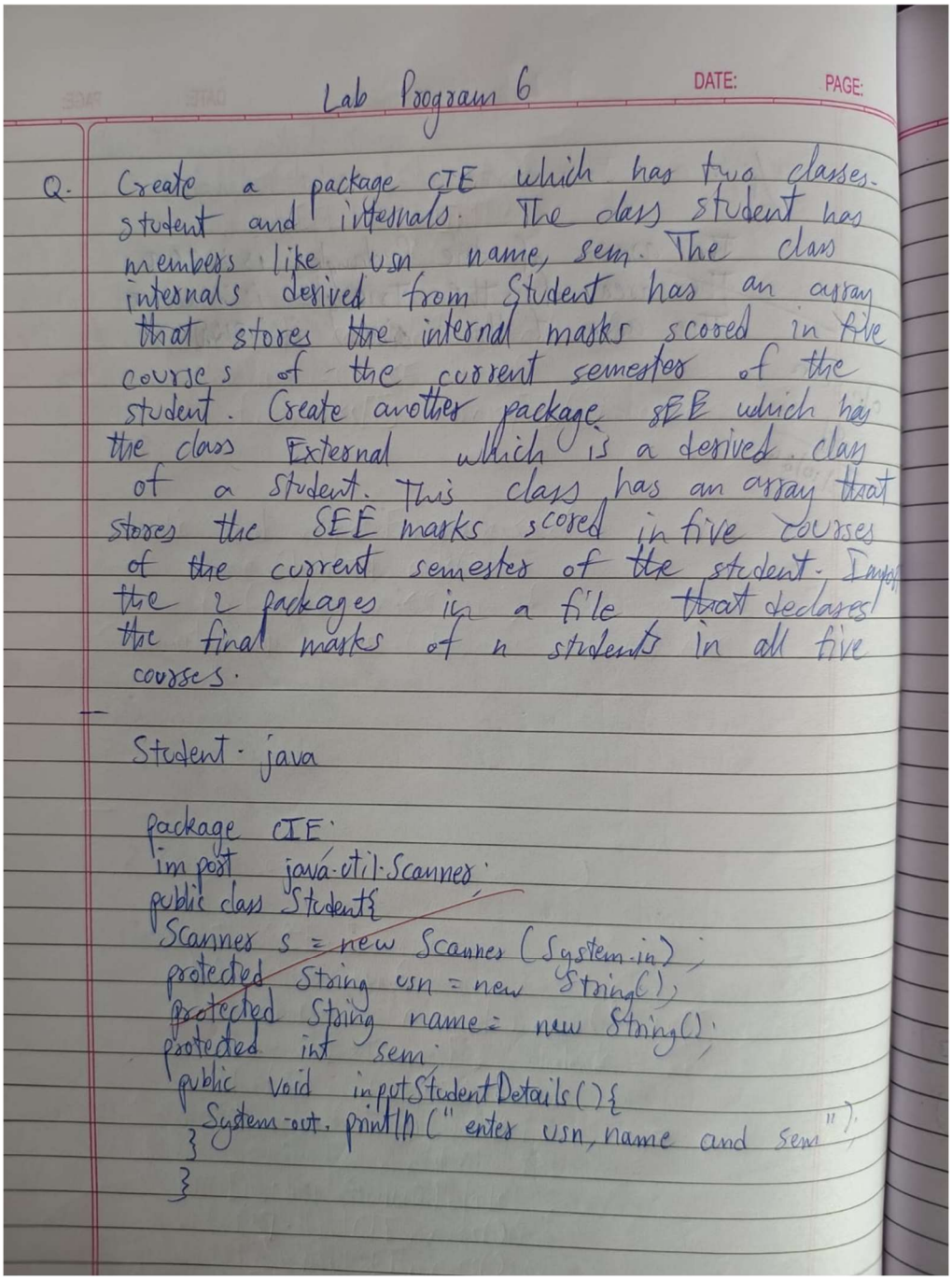
Output:

```
D:\abhiram>java Bank
Savings Account:
Amount deposited: 500.0
Interest added: 60.0
Amount withdrawn: 200.0
Current balance: 1360.0

Current Account:
Amount deposited: 300.0
Service charge: 50.0
Amount withdrawn: 700.0
Current balance: 50.0

D:\abhiram>Name: Abhiram Rayadurgam USN: 1BM23CS011
```

Program 6

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

Observation:



Lab Program 6 DATE: PAGE:

Q. Create a package CIE which has two classes- student and internals. The class student has members like usn, name, sem. The class internals derived from Student has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of a student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the 2 packages in a file that declares the final marks of n students in all five courses.

Student.java

```
package CIE;
import java.util.Scanner;
public class Student{
Scanner s = new Scanner(System.in);
protected String usn = new String();
protected String name = new String();
protected int sem;
public void inputStudentDetails(){
System.out.println("enter usn, name and sem");
}
}
```

DATE: PAGE:

Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
Scanner s = new Scanner(System.in);
protected int cmarks[] = new int[5];
public int total = 0;
public void inputCIE_marks(){
System.out.println("enter 5 courses CIE marks");
for(int i=0; i<5; i++){
cmarks[i] = s.nextInt();
}
}
}
```

Externals.java

```
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals {
protected int marks[] = new int[5];
protected int finalMarks[] = new int[5];
public void inputSEEmarks(){
Scanner s = new Scanner(System.in);
System.out.println("Enter 5 course marks:");
```

```
        for(int i = 0 ; i<5; i++ ){
            marks[i] = s.nextInt();
        }
    }
    public void calculateFinalMarks(){
        for(int i=0; i<5 ; i++){
            finalMarks[i] = (marks[i]/2) + cmarks[i];
        }
    }
    public void displayFinalMarks(){
        displayStudentDetails();
        for(int i=0; i<5; i++){
System.out.println("marks"+(i+1)+": "+finalMarks[i]+"\n");
    }
}
```

Run.java (driver class)

```
import SEE.Externals;
public class Run{
    public static void main(String[] args){
        Externals[] students = new Externals[3];
        for(int i=0; i<3; i++){
System.out.println("Enter details for student "+(i+1)+":");
students[i] = new Externals();
students[i].inputStudentDetails();
students[i].inputCIEmarks();
students[i].inputSEEmarks();
students[i].calculateFinalMarks(); }
```

DATE: PAGE:

```
System.out.println("\n Final marks for each Student:");
for(int i=0; i<3; i++){
    System.out.println("\n Student " + (i+1) + " details:");
    students[i].displayFinalMarks();
  }
 }
}
```

o/p: Enter details for student1:
enter usn, name, sem
1bm231
abhiram
2
enter 5 courses' CIE marks
50
49
48
48
49
Enter 5 course marks:
99
98
89
100
90
Enter details for student 2:
enter usn, name and sem
1bm232
kumar
2
enter 5 courses' CIE marks
47
46
48
49
40
enter 5 course marks:
89
88
80
92
95
Enter details for Student 3:
enter usn, name and sem
1bm233
advaith
2
enter 5 courses' CIE marks
48
49
50
50
46

Enter 5 course marks:
100
99
98
80
86

Final marks for each student:

Student 1 details:
usn = 1bm231, name = abhiram, sem = 2
marks1: 99
marks2: 98
marks3: 92
marks4: 98
marks5: 94

Student 2 details:
usn = 1bm232, name = kumar, sem = 2
marks1: 91
marks2: 90
marks3: 88
marks4: 95
marks5: 87

Student 3 details:
usn = advaith, name = advaith, sem = 2
marks1: 98
marks2: 98
marks3: 99
marks4: 90
marks5: 88

Code:

```
//USN:1BM23CS011
//name:Abhiram Rayadurgam

//Externals.java

package SEE;
import CIE.Internals;

import java.util.Scanner;

public class Externals extends Internals {
protected int marks[] = new int[5];
protected int finalMarks[] = new int[5];

public void inputSEEmarks() {

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

 }}
public void calculateFinalMarks() {

for(int i=0;i<5;i++){
finalMarks[i]=(marks[i]/2)+cmarks[i];
}

 }

public void displayFinalMarks() {

displayStudentDetails();

for(int i=0;i<5;i++){
System.out.printf("marks"+(i+1)+": "+finalMarks[i]+"\n");
}

}

 }


//Internals.java

package CIE;

import java.util.Scanner;

public class Internals extends Student {
```

```
Scanner s = new Scanner(System.in);
protected int cmarks[] = new int[5];
public int total=0;
public void inputCIEmarks()
{
System.out.println("enter 5 courses' CIE marks");
for(int i=0;i<5;i++){

cmarks[i]=s.nextInt();
}
}


}

//Student.java

package CIE;

import java.util.Scanner;

public class Student {
Scanner s = new Scanner(System.in);
protected String usn=new String();
protected String name=new String();
protected int sem;
public void inputStudentDetails(){
System.out.println("enter usn, name and sem");
usn = s.nextLine();
name = s.nextLine();
sem = s.nextInt();
}
public void displayStudentDetails() {
System.out.printf("usn=" + usn +", name="+ name+", sem="+sem+"\n");
}

}

//Run.java

import SEE.Externals;

public class Run {
    public static void main(String[] args) {
        Externals[] students = new Externals[3];

        for (int i = 0; i < 3; i++) {
            System.out.println("Enter details for Student " + (i + 1) + ":");
            students[i] = new Externals();
            students[i].inputStudentDetails();
```

```
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
            students[i].calculateFinalMarks();
        }
        System.out.println("\nFinal marks for each student:");
        for (int i = 0; i < 3; i++) {
            System.out.println("\nStudent " + (i + 1) + " details:");
            students[i].displayFinalMarks();
        }
    }
}
```

Output:

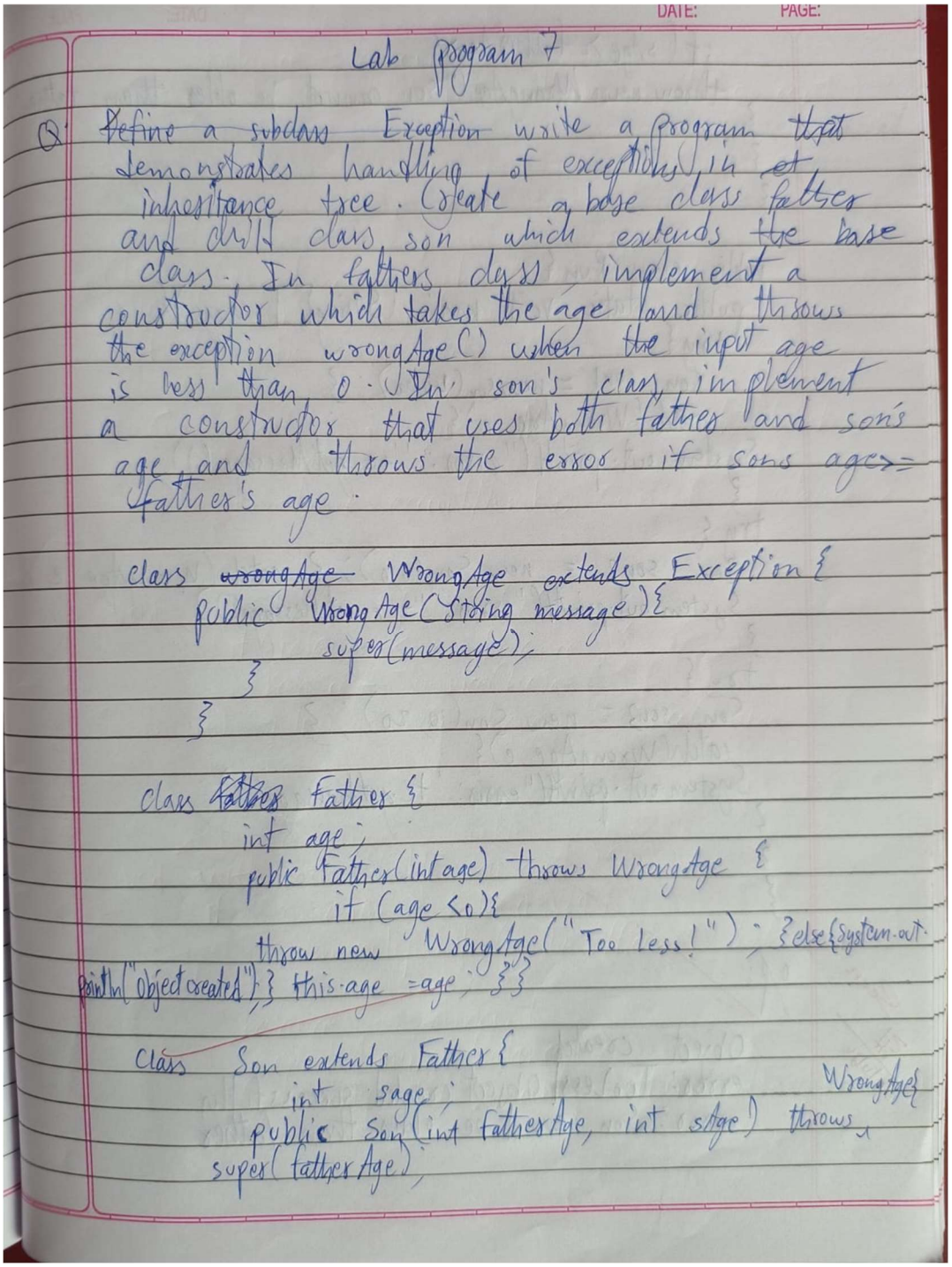
```
D:\abhiram>java Run
Enter details for Student 1:
enter usn, name and sem
1bm231
abhiram
2
enter 5 courses' CIE marks
50
49
48
47
49
Enter 5 course marks:
100
99
98
97
90
Enter details for Student 2:
enter usn, name and sem
1bm232
ariz
2
enter 5 courses' CIE marks
44
45
49
50
50
Enter 5 course marks:
100
99
98
90
100
Enter details for Student 3:
enter usn, name and sem
1bm233
arnav
2
```

```
enter 5 courses' CIE marks
45
46
47
48
49
Enter 5 course marks:
100
99
89
97
94

Final marks for each student:

Student 1 details:
usn=1bm231, name=abhiram, sem=2
marks1: 100
marks2: 98
marks3: 97
marks4: 95
marks5: 94

Student 2 details:
usn=1bm232, name=ariz, sem=2
marks1: 94
marks2: 94
marks3: 98
marks4: 95
marks5: 100

Student 3 details:
usn=1bm233, name=arnav, sem=2
marks1: 95
marks2: 95
marks3: 91
marks4: 96
marks5: 96

