VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Amith R(1BM23CS028)

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 **Amith R(1BM23CS028),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java Programming (23CS3PCOOJ) work prescribed for the said degree.

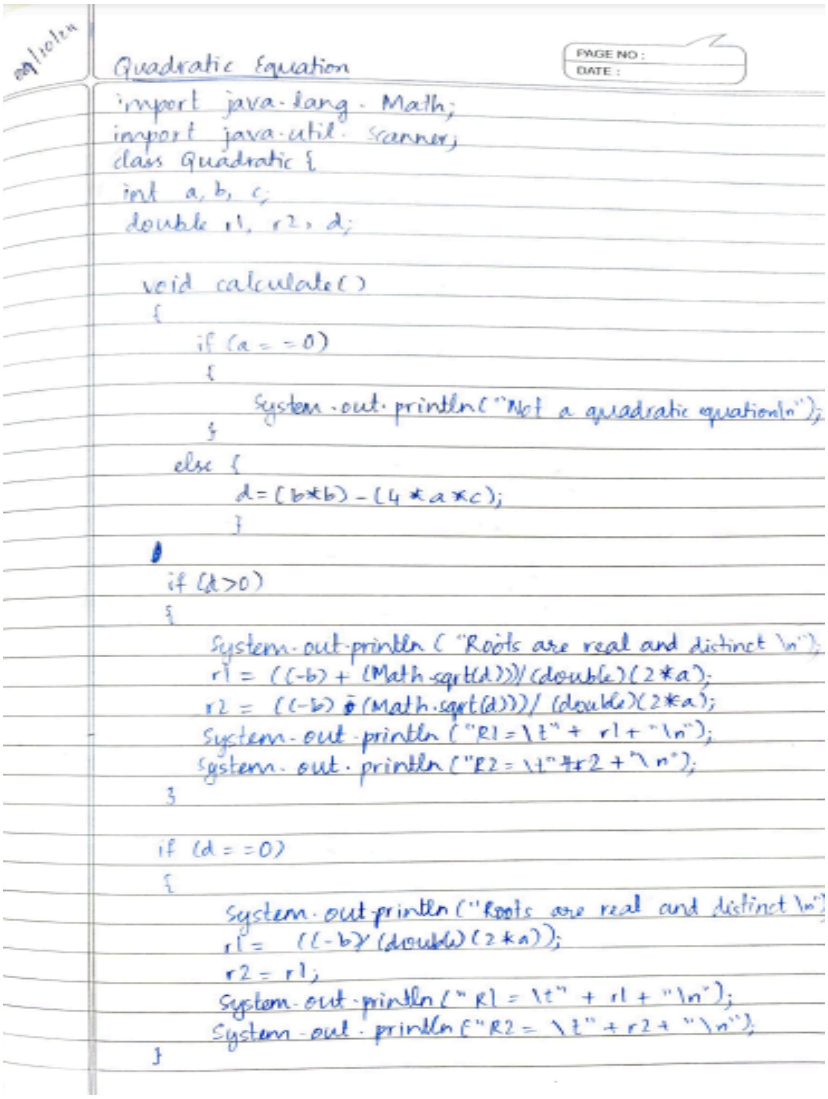
Basavaraj Jakkali Associate Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

Github Link:
https://github.com/amith028/OOJ

Program 1
Implement Quadratic Equation

Algorithm:

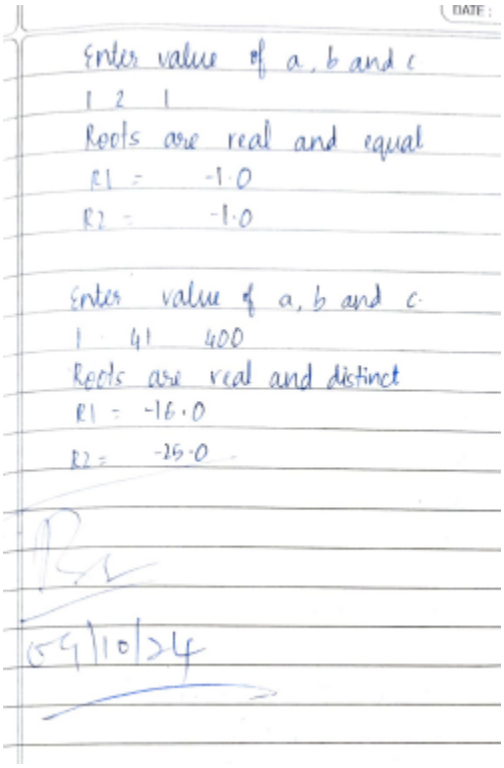


```
Quadratic Equation

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

void calculate()
{
    if (a==0)
    {
        System.out.println("Not a quadratic equation\n");
    }
  else {
        d=(b*b)-(4*a*c);
        }

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

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

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

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

Output:

```
Enter the value of a, b and c
1 1 1
Roots are distinct and imaginary
R1 = 0.3660254037844386i
R2 = -1.3660254037844386i
```



Code:

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

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

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

        }

}

class Run{
        public static void main(String args[])
        {
            System.out.println("NAME: AMITH R");
            System.out.println("USN: 1BM23CS028");
            Quadratic Q= new Quadratic();
            System.out.println("Enter the value of a, b and c \n");
            Scanner sc = new Scanner(System.in);
            Q.a=sc.nextInt();
            Q.b=sc.nextInt();
            Q.c=sc.nextInt();
            Q.calculate();
        }
}
```

```
NAME: AMITH R
USN: 1BM23CS028
Enter the value of a, b and c

1 41 400
Roots are real and distinct

R1=     -16.0

R2=     -25.0
```

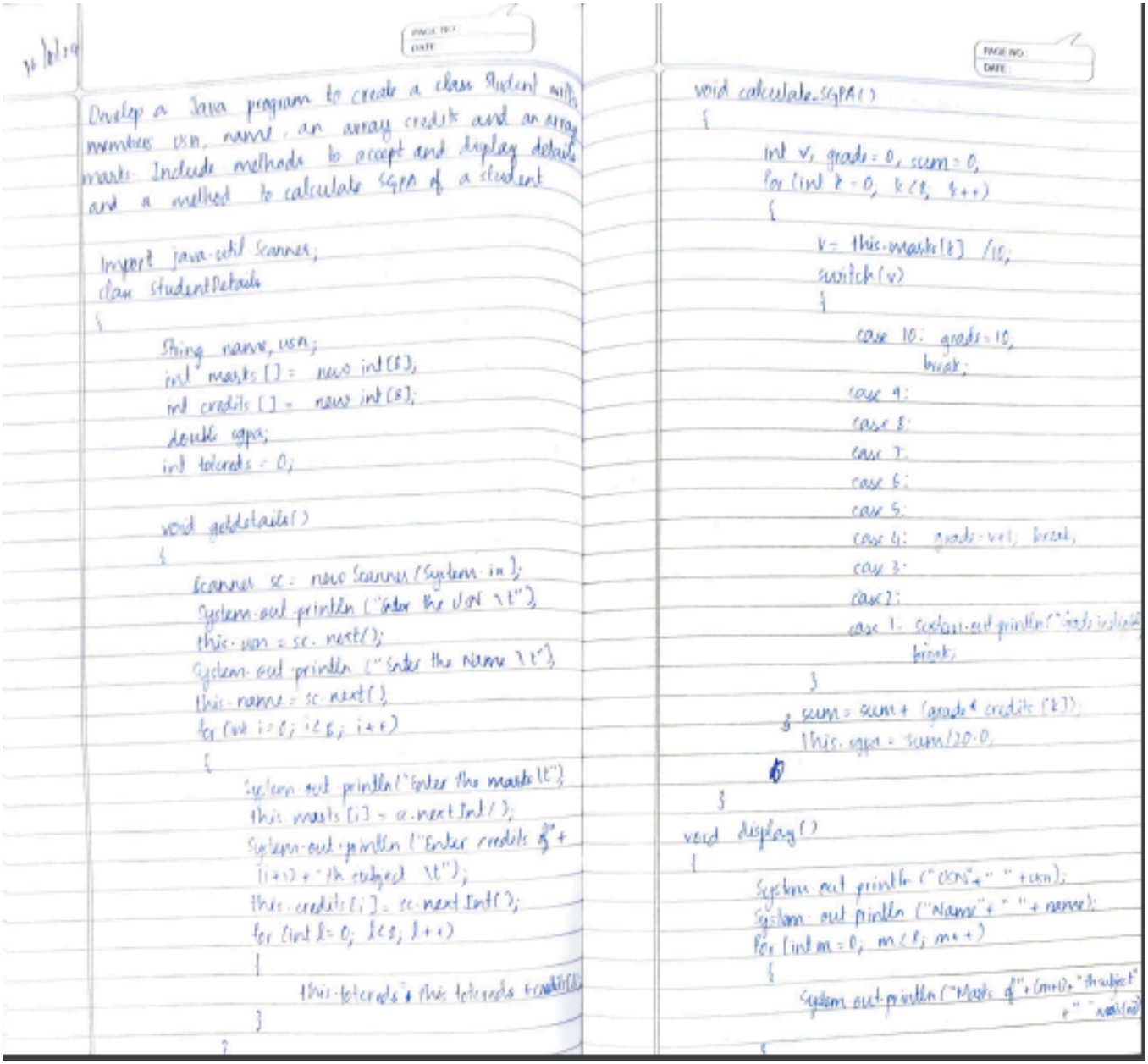
```
NAME: AMITH R
USN: 1BM23CS028
Enter the value of a, b and c

1 2 1
Roots are real and equal

R1=     -1.0

R2=     -1.0
```

```
NAME: AMITH R
USN: 1BM23CS028
Enter the value of a, b and c

1 1 1
Roots are distinct and imaginary

R1=     0.36602540378443864i

R2=     -1.36602540378443864i
```