D:\abhiram>Name: Abhiram Rayadurgam USN: 1BM23CS011
```

Program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class father and child class son which extends the base class. In father class, implement a constructor which takes the age and throws the exception wrongAge() when the input is less than 0. In son's class implement a constructor that uses both father and son's age and throws the error if son's age >= father's age.

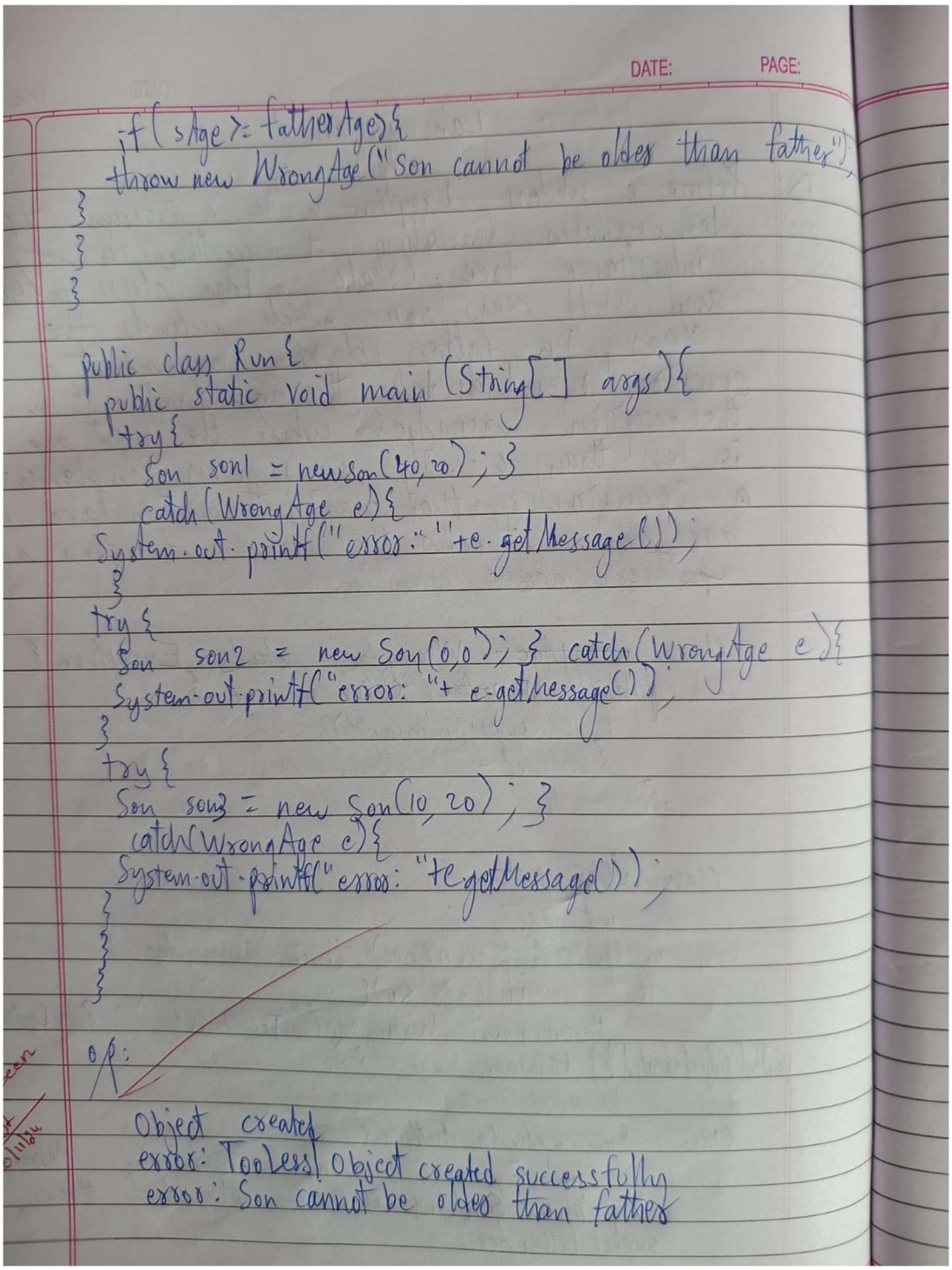
Observation:



Lab program 7

Q: Define a subclass Exception write a program that demonstrates handling of exceptions in inheritance tree. Create a base class father and child class son which extends the base class. In fathers class implement a constructor which takes the age and throws the exception wrongAge() when the input age is less than 0. In son's class implement a constructor that uses both father and son's age and throws the error if sons age >= father's age.

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

class Father {
    int age;
    public Father(int age) throws WrongAge {
        if (age <0){
            throw new WrongAge("Too less!");} else{System.out.println("Object created");} this.age =age;}}

class Son extends Father{
    int sage;
    public Son(int fatherAge, int sage) throws WrongAge{
        super(fatherAge);
```



```
    if(sAge >= fatherAge){
    throw new WrongAge("Son cannot be older than father");
}
}
}

public class Run{
  public static void main(String[] args){
    try{
      Son son1 = new Son(40,20);}
      catch(WrongAge e){
  System.out.printf("error: "+e.getMessage());
    }
  try{
    Son son2 = new Son(0,0);} catch(WrongAge e){
    System.out.printf("error: "+ e.getMessage());
  }
  try{
    Son son3 = new Son(10,20);}
    catch(WrongAge e){
    System.out.printf("error: "+e.getMessage());
  }
  }
  }

o/p:

    Object created
    error: TooLess! Object created successfully
    error: Son cannot be older than father
```

Code:

```
//USN:1BM23CS011
//name:Abhiram Rayadurgam
class WrongAge extends Exception{
public WrongAge(String m){
super(m);
}
}

class Father{
int age;
public Father(int age) throws WrongAge{
if(age<=0){
throw new WrongAge("Too Less!");
} else {
System.out.println("Object created successfully");
}
this.age=age;
}
}

class Son extends Father{
int sAge;
public Son(int fatherAge,int sAge) throws WrongAge{
super(fatherAge);
if(sAge>=fatherAge){
throw new WrongAge("Son cannot be older than father");
}
}
}

public class Run{
public static void main(String[] args){
try{
Son son1 = new Son(40,20);
} catch(WrongAge e){
System.out.printf("error: "+e.getMessage());
}
try{
Son son1 = new Son(0,0);
} catch(WrongAge e){
System.out.printf("error: "+e.getMessage());
}
try{
Son son1 = new Son(10,20);
} catch(WrongAge e){
System.out.printf("error: "+e.getMessage());
}
}
}
```

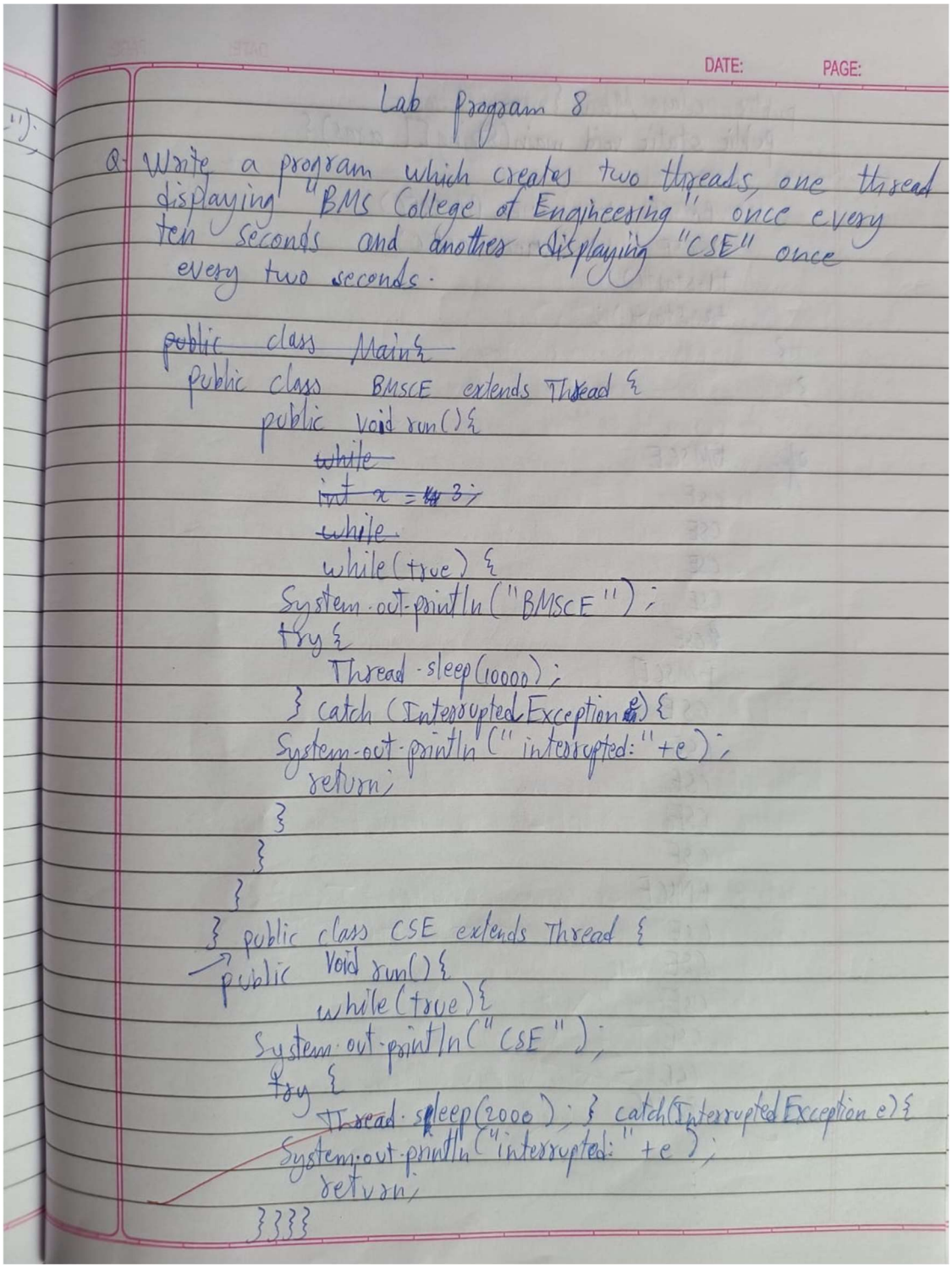
Output:

```
D:\>javac Run.java

D:\>java Run
Object created successfully
error: Too Less!Object created successfully
error: Son cannot be older than father
D:\>Name: Abhiram Rayadurgam USN: 1BM23CS011
```

Program 8

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.

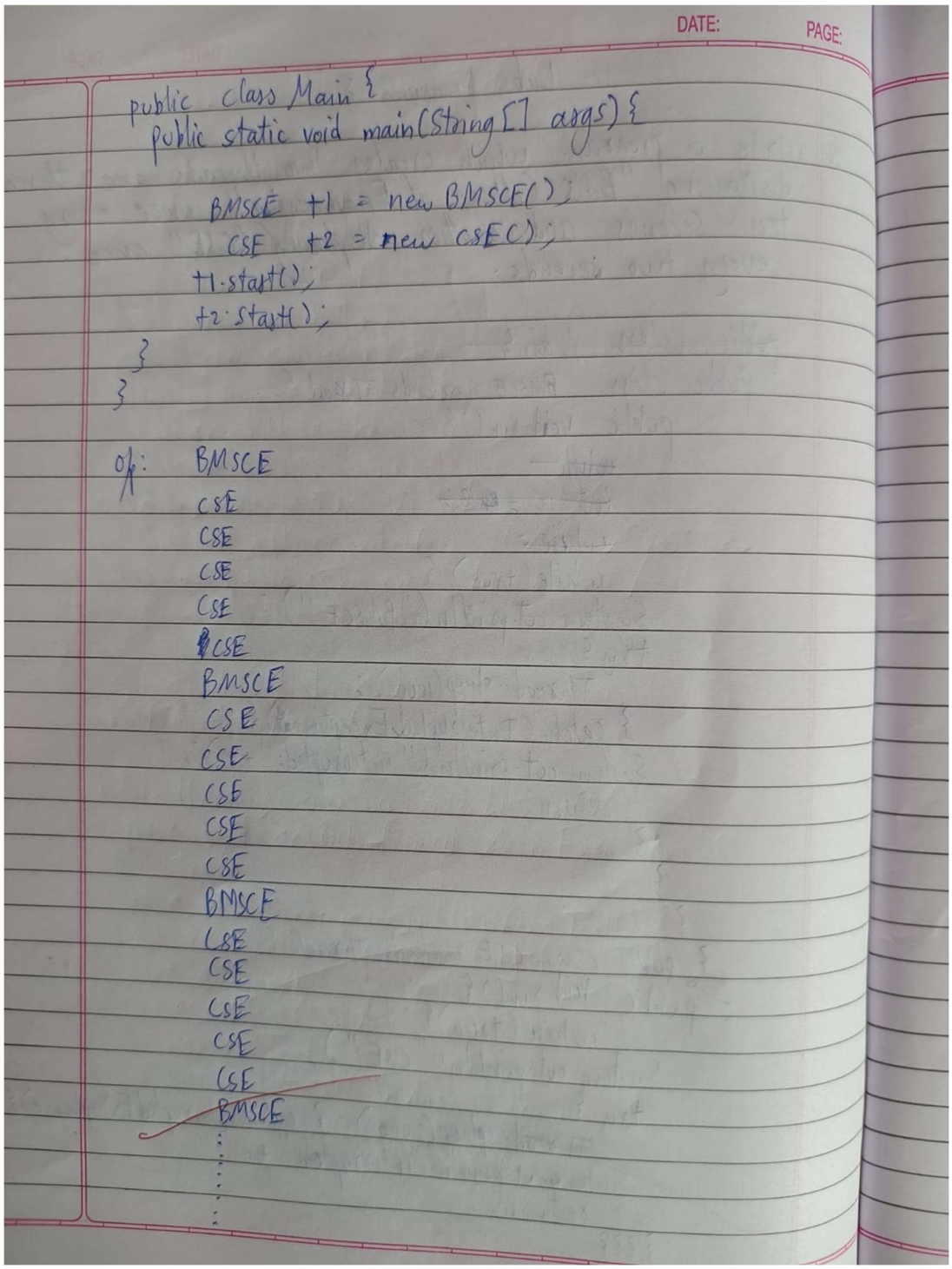
Observation:



Lab 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.

```
public class BMSCE extends Thread {
    public void run(){
        while(true) {
            System.out.println("BMSCE");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                System.out.println("interrupted:"+e);
                return;
            }
        }
    }
public class CSE extends Thread {
    public void run(){
        while(true){
            System.out.println("CSE");
            try {
                Thread.sleep(2000); } catch(InterruptedException e){
                System.out.println("interrupted:"+e);
                return;
            }}}}
```


DATE:
PAGE:
public class Main {
public static void main(String[] args) {
BMSCE t1 = new BMSCE();
CSE t2 = new CSE();
t1.start();
t2.start();
}
}
O/P: BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE

Code:

```
//USN:1BM23CS011
//name:Abhiram Rayadurgam

//BMSCE.java

public class BMSCE extends Thread{
public void run(){
while(true){
System.out.println("BMSCE College of Engineering");
try{
Thread.sleep(10000);}
catch(InterruptedException e){
System.out.println("interrupted: "+e);
return;
}}}}

//CSE.java

public class CSE extends Thread{
public void run(){
while(true){
System.out.println("CSE");
try{
Thread.sleep(2000);}
catch(InterruptedException e){
System.out.println("interrupted: "+e);
return;
}}}}

//Runs.java

public class Runs{
public static void main(String[] args){
BMSCE t1 = new BMSCE();
CSE t2 = new CSE();
t1.start();
t2.start(); }}
```