Program 2

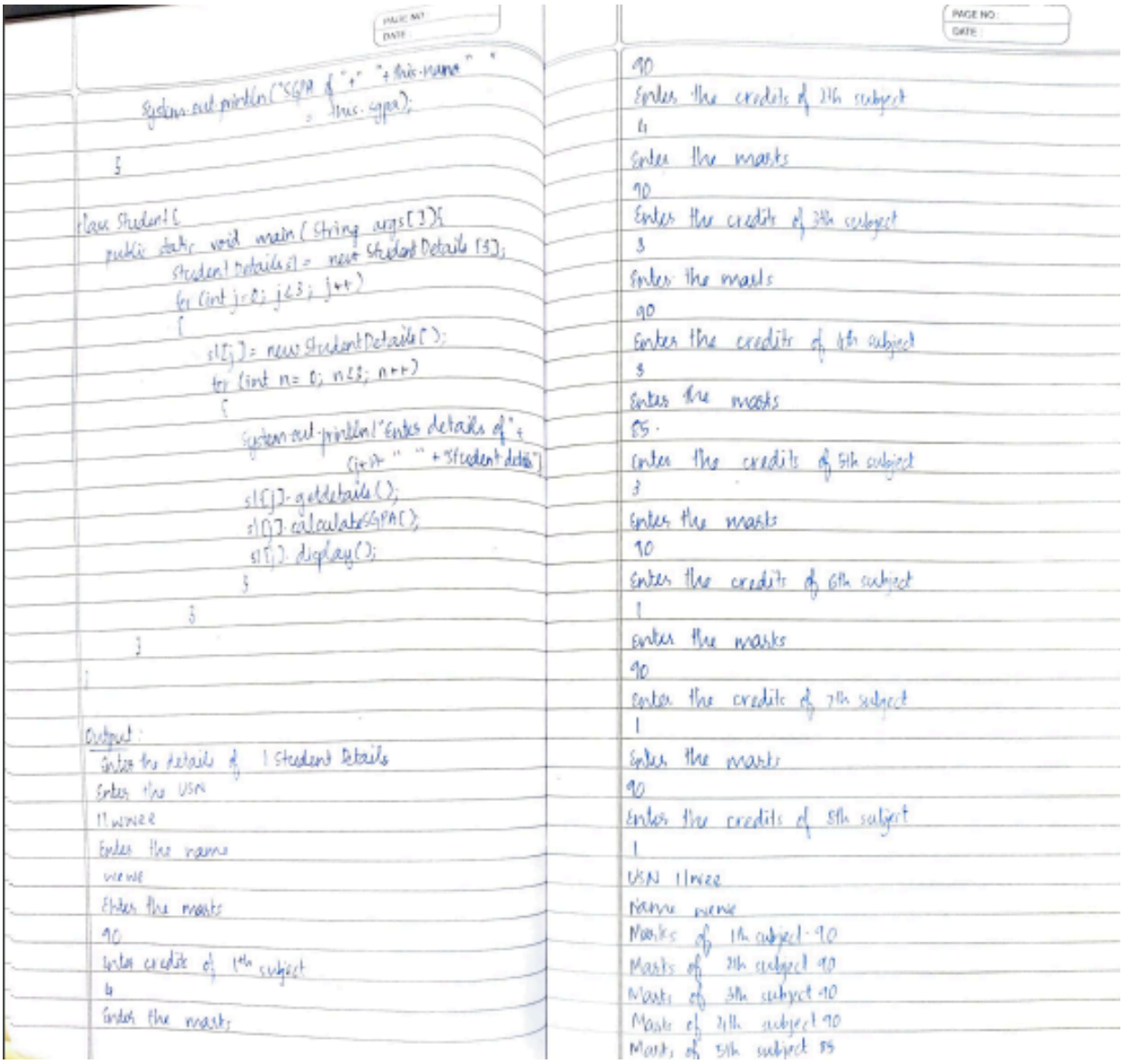
SGPA Calculator

Algorithm:



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

```
import java.util.Scanner;
class StudentDetails
{
    String name, usn;
    int marks[] = new int[8];
    int credits[] = new int[8];
    double sgpa;
    int totcredits = 0;

    void getdetails()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the USN \t");
        this.usn = sc.next();
        System.out.println("Enter the Name \t");
        this.name = sc.next();
        for(int i=0; i<8; i++)
        {
            System.out.println("Enter the marks \t");
            this.marks[i] = a.nextInt();
            System.out.println("Enter credits of "+
            (i+1)+"th subject \t");
            this.credits[i] = sc.nextInt();
            for(int k=0; k<8; k++)
            {
                this.totcredits = this.totcredits + credits[k];
            }
        }
    void calculate_SGPA()
    {
        int v, grade=0, sum=0;
        for(int k=0; k<8; k++)
        {
            v = this.marks[k] /10;
            switch(v)
            {
                case 10: grade=10;
                         break;
                case 9:
                case 8:
                case 7:
                case 6:
                case 5:
                case 4: grade=v; break;
                case 3:
                case 2:
                case 1: System.out.println("Grade is less
                         break;
            }
            sum = sum + (grade * credits[k]);
            this.sgpa = sum/10.0;
        }
    void display()
    {
        System.out.println("USN"+" "+usn);
        System.out.println("Name"+" "+name);
        for(int m=0; m<8; m++)
        {
            System.out.println("Marks of"+(m+1)+"subject
            e" "+marks[m]
        }
```



```
        System.out.println("SGPA of " + " " + this.name + " "
                                    + this.sgpa);
    }

Class Student{
    public static void main(String args[]){
        Student details[] = new StudentDetails[3];
        for(int j=0; j<3; j++)
        {
            s[j] = new StudentDetails();
            for(int n=0; n<8; n++)
            {
                System.out.println("Enter details of" +
                                (j+1) + " " + "Student data");
                s[j].getdetails();
                s[j].calculateSGPA();
                s[j].display();
            }
        }
    }
}
```

Output:

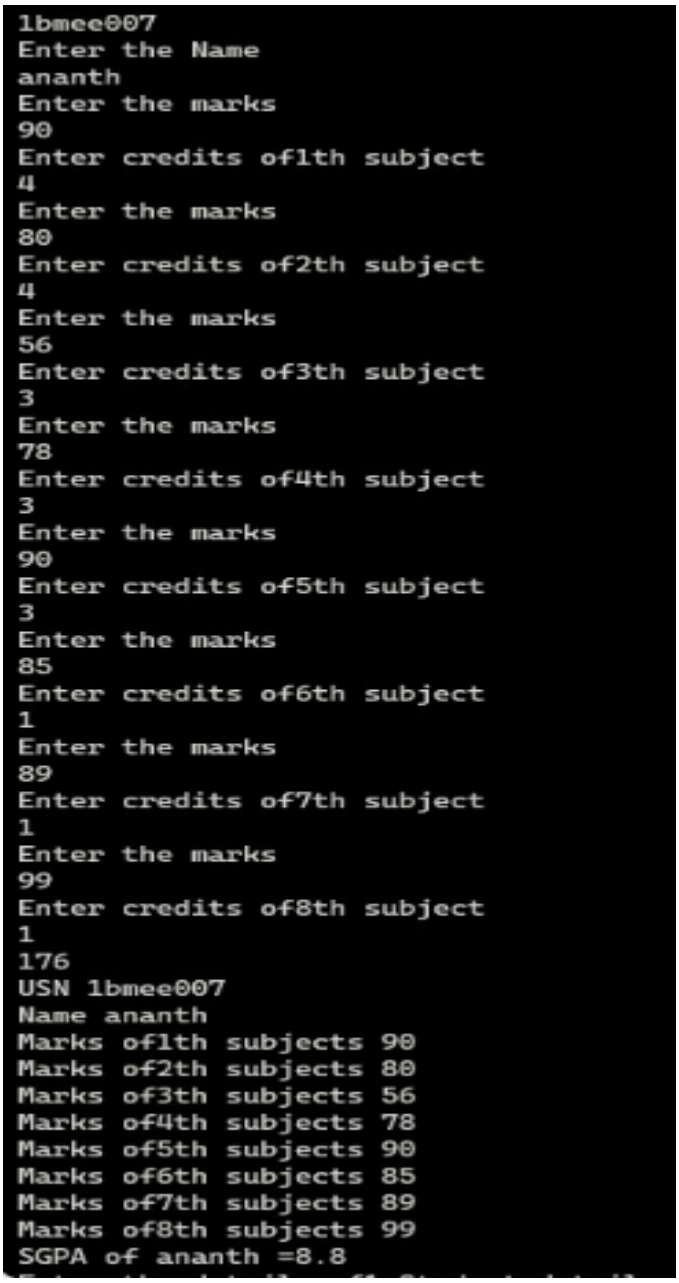
```
Enter the details of 1 Student Details
Enter the USN
1mw22
Enter the name
meme
Enter the marks
90
Enter credits of 1st subject
4
Enter the marks
90
Enter the credits of 2th subject
4
Enter the marks
90
Enter the credits of 3th subject
3
Enter the marks
90
Enter the credits of 4th subject
3
Enter the marks
85
Enter the credits of 5th subject
3
Enter the marks
90
Enter the credits of 6th subject
1
Enter the marks
90
Enter the credits of 7th subject
1
Enter the marks
90
Enter the credits of 8th subject
1
USN 1mw22
Name meme
Marks of 1th subject 90
Marks of 2th subject 90
Marks of 3th subject 90
Marks of 4th subject 90
Marks of 5th subject 85
Marks of 6th subject 90
Marks of 7th subject 90
Marks of 8th subject 90
SGPA of meme = 9.85
```

Code:

```
import java.util.Scanner;
class StudentDetails
{
   String name, usn;
   int marks[]=new int[8];
   int credits[]=new int[8];
   double sgpa,totcreds=0.0;


void getdetails()
{

   Scanner sc= new Scanner(System.in);
   System.out.println("Enter the USN \t");
   this.usn=sc.next();
   System.out.println("Enter the Name \t");
   this.name=sc.next();
   for(int i=0;i<8;i++)
   {
      System.out.println("Enter the marks of "+(i+1)+"th subject\t");
      this.marks[i]=sc.nextInt();
      System.out.println("Enter credits of"+(i+1)+"th subject \t");
      this.credits[i]=sc.nextInt();
      this.totcreds=this.totcreds + this.credits[i];
   }
}
void calculateSGPA()
{
   int v, grade=0;
   double sum=0.0;
   for(int k=0; k<8; k++)
   {
      v=(this.marks[k])/10;
      switch(v)
      {
         case 10: grade=10;
               break;
         case 9:
         case 8:
         case 7:
         case 6:
         case 5:
         case 4: grade=v+1;
               break;
         case 3:
         case 2:
```

```
            case 1: System.out.println("Grade ineligible for the "+(k+1)+"th subject \n");
                    break;

        }
        sum=sum+(grade*credits[k]);
    }
     this.sgpa= sum/this.totcreds;
    System.out.println(sum);
}
void display()
{
    System.out.println("USN"+" "+usn);
    System.out.println("Name"+" "+name);
    for(int m=0;m<8;m++)
    {
        System.out.println("Marks of"+(m+1)+"th subjects"+" "+marks[m]);
    }
    System.out.println("SGPA of  "+this.name+"  ="+this.sgpa);
}
}
class Student
{
    public static void main(String args[])
    {
        StudentDetails s1[]=new StudentDetails[3];
        for(int j=0;j<3;j++)
        {
            s1[j]=new StudentDetails();
            for(int n=0;n<3;n++)
            {
                System.out.println("Enter the details of  "+(n+1)+"th student \n");
                s1[j].getdetails();
                s1[j].calculateSGPA();
                s1[j].display();
            }
        }
    }
}
```

```
C:\Users\bmsce\Desktop\23CS028>java Student
Enter the details of1 Student details
Enter the USN
11wwee
Enter the Name
wewe
Enter the marks
90
Enter credits of1th subject
4
Enter the marks
90
Enter credits of2th subject
4
Enter the marks
90
Enter credits of3th subject
3
Enter the marks
90
Enter credits of4th subject
3
Enter the marks
85
Enter credits of5th subject
3
Enter the marks
90
Enter credits of6th subject
1
Enter the marks
90
Enter credits of7th subject
1
Enter the marks
90
Enter credits of8th subject
1
197
USN 11wwee
Name wewe
Marks of1th subjects 90
Marks of2th subjects 90
Marks of3th subjects 90
Marks of4th subjects 90
Marks of5th subjects 85
Marks of6th subjects 90
Marks of7th subjects 90
Marks of8th subjects 90
SGPA of wewe =9.85
Enter the details of1 Student details
```

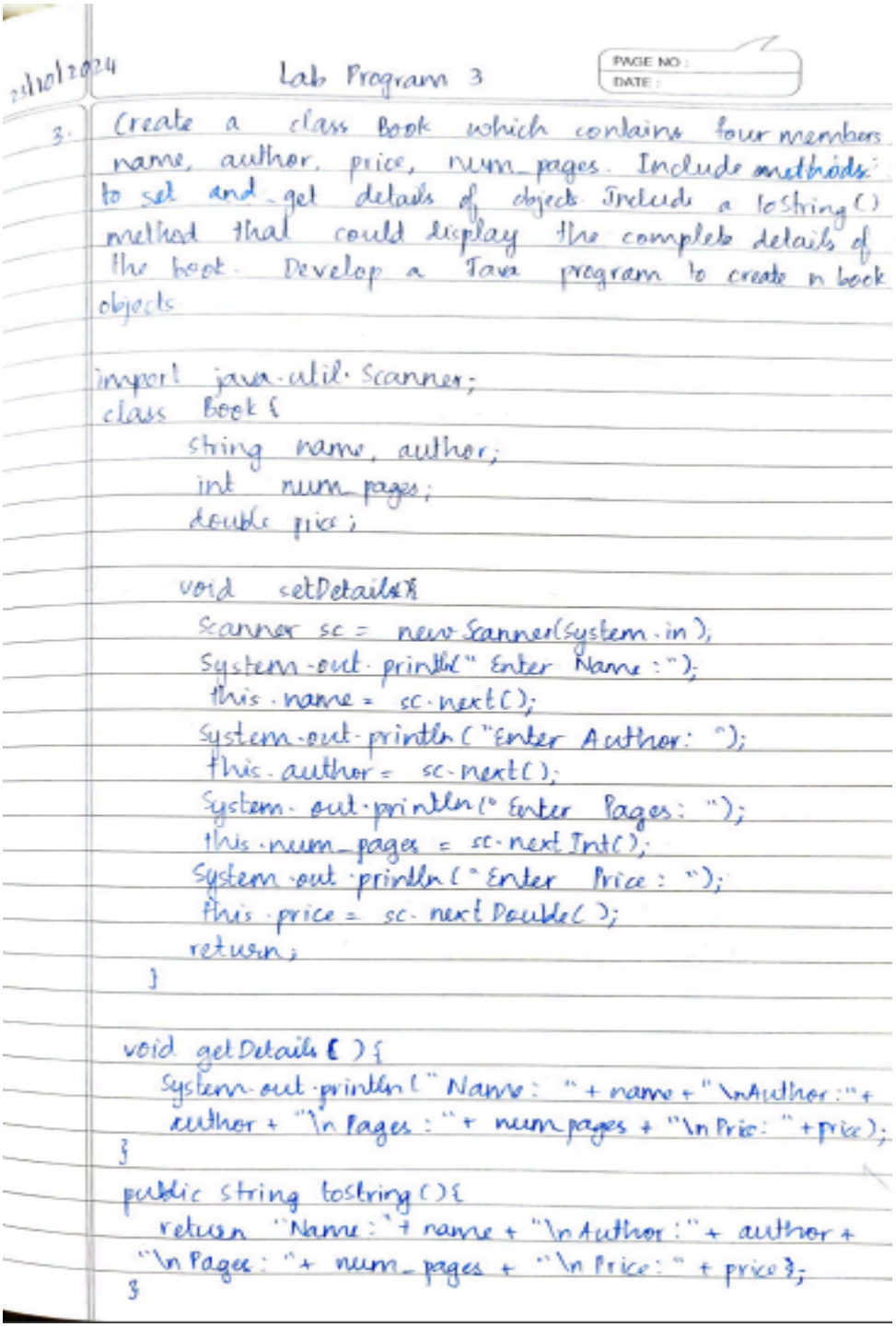

1bmee007
Enter the Name
ananth
Enter the marks
90
Enter credits of1th subject
4
Enter the marks
80
Enter credits of2th subject
4
Enter the marks
56
Enter credits of3th subject
3
Enter the marks
78
Enter credits of4th subject
3
Enter the marks
90
Enter credits of5th subject
3
Enter the marks
85
Enter credits of6th subject
1
Enter the marks
89
Enter credits of7th subject
1
Enter the marks
99
Enter credits of8th subject
1
176
USN 1bmee007
Name ananth
Marks of1th subjects 90
Marks of2th subjects 80
Marks of3th subjects 56
Marks of4th subjects 78
Marks of5th subjects 90
Marks of6th subjects 85
Marks of7th subjects 89
Marks of8th subjects 99
SGPA of ananth =8.8

```
Enter the details of  1th student

Enter the USN
21IS033
Enter the Name
Abhishek
Enter the marks of 1th subject
69
Enter credits of1th subject
4
Enter the marks of 2th subject
79
Enter credits of2th subject
4
Enter the marks of 3th subject
74
Enter credits of3th subject
3
Enter the marks of 4th subject
88
Enter credits of4th subject
3
Enter the marks of 5th subject
92
Enter credits of5th subject
3
Enter the marks of 6th subject
87
Enter credits of6th subject
1
Enter the marks of 7th subject
95
Enter credits of7th subject
1
Enter the marks of 8th subject
82
Enter credits of8th subject
1
169.0
USN 21IS033
Name Abhishek
Marks of1th subjects 69
Marks of2th subjects 79
Marks of3th subjects 74
Marks of4th subjects 88
Marks of5th subjects 92
Marks of6th subjects 87
Marks of7th subjects 95
Marks of8th subjects 82
SGPA of  Abhishek  =8.45
```