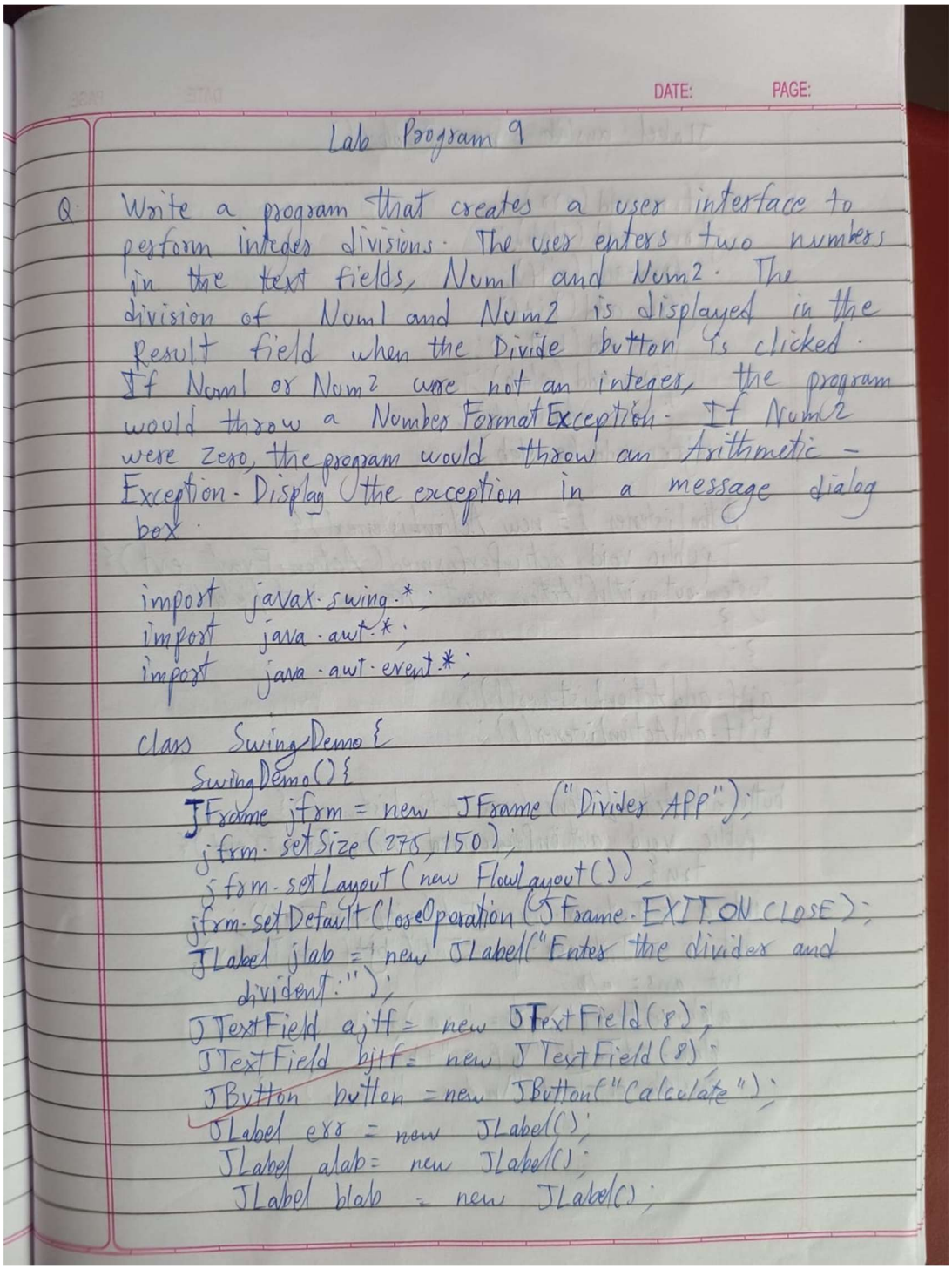
Output:

```
BMSCE College of Engineering
CSE
CSE
CSE
CSE
CSE
BMSCE College of Engineering
CSE
CSE
CSE
CSE
CSE
BMSCE College of Engineering
CSE
CSE
CSE
CSE
CSE
BMSCE College of Engineering
CSE
CSE
CSE
CSE
^C
D:\1BM23CS011>Name: Abhiram Rayadurgam USN: 1BM23CS011
```

Program 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. 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.

Observation:



DATE: PAGE:

Lab Program 9

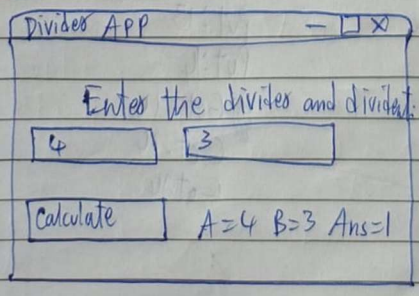
Q: Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw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

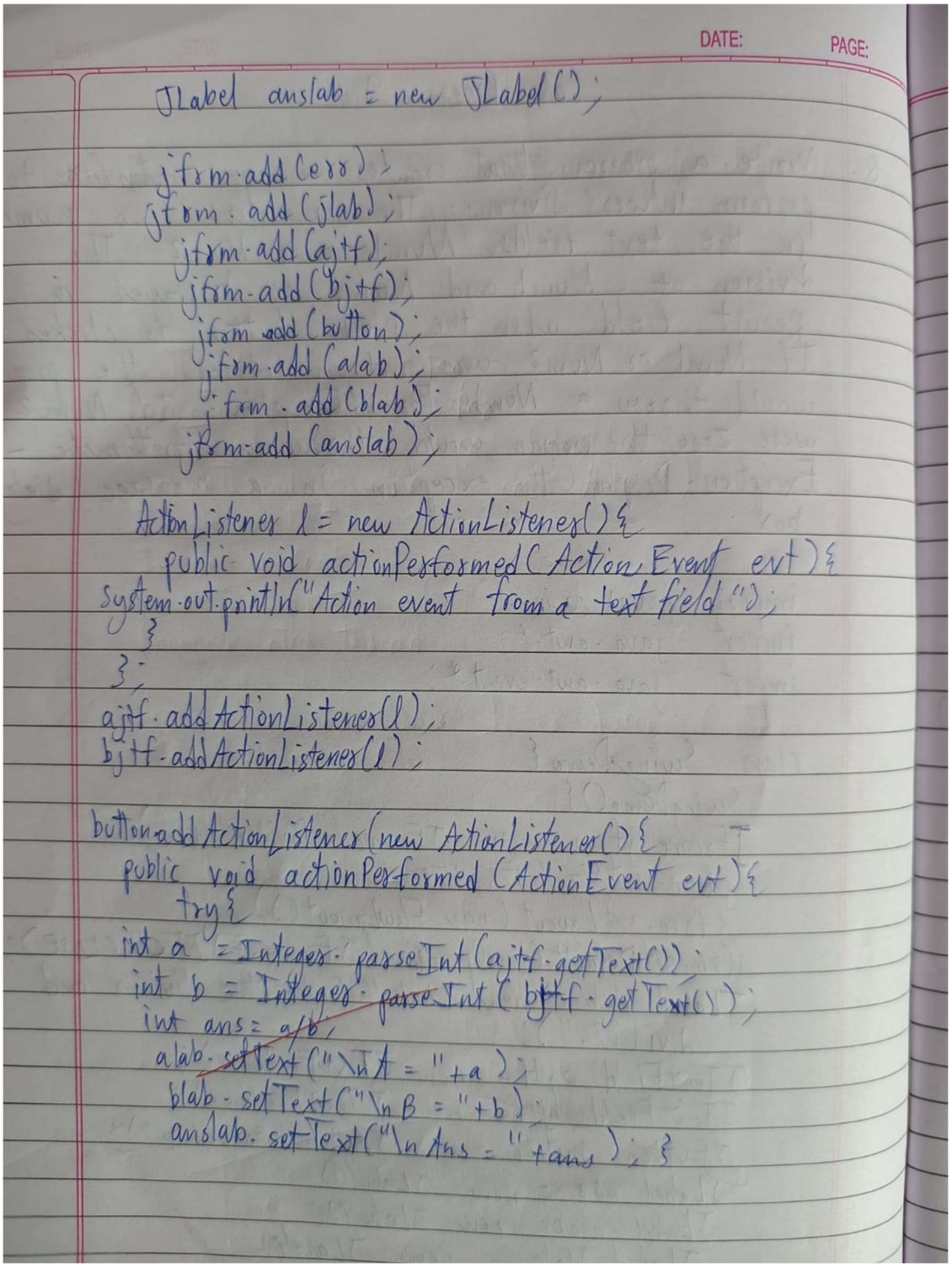
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
    SwingDemo() {
    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());
    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JLabel jlab = new JLabel("Enter the divider and
        divident:");
    JTextField ajtf = new JTextField(8);
    JTextField bjtf = new JTextField(8);
    JButton button = new JButton("Calculate");
    JLabel err = new JLabel();
    JLabel alab = new JLabel();
    JLabel blab = new JLabel();
```

DATE: PAGE:

```
catch(NumberFormatException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    esr.setText("Enter Only Integers!");}
Catch(ArithmeticException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    esr.setText("B should be NON zero!");
  }
}
});
    jfrm.setVisible(true);
}
public static void main(String[] args){
SwingUtilities.invokeLater(new Runnable(){
    public void run(){
      new SwingDemo();
    }
  });
}
}
```

o/p:





```
JLabel anslab = new JLabel();

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

ActionListener l = new ActionListener() {
   public void actionPerformed(ActionEvent evt) {
System.out.println("Action event from a text field");
   }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);

button.addActionListener(new ActionListener() {
  public void actionPerformed(ActionEvent evt) {
     try{
int a = Integer.parseInt(ajtf.getText());
 int b = Integer.parseInt(bjtf.getText());
  int ans = a/b;
  alab.setText("\n A = "+a);
   blab.setText("\n B = "+b);
   anslab.setText("\n Ans = "+ans);  }
```

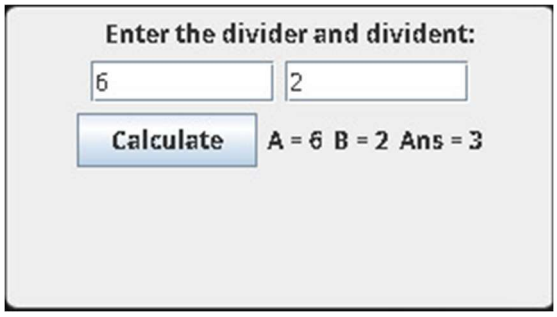
Code:

//USN:1BM23CS011

```
//name:Abhiram Rayadurgam
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

class SwingDemo {
  SwingDemo() {

    JFrame jfrm = new JFrame("Divider App");
    jfrm.setSize(275, 150);
    jfrm.setLayout(new FlowLayout());

    jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    JLabel jlab = new JLabel("Enter the divider and divident:");

    JTextField ajtf = new JTextField(8);
    JTextField bjtf = new JTextField(8);

    JButton button = new JButton("Calculate");

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

    JLabel anslab = new JLabel();

    jfrm.add(err);
    jfrm.add(jlab);
    jfrm.add(ajtf);
    jfrm.add(bjtf);
    jfrm.add(button);
    jfrm.add(alab);
    jfrm.add(blab);
    jfrm.add(anslab);
    ActionListener l = new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
        System.out.println("Action event from a text field");
      }
    };
    ajtf.addActionListener(l);
    bjtf.addActionListener(l);
    button.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent evt) {
        try {
          int a = Integer.parseInt(ajtf.getText());
          int b = Integer.parseInt(bjtf.getText());
          int ans = a / b;
          alab.setText("\nA = " + a);
          blab.setText("\nB = " + b);
```

```
          anslab.setText("\nAns = " + ans);
        } catch (NumberFormatException e) {
          alab.setText("");
          blab.setText("");
          anslab.setText("");

          err.setText("Enter Only Integers!");
        } catch (ArithmeticException e) {
          alab.setText("");
          blab.setText("");
          anslab.setText("");
          err.setText("B should be NON zero!");
        }
      }
    });

    jfrm.setVisible(true);
  }

  public static void main(String args[]) {

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

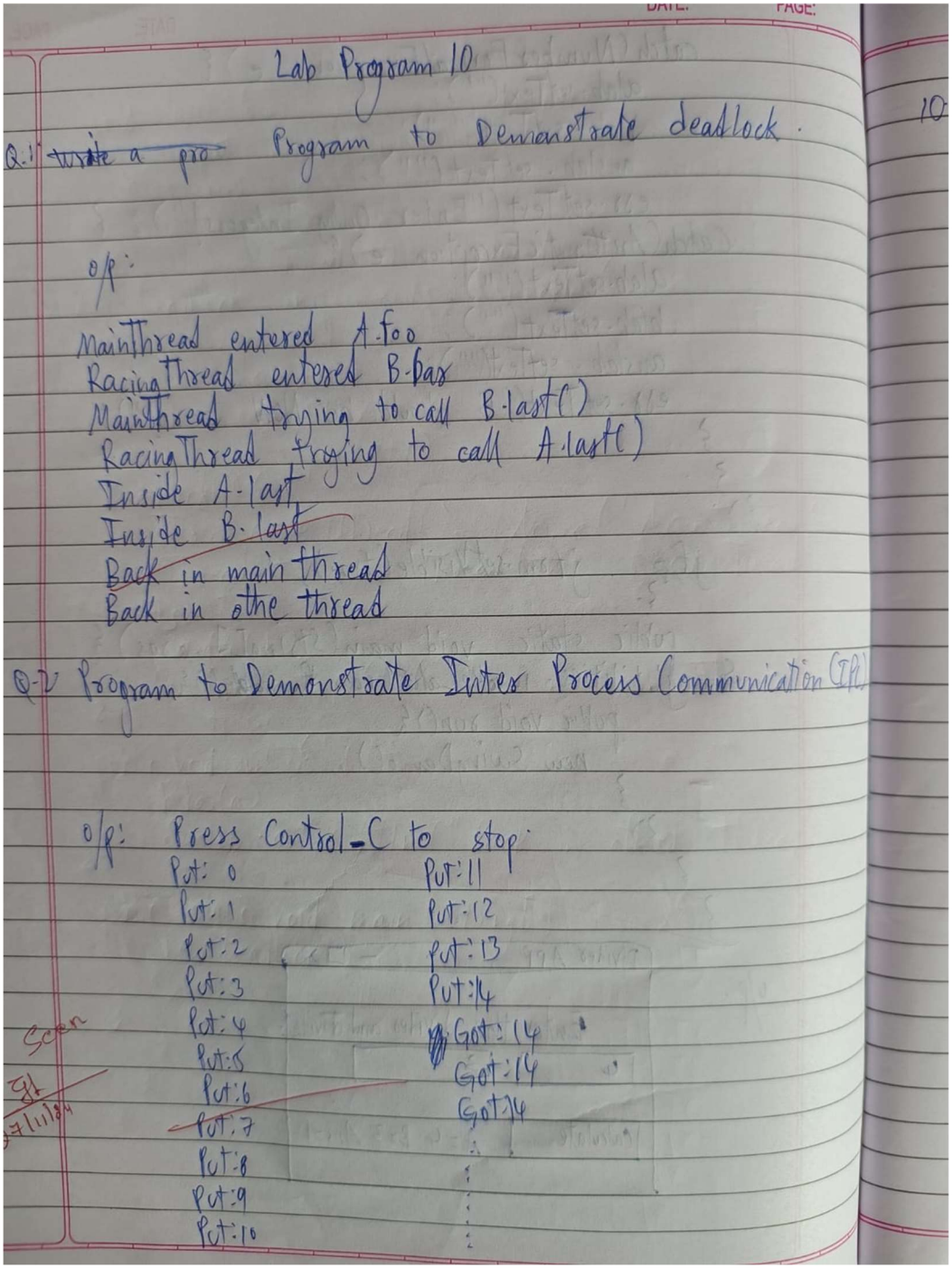
Output:



Program 10

1. Write a program to demonstrate deadlock.

Observation:



Lab Program 10

Q.1) ~~write a pro~~ Program to Demonstrate deadlock.

o/p:

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

Q.2) Program to Demonstrate Inter Process Communication (IPC)

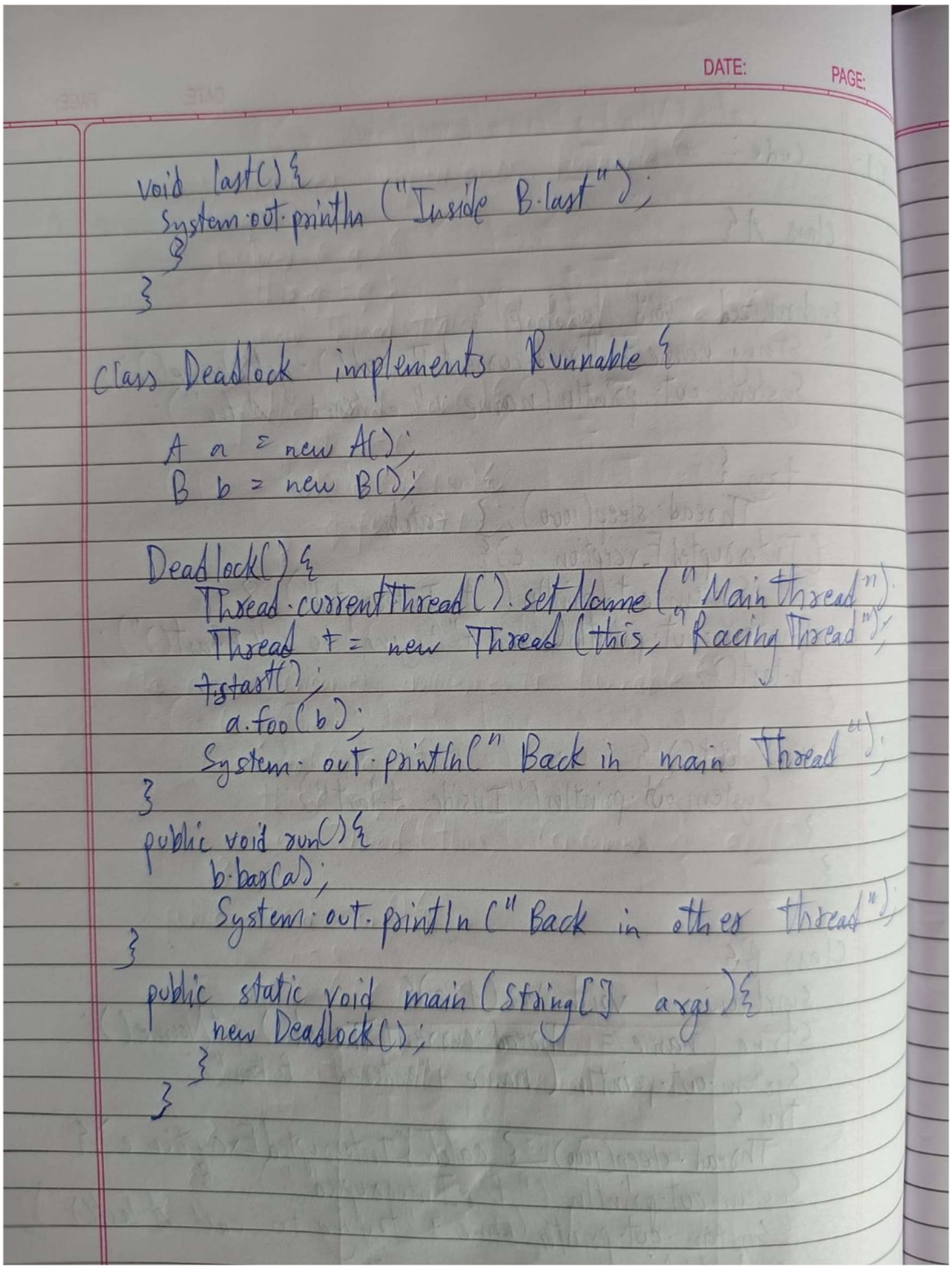
o/p: Press Control-C to stop

Put: 0
Put: 1
Put: 2
Put: 3
Put: 4
Put: 5
Put: 6
Put: 7
Put: 8
Put: 9
Put: 10
Put: 11
Put: 12
Put: 13
Put: 14
Got: 14
Got: 14
Got: 14

DATE: PAGE:

101. code:

```
Class A{

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

  try {
    Thread.sleep(1000); } catch
  (InterruptedException e){
    System.out.println("A Interrupted"); }
  System.out.println(name + " trying to call B.last()");
  b.last();
}

  void last() {
    System.out.println("Inside A.last");
  }
}

Class B{
  Synchronized void bas(A a){
  String name = Thread.currentThread().getName();
  System.out.println(name + "entered B.bas");
  try {
    Thread.sleep(1000); } catch (InterruptedException e){
  System.out.println("B Interrupted"); }
    System.out.println(name + "trying to call A.last()");
    a.last();
  }
```



```
void last(){
  System.out.println("Inside B.last");
}
}

class Deadlock implements Runnable {

  A a = new A();
  B b = new B();

  Deadlock(){
    Thread.currentThread().setName("Main thread");
    Thread t = new Thread(this, "Racing Thread");
    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();
  }
}
```

Code:

```
//USN:1BM23CS011
//name:Abhiram Rayadurgam
class A {

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

    try {
      Thread.sleep(1000);
    } catch (InterruptedException e) {
      System.out.println("A Interrupted");
    }

    System.out.println(name + " trying to call B.last()");
    b.last();
  }

  void last() {
    System.out.println("Inside A.last");
  }
}

class B {

  synchronized void bar(A a) {
    String name = Thread.currentThread().getName();
    System.out.println(name + " entered B.bar");

    try {
      Thread.sleep(1000);
    } catch (InterruptedException e) {
      System.out.println("B Interrupted");
    }

    System.out.println(name + " trying to call A.last()");
    a.last();
  }

  void last() {
    System.out.println("Inside B.last");
  }
}

class Deadlock implements Runnable {

  A a = new A();
  B b = new B();

  Deadlock() {
```