Program 3

Book Details

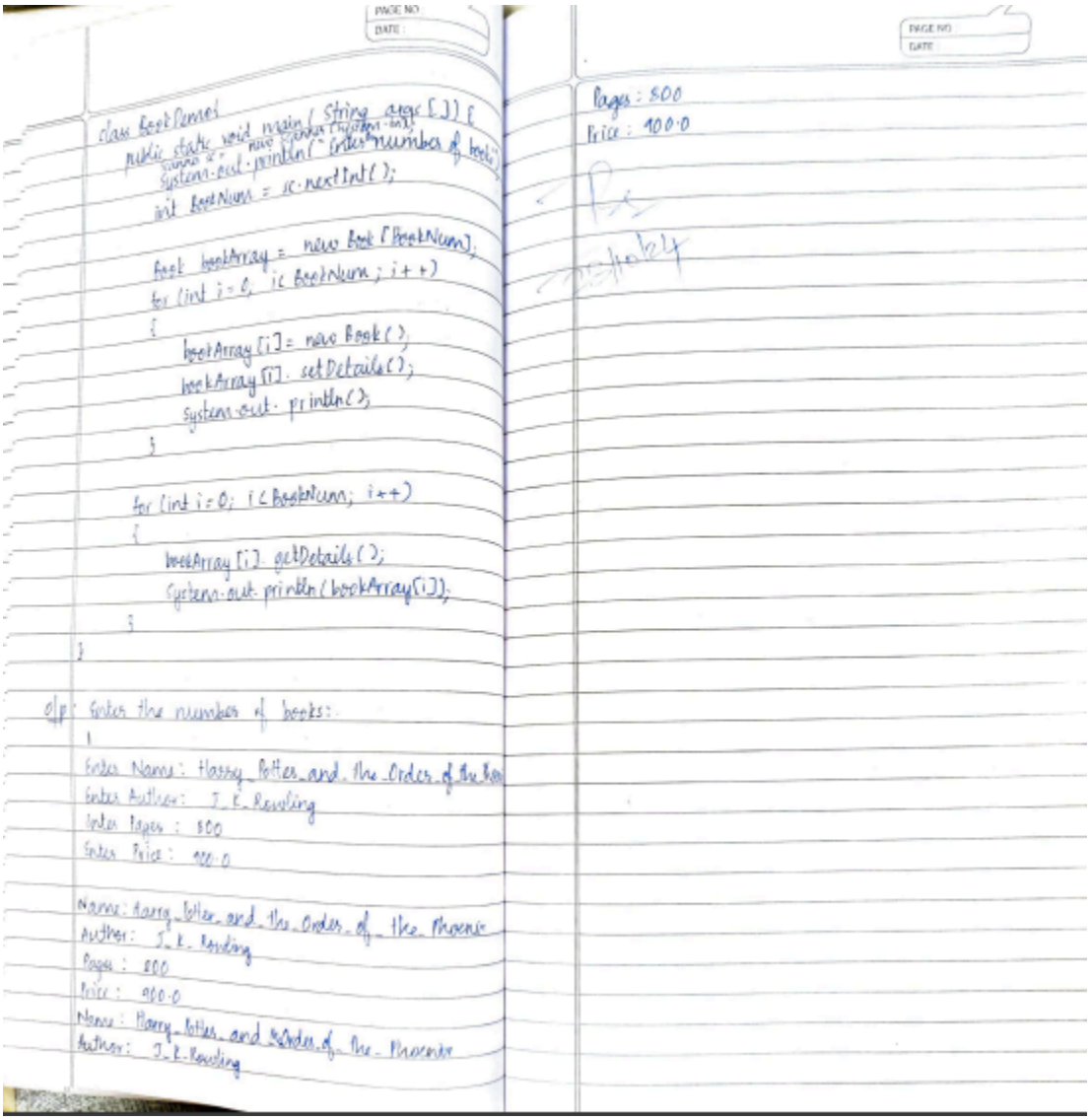
Algorithm:



Lab Program 3

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

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

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

void getDetails(){
   System.out.println(" Name : " + name + "\nAuthor :" +
   author + "\n Pages :" + num_pages + "\n Price: " + price);
}
public String toString(){
   return "Name :" + name + "\n Author :" + author +
  "\n Pages : " + num_pages + "\n Price :" + price};
}
```



Code:

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

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

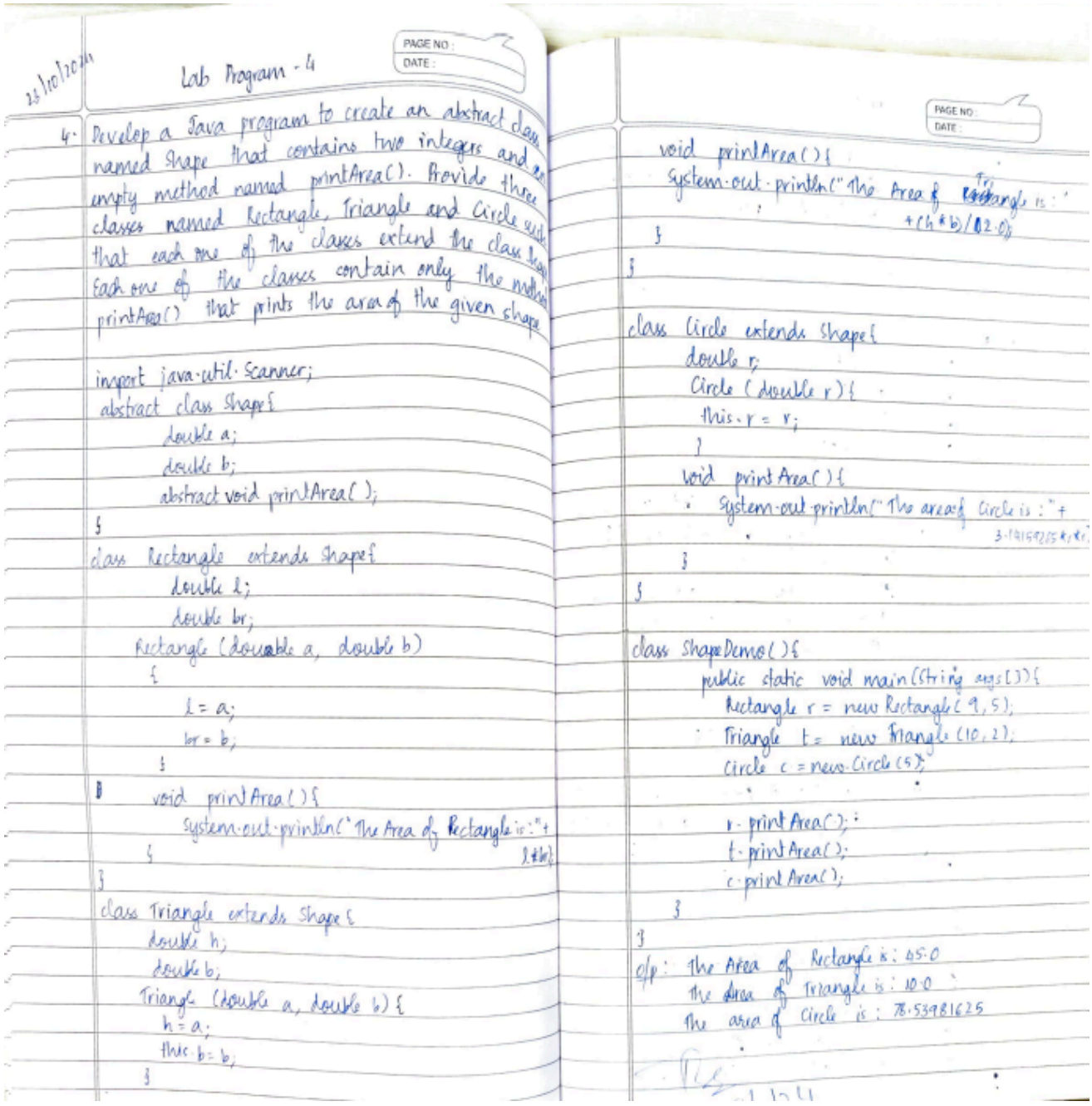
```
        this.price = sc.nextDouble();
        return;
    }

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

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

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

            System.out.print("Name: Amith R");
            System.out.print("USN: 1BM23CS028");
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int bookNum = sc.nextInt();

        Book bookArray[] = new Book[bookNum];

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

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

```
C:\23CS028>javac BookDemo.java

C:\23CS028>java BookDemo
Name: Amith R
USN: 1BM23CS028
Enter the number of books:
3
Enter Name:Harry_Potter_and_the_Order_of_the_Phoenix
Enter Author:J_K_Rowling
Enter Pages:800
Enter Price:900.0

Enter Name:Too_Late
Enter Author:Collen_Hoover
Enter Pages:450
Enter Price:500.0

Enter Name:One_Night_at_the_Call_Centre
Enter Author:Chethan_Bhagat
Enter Pages:700
Enter Price:800.0

Name: Harry_Potter_and_the_Order_of_the_Phoenix
Author: J_K_Rowling
Pages: 800
Price: 900.0
Name: Harry_Potter_and_the_Order_of_the_Phoenix
Author: J_K_Rowling
Pages: 800
Price: 900.0
Name: Too_Late
Author: Collen_Hoover
Pages: 450
Price: 500.0
Name: Too_Late
Author: Collen_Hoover
Pages: 450
Price: 500.0
Name: One_Night_at_the_Call_Centre
Author: Chethan_Bhagat
Pages: 700
Price: 800.0
Name: One_Night_at_the_Call_Centre
Author: Chethan_Bhagat
Pages: 700
Price: 800.0
```

Program 4

Abstract Class shape

Algorithm:



Lab Program - 4

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 extend the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;
abstract class Shape{
    double a;
    double b;
    abstract void printArea();
}
class Rectangle extends Shape{
    double l;
    double br;
    Rectangle (double a, double b)
    {
        l = a;
        br = b;
    }
    void printArea(){
        System.out.println("The Area of Rectangle is:"+
        l*b);
    }
}
class Triangle extends Shape{
    double h;
    double b;
    Triangle (double a, double b){
        h = a;
        this.b = b;
    }
    void printArea(){
        System.out.println("The Area of Triangle is:"+
        (h*b)/2.0);
    }
}
class Circle extends Shape{
    double r;
    Circle (double r){
        this.r = r;
    }
    void printArea(){
        System.out.println("The area of Circle is:"+
        3.14159265*r*r);
    }
}
class ShapeDemo(){
    public static void main(String args[]){
        Rectangle r = new Rectangle(9,5);
        Triangle t = new Triangle(10,2);
        Circle c = new Circle(5);
        r.printArea();
        t.printArea();
        c.printArea();
    }
}
```

o/p: The Area of Rectangle is: 45.0
The Area of Triangle is: 10.0
The area of Circle is: 78.53981625

Code:

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

        void printArea(){
                System.out.println("The Area of the rectangle is: "+l*br);
}
}
class Triangle extends Shape{
        double h;
        double b;
        Triangle(double a, double b){
                h=a;
                this.b=b;
        }
        void printArea(){
                System.out.println("The Area of the triangle is: "+(h*b)/2.0);
        }
}
class Circle extends Shape{
        double r;
        Circle(double r){
                this.r=r;
        }
        void printArea(){
                System.out.println("The area of the Circle is: "+ 3.14159265*r*r);

        }
}
class ShapeDemo{
        public static void main(String args[]){
                System.out.println("Name: Amith R");
                System.out.println("USN: 1BM23CS028");
                Rectangle r = new Rectangle(9,5);
                Triangle t = new Triangle(10,2);
                Circle c = new Circle(5);

                r.printArea();
```

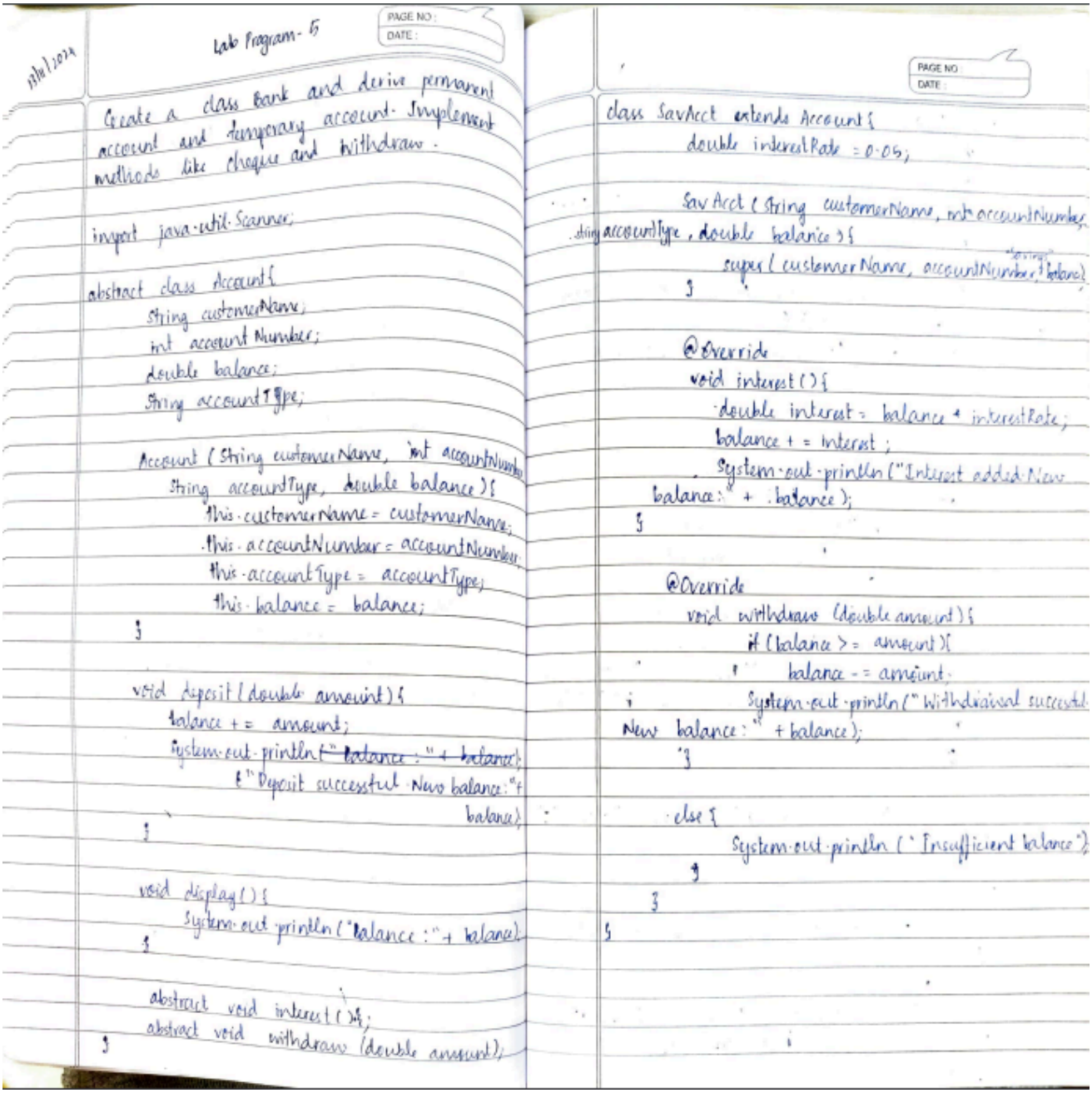
```
            t.printArea();
            c.printArea();

        }
}
```

```
C:\23CS028>java ShapeDemo
Name: Amith R
USN: 1BM23CS028
The Area of the rectangle is: 45.0
The Area of the triangle is: 10.0
The area of the Circle is: 78.53981625
```

Program 5

Bank Class

Algorithm:

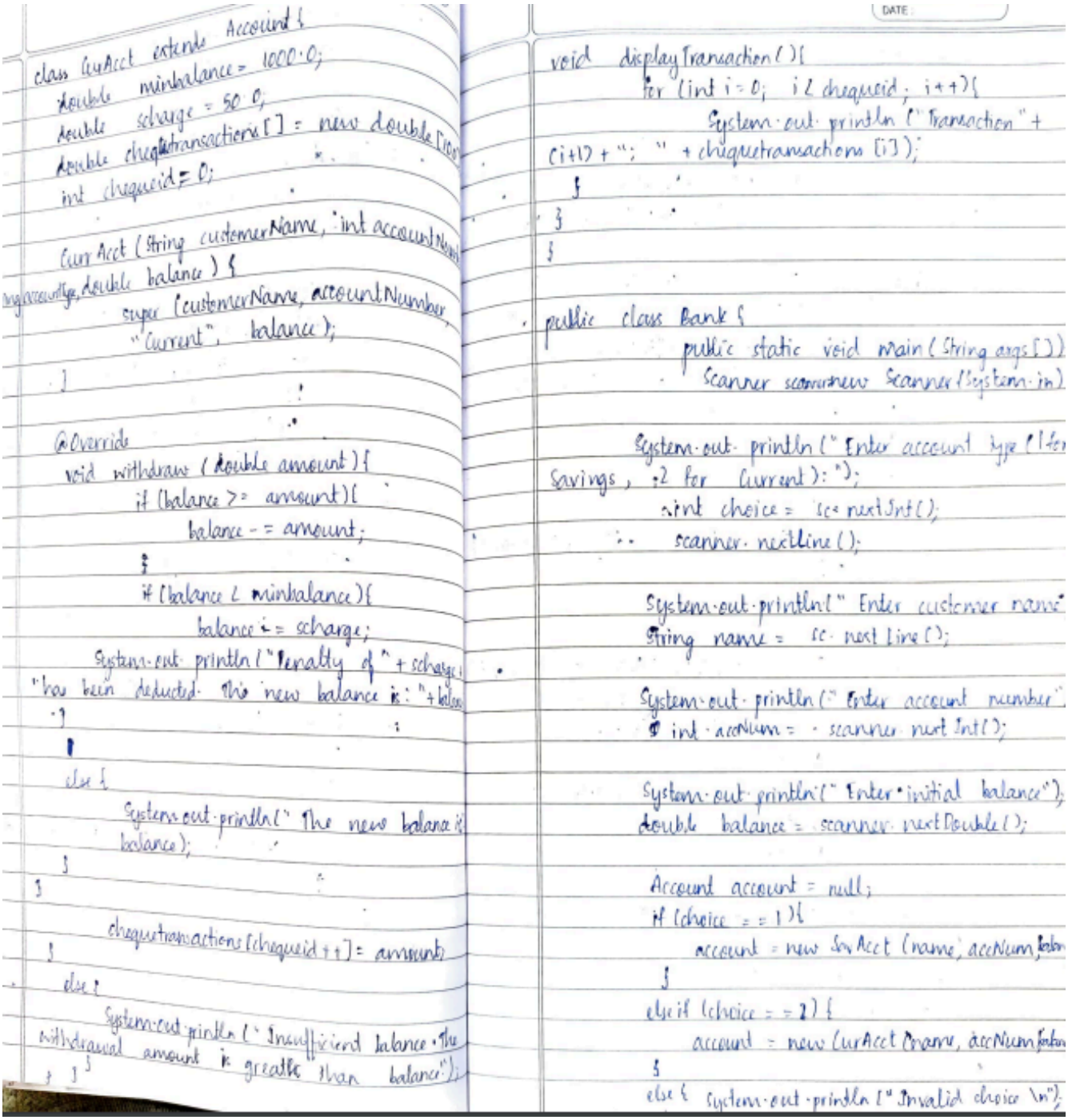


Lab Program - 5

Create a class bank and derive permanent account and temporary account. Implement methods like cheque and withdraw.