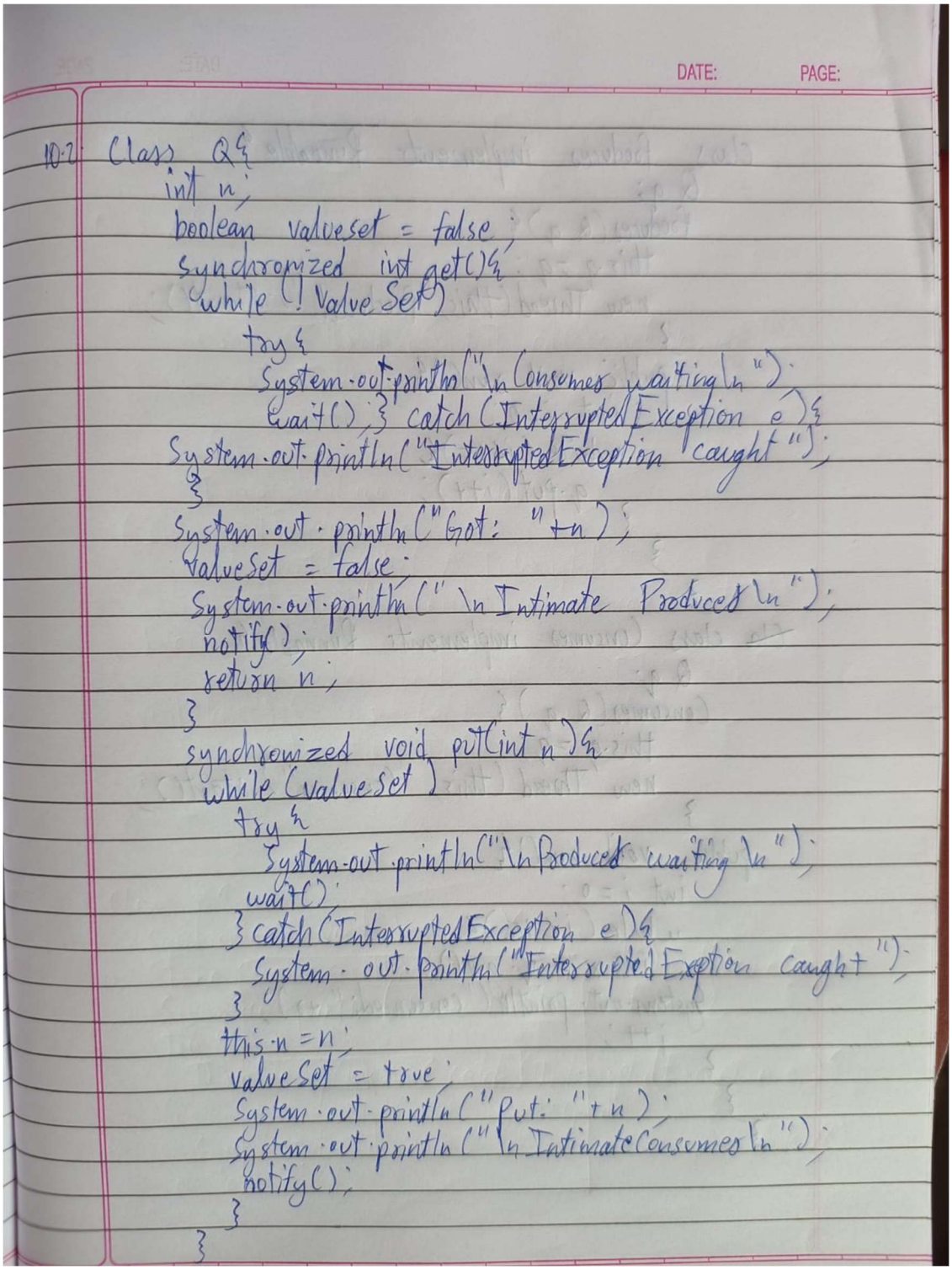
```
    Thread.currentThread().setName("MainThread");
    Thread t = new Thread(this, "RacingThread");
    t.start();

    a.foo(b); // Main thread calls foo on object A, locking A
    System.out.println("Back in main thread");
  }

  public void run() {
    b.bar(a); // Racing thread calls bar on object B, locking B
    System.out.println("Back in other thread");
  }

  public static void main(String args[]) {
    new Deadlock();
  }
}
```

Output:

```
Producer waiting

Got: 12

Intimate Producer

consumed:12
Put: 13

Intimate Consumer


Producer waiting

Got: 13

Intimate Producer

consumed:13
Put: 14

Intimate Consumer

Got: 14

Intimate Producer

consumed:14
PS C:\foder> Name: Abhiram Rayadurgam USN: 1BM23CS011
```

2. Write a program to demonstrate Inter Process Communication (IPC).

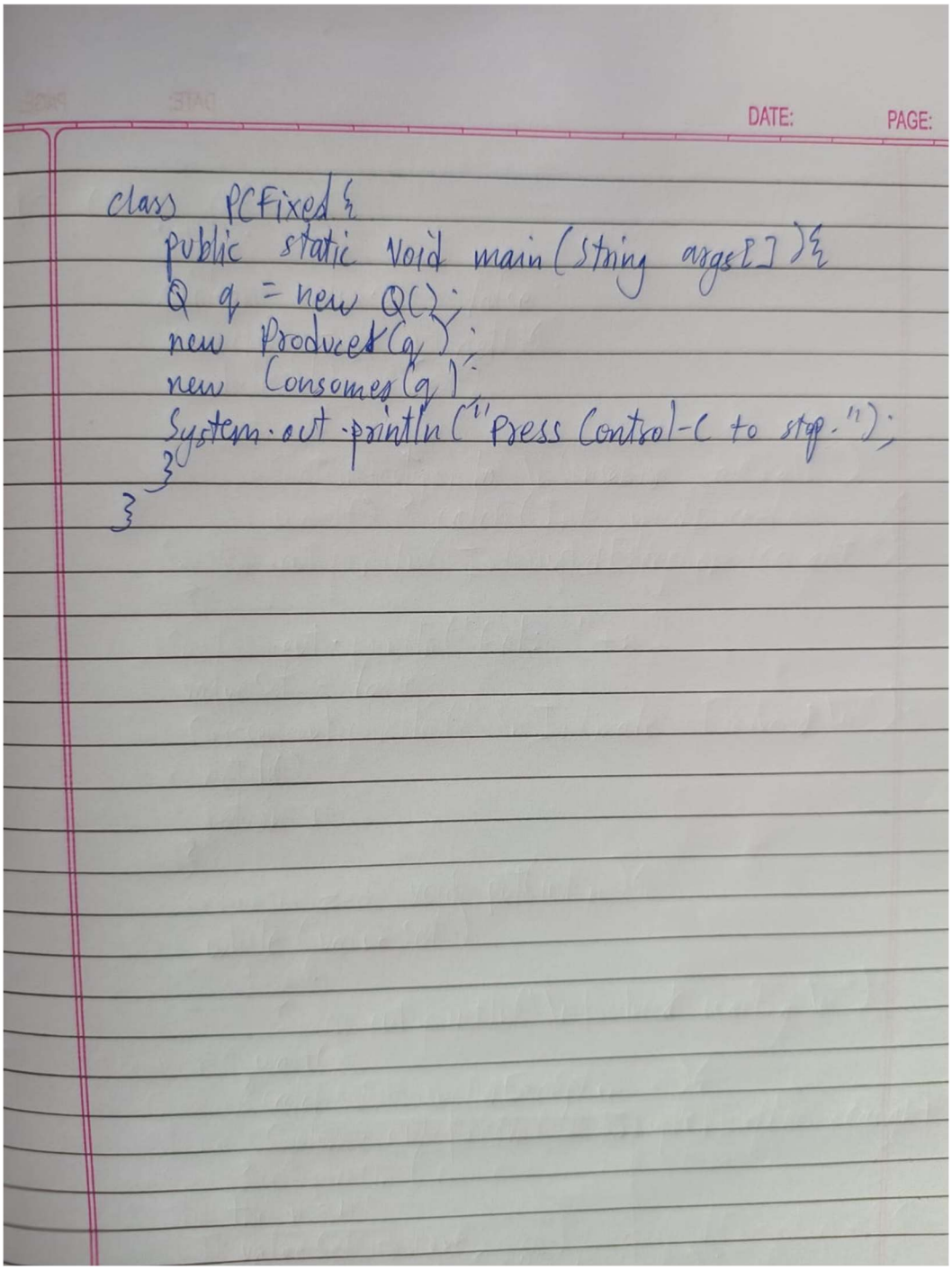
Observation:



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

DATE: PAGE:

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



Code:

//USN:1BM23CS011

```
//name:Abhiram Rayadurgam
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:

```
RacingThread entered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
Inside A.last
Inside B.last
Back in other thread
Back in main thread
PS C:\foder> Name: Abhiram Rayadurgam USN: 1BM23CS011
```