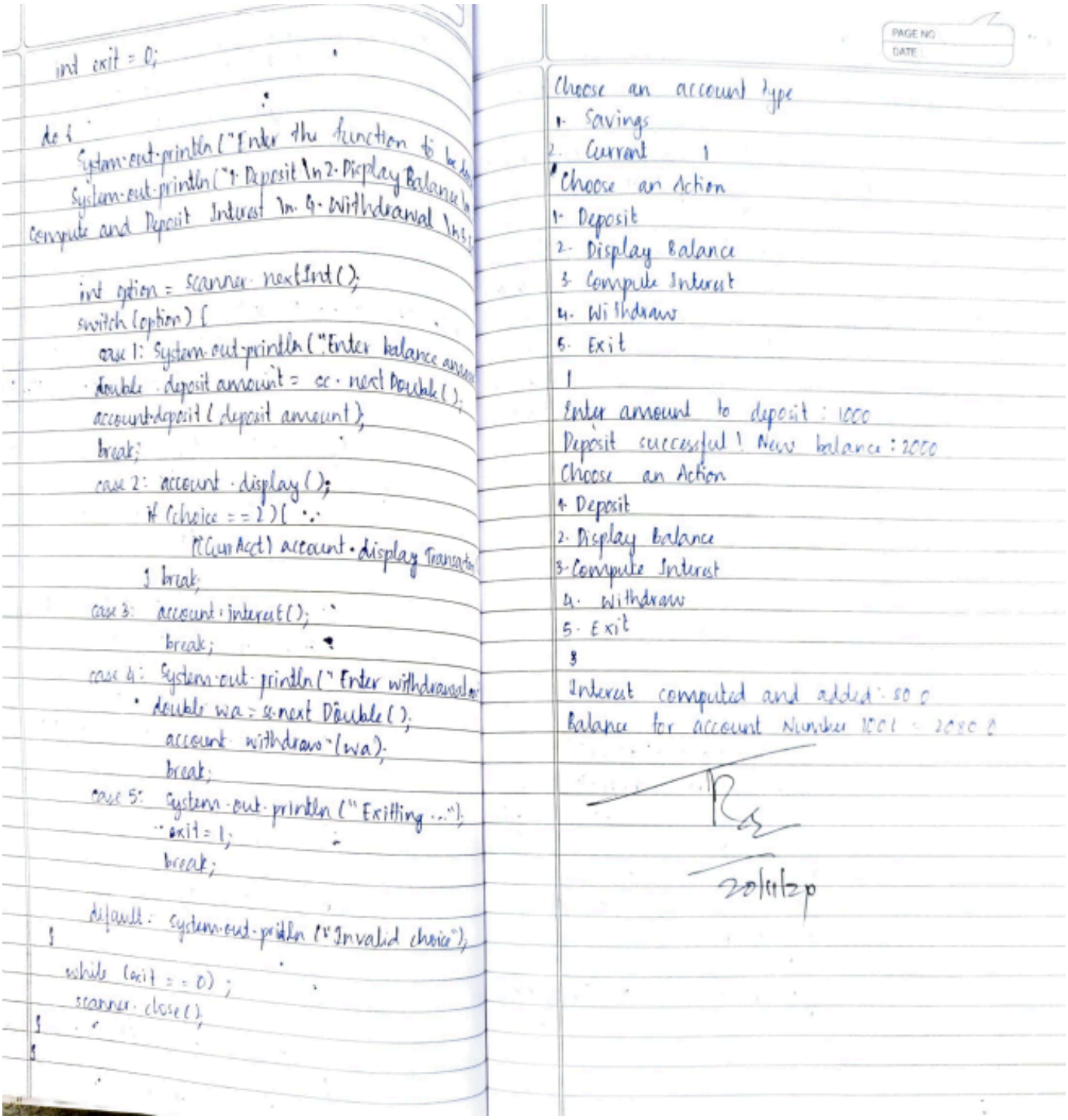
```
import java.util.Scanner;

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

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

    void deposit (double amount){
        balance += amount;
        System.out.println("~~balance : " + balance~~
            "Deposit successful. New balance:"+
                                    balance);
    }

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

    abstract void interest();
    abstract void withdraw (double amount);
}
```

```
class SavAcct extends Account{
    double interestRate = 0.05;

    SavAcct (String customerName, int accountNumber,
String accountType, double balance){
        super (customerName, accountNumber, balance);
    }

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

    @Override
    void withdraw (double amount){
        if (balance >= amount){
            balance -= amount;
            System.out.println("Withdrawal successful.
New balance:" + balance);
        }
        else{
            System.out.println("Insufficient balance");
        }
    }
}
```



```
class CurAcct extends Account {
    double minbalance = 1000.0;
    double scharge = 50.0;
    double chequetransactions[] = new double[10];
    int chequeid = 0;

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

    @Override
    void withdraw ( double amount ) {
        if (balance >= amount){
            balance -= amount;
        }
        if (balance < minbalance){
            balance -= scharge;
        System.out.println("Penalty of " + scharge +
"has been deducted. The new balance is: " + balance
        }
        else {
            System.out.println(" The new balance is
            balance);
        }
    }

        chequetransactions[chequeid++] = amount
    }
    else {
        System.out.println(" Insufficient balance. The
withdrawal amount is greater than balance.");
    }
}
```

```
void displayTransaction(){
        for (int i = 0; i < chequeid; i++){
            System.out.println("Transaction " +
(i+1) + ": " + chequetransactions[i]);
        }
    }
}

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

        System.out.println(" Enter account type (1 for
Savings, 2 for Current): ");
        int choice = sc.nextInt();
        scanner.nextLine();

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

        System.out.println(" Enter account number:
        int accNum = scanner.nextInt();

        System.out.println(" Enter initial balance:");
        double balance = scanner.nextDouble();

        Account account = null;
        if (choice == 1){
            account = new SavAcct (name, accNum, balan
        }
        else if (choice == 2) {
            account = new CurAcct (name, accNum, balan
        }
        else { System.out.println(" Invalid choice \n");
```



```
int exit = 0;

do {
    System.out.println("Enter the function to be
    System.out.println("1. Deposit \n2. Display Balance \n
Compute and Deposit Interest \n4. Withdrawal \n5.

    int option = scanner.nextInt();
    switch (option) {
        case 1: System.out.println("Enter balance amount
        double deposit amount = sc.nextDouble();
        account.deposit(deposit amount);
        break;
        case 2: account.display();
            if (choice == 2){
                ((CurrAcct) account.display Transaction
            } break;
        case 3: account.interest();
            break;
        case 4: System.out.println("Enter withdrawal
            double wa = s.nextDouble();
            account.withdraw(wa);
            break;
        case 5: System.out.println("Exiting...");
            exit = 1;
            break;

        default: System.out.println("Invalid choice");
    }
    while (exit == 0);
    scanner.close();
}
}
```

```
Choose an account type
1. Savings
2. Current
1
Choose an Action
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Exit
1
Enter amount to deposit : 1000
Deposit successful! New balance : 2050
Choose an Action
1. Deposit
2. Display Balance
3. Compute Interest
4. Withdraw
5. Exit
3
Interest computed and added: 50.0
Balance for account Number 1001 : 2050.0
```

Code:

```
import java.util.Scanner;
import java.util.Random;

abstract class BankAccount {
    protected String CustName;
    protected String AccNo;
```

```
    protected double balance;


    Boolean generateAccNo() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name of the account holder: ");
        this.CustName = sc.nextLine();
        Random r = new Random();
        this.AccNo = "";
        for (int i = 0; i < 11; i++) {
            int d = r.nextInt(10);
            this.AccNo += d;
        }
        System.out.println("The account number of the holder is: " + this.AccNo);
        return true;
    }

    // Method to get balance
    void getBalance() {
                    Scanner sc = new Scanner(System.in);
                    System.out.println("Enter the balance for "+this.CustName);
                    this.balance = sc.nextDouble();
        System.out.println("The current balance for " + CustName + " (" + AccNo + "): " + this.balance);
    }

    abstract void displayBalance();
    abstract void withdraw();
    abstract void deposit();
}

class SavingsAccount extends BankAccount {

    @Override
    void displayBalance() {
        System.out.println("Savings Account Balance for " + CustName + " (" + AccNo + "): " +
balance);
    }

    @Override
    void withdraw() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the amount to withdraw:");
        double amount = sc.nextDouble();
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance!");
```

```
        }
    }

    @Override
    void deposit() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the amount to deposit:");
        double amount = sc.nextDouble();
        balance += amount;
        System.out.println("Deposit successful. New balance: " + balance);
    }
}

class CurrentAccount extends BankAccount {
    private double minBalance = 5000.0;
    private double serviceCharge = 100.0;

    @Override
    void displayBalance() {
        System.out.println("Current Account Balance for " + CustName + " (" + AccNo + "): " +
balance);
    }

    @Override
    void withdraw() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the amount to withdraw:");
        double amount = sc.nextDouble();
        if (balance >= amount) {
            balance -= amount;
            if (balance < minBalance) {
                System.out.println("Balance below minimum. Applying service charge.");
                balance -= serviceCharge;
            }
            System.out.println("Withdrawal successful. New balance: " + balance);
        } else {
            System.out.println("Insufficient balance!");
        }
    }

    @Override
    void deposit() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the amount to deposit:");
        double amount = sc.nextDouble();
        balance += amount;
        System.out.println("Deposit successful. New balance: " + balance);
    }
```

```
    void chequeTransfer(BankAccount receiver, double amount) {
        if (this == receiver) {
            System.out.println("Cannot transfer cheque to the same account.");
        } else if (balance >= amount) {
            this.balance -= amount;
            receiver.balance += amount;
            System.out.println("Cheque transfer successful.");
            System.out.println("Donor("+this.CustName+")'s new balance: " + this.balance);
                                System.out.println("Recipient("+receiver.CustName+")'s new balance:
"+receiver.balance );
        } else {
            System.out.println("Insufficient balance for cheque transfer!");
        }
    }
}

public class BankDriver {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of bank customers:");
        int n = sc.nextInt();
        sc.nextLine(); // Consume the newline
        BankAccount[] accounts = new BankAccount[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Choose account type for customer " + (i + 1) + ":");
            System.out.println("1. Savings Account\n2. Current Account");
            int accountType = sc.nextInt();
            sc.nextLine(); // Consume the newline

            if (accountType == 1) {
                accounts[i] = new SavingsAccount();
            } else if (accountType == 2) {
                accounts[i] = new CurrentAccount();
            } else {
                System.out.println("Invalid choice! Defaulting to Savings Account.");
                accounts[i] = new SavingsAccount();
            }


            if (accounts[i].generateAccNo()) {
                System.out.println("Enter initial balance:");
                accounts[i].balance = sc.nextDouble();
                sc.nextLine(); // Consume the newline
            }
        }
```

```
        boolean exit = false;
        while (!exit) {
            System.out.println("\nMain Menu:");
            System.out.println("1. Display Balance");
            System.out.println("2. Withdraw");
            System.out.println("3. Deposit");
            System.out.println("4. Cheque Transfer");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    for (BankAccount account : accounts) {
                        account.displayBalance();
                    }
                    break;

                case 2:
                    System.out.println("Enter customer index for withdrawal (1 to " + n + "):");
                    int withdrawIndex = sc.nextInt() - 1;
                    if (withdrawIndex >= 0 && withdrawIndex < n) {
                        accounts[withdrawIndex].withdraw();
                    } else {
                        System.out.println("Invalid index!");
                    }
                    break;

                case 3:
                    System.out.println("Enter customer index for deposit (1 to " + n + "):");
                    int depositIndex = sc.nextInt() - 1;
                    if (depositIndex >= 0 && depositIndex < n) {
                        accounts[depositIndex].deposit();
                    } else {
                        System.out.println("Invalid index!");
                    }
                    break;

                case 4:
                    System.out.println("Enter donor index (1 to " + n + "):");
                    int donorIndex = sc.nextInt() - 1;
                    System.out.println("Enter recipient index (1 to " + n + "):");
                    int recipientIndex = sc.nextInt() - 1;
                    if (donorIndex >= 0 && donorIndex < n && recipientIndex >= 0 && recipientIndex < n)
{
                        System.out.println("Enter cheque amount:");
                        double chequeAmount = sc.nextDouble();
                    } else {
```

```
                System.out.println("Invalid indices!");
            }
            break;

        case 5:
            exit = true;
            break;

        default:
            System.out.println("Invalid choice! Try again.");
      }
    }

    System.out.println("Exiting program. Thank you!");
  }
}
```

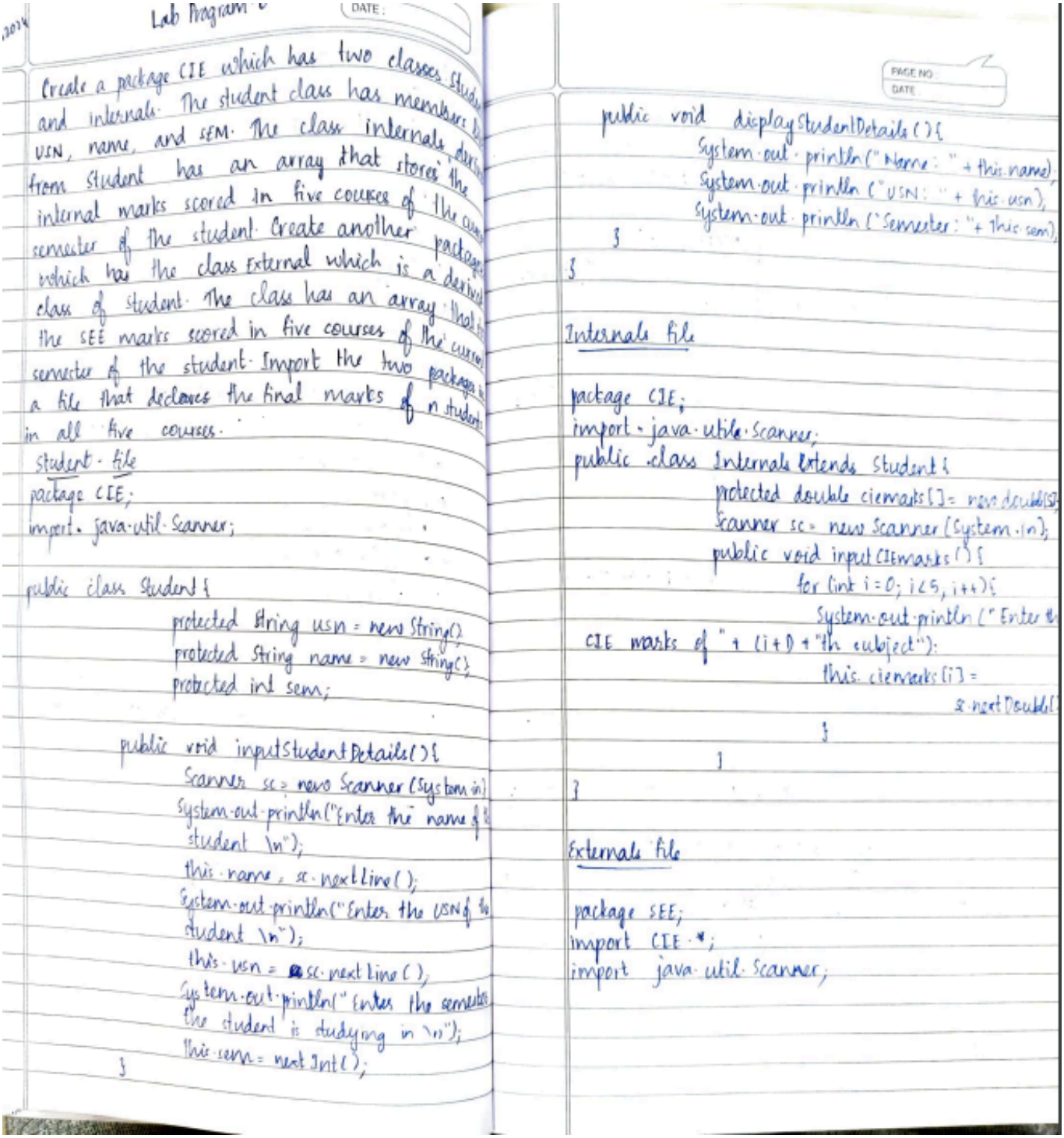
```
Enter the number of bank customers:
3
Choose account type for customer 1:
1. Savings Account
2. Current Account
1
Enter the name of the account holder:
Rohan
The account number of the holder is: 57336102868
Enter initial balance:
100000
Choose account type for customer 2:
1. Savings Account
2. Current Account
1
Enter the name of the account holder:
Ahaan
The account number of the holder is: 20515556915
Enter initial balance:
150000
Choose account type for customer 3:
1. Savings Account
2. Current Account
2
Enter the name of the account holder:
Arjun
The account number of the holder is: 26079682694
Enter initial balance:
300000

Main Menu:
1. Display Balance
2. Withdraw
3. Deposit
4. Cheque Transfer
5. Exit
Enter your choice: 1
Savings Account Balance for Rohan (57336102868): 100000.0
Savings Account Balance for Ahaan  (20515556915): 150000.0
Current Account Balance for Arjun (26079682694): 300000.0
```

```
Main Menu:
1. Display Balance
2. Withdraw
3. Deposit
4. Cheque Transfer
5. Exit
Enter your choice: 2
Enter customer index for withdrawal (1 to 3):
2
Enter the amount to withdraw:
10000
Withdrawal successful. New balance: 140000.0

Main Menu:
1. Display Balance
2. Withdraw
3. Deposit
4. Cheque Transfer
5. Exit
Enter your choice: 3
Enter customer index for deposit (1 to 3):
1
Enter the amount to deposit:
100000
Deposit successful. New balance: 200000.0

Main Menu:
1. Display Balance
2. Withdraw
3. Deposit
4. Cheque Transfer
5. Exit
Enter your choice: 4
Enter donor index (1 to 3):
3
Enter recipient index (1 to 3):
2
Enter cheque amount:
25000
Cheque transfer successful.
Donor(Arjun)'s new balance: 275000.0
Recipient(Ahaan )'s new balance: 165000.0
```

Program 6:

Package

Algorithm:



Lab Program

Create a package CIE which has two classes Student and Internals. The student class has members usn, name, and 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 which has the class External which is a derived class of student. The 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.

Student file

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

public class Student {
        protected String usn = new String();
        protected String name = new String();
        protected int sem;

    public void inputStudentDetails(){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name of the
        student \n");
        this.name = sc.nextLine();
        System.out.println("Enter the USN of the
        student \n");
        this.usn = sc.nextLine();
        System.out.println("Enter the semester in
        the student is studying in \n");
        this.sem = nextInt();
    }
    public void displayStudentDetails(){
        System.out.println("Name: " + this.name);
        System.out.println("USN: " + this.usn);
        System.out.println("Semester: " + this.sem);
    }
}
```

Internals file

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

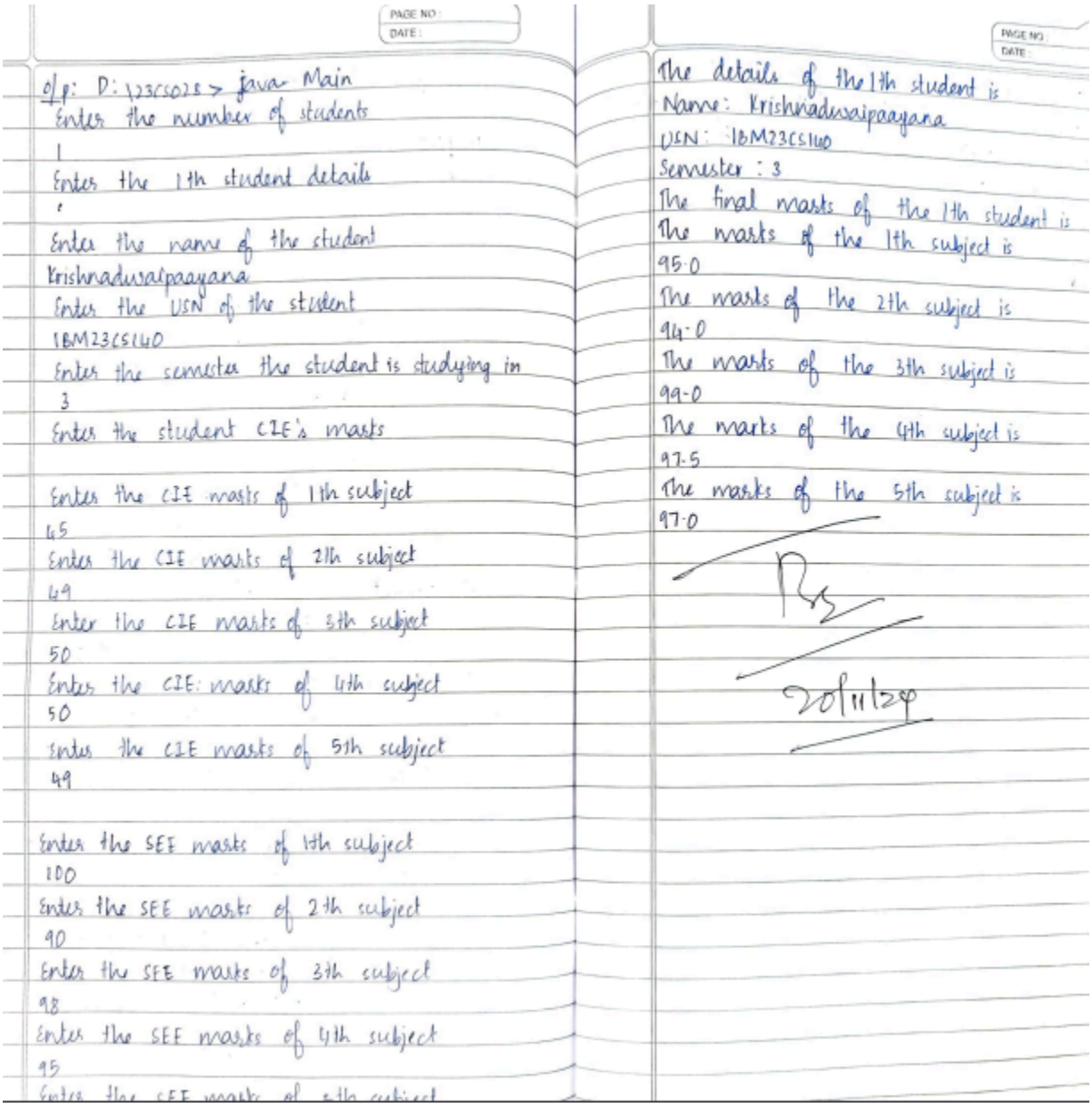
Externals file

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

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

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

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


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

}
```

Main.java file

```
import SEE.Externals;
import java.util.Scanner;

class Main {
    public static void main (String args[]){
        int n;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter number of student
\n");
        n = sc.nextInt();
        Externals e = new Externals [n];
        for (int i = 0; i<n; i++){
            e[i] = new Externals();
            System.out.println ("Enter the" + (i+1)+
"th student details \n");
            e[i].inputStudentDetails();
            System.out.println (" Enter the " + (i+1)+
"th student's CIE marks \n");
            e[i].inputCIEmarks();
            System.out.println (" Enter the " + (i+1)+
"th student's SEE marks \n");
            e[i].inputSEEmarks();
            System.out.println(" the details of th
" + (i+1) + "th student is \n");
            e[i].displayStudentDetails();
            e[i].definefinalmarks();
            System.out.println(" the final marks of th
" + (i+1) + "th student is");
            e[i].displayfinalmarks();
    }
```



o/p: D:\23CS025> java Main
Enter the number of students
1
Enter the 1th student details
Enter the name of the student
Krishnadwaipayana
Enter the USN of the student
1BM23CS140
Enter the semester the student is studying in
3
Enter the student CIE's masts

Enter the CIE masts of 1th subject
45
Enter the CIE masts of 2th subject
49
Enter the CIE masts of 3th subject
50
Enter the CIE masts of 4th subject
50
Enter the CIE masts of 5th subject
49

Enter the SEE masts of 1th subject
100
Enter the SEE masts of 2th subject
90
Enter the SEE masts of 3th subject
98
Enter the SEE masts of 4th subject
95
Enter the SEE masts of 5th subject

The details of the 1th student is
Name: Krishnadwaipayana
USN: 1BM23CS140
Semester : 3
The final masts of the 1th student is
The masts of the 1th subject is
95.0
The masts of the 2th subject is
94.0
The masts of the 3th subject is
99.0
The masts of the 4th subject is
97.5
The masts of the 5th subject is
97.0

20/11/24

Code:
Student.java file

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

public class Student {
		protected String usn = new String();
		protected String name = new String();
		protected int sem;

		public void inputStudentDetails(){
```

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

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

Internals.java file:

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

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

Externals.java file:

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

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

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

```
                }
        }

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

        }

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

        }
}
```

Main.java file

```
import SEE.Externals;
import java.util.Scanner;

class Main{
        public static void main(String args[]){
                int n;
                Scanner sc = new Scanner(System.in);
                System.out.println("Name: Amith R");
                System.out.println("USN: 1BM23CS028");
                System.out.println("Enter the number of students \t");
                n= sc.nextInt();
                Externals e[] = new Externals[n];
                for(int i=0; i<n; i++){
                        e[i] = new Externals();
                        System.out.println("Enter the" +(i+1)+"th student details \n");
                        e[i].inputStudentDetails();
                        System.out.println("Enter the" +(i+1)+"th student's CIE marks\n");
                        e[i].inputCIEmarks();
                        System.out.println("Enter the" +(i+1)+"th student's SEE marks\n");
                        e[i].inputSEEmarks();
                        System.out.println("The details of the" +(i+1)+"th student is");
                        e[i].displayStudentDetails();
                        e[i].definefinalmarks();
                        System.out.println("The final marks of the" +(i+1)+"th student is");
                        e[i].displayfinalmarks();
                }
```

```
        }
}
```

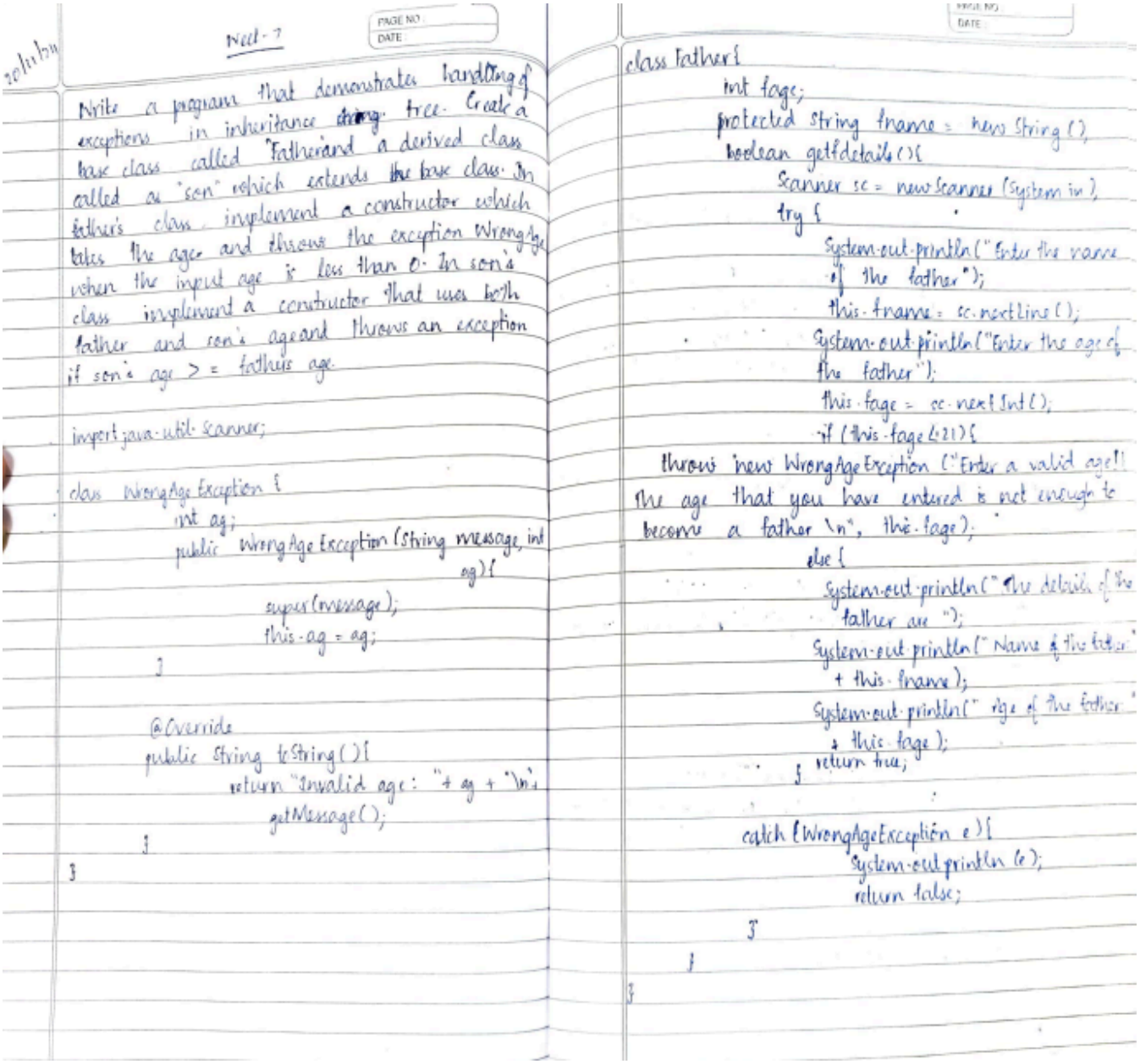
```
D:\23CS028>java Main
Name: Amith R
USN: 1BM23CS028
Enter the number of students
2
Enter the1th student details

Enter the name of the student

Krishnadwaipaayana
Enter the USN of the student

1BM23CS140
Enter the semester the student is studying in

3
Enter the1th student's CIE marks

Enter the CIE marks of1th subject
45
Enter the CIE marks of2th subject
49
Enter the CIE marks of3th subject
50
Enter the CIE marks of4th subject
50
Enter the CIE marks of5th subject
49
Enter the1th student's SEE marks

Enter the SEE marks of1th subject
100
Enter the SEE marks of2th subject
90
Enter the SEE marks of3th subject
98
Enter the SEE marks of4th subject
95
Enter the SEE marks of5th subject
96
The details of the1th student is
Name: Krishnadwaipaayana
USN: 1BM23CS140
Semester: 3
The final marks of the1th student is
The final marks of the student is

The marks of the1th subject is
95.0
The marks of the2th subject is
94.0
The marks of the3th subject is
99.0
The marks of the4th subject is
97.5
The marks of the5th subject is
97.0
```

```
Enter the2th student details

Enter the name of the student

Gopi_Bahu
Enter the USN of the student

1BM23CS101
Enter the semester the student is studying in

3
Enter the2th student's CIE marks

Enter the CIE marks of1th subject
40
Enter the CIE marks of2th subject
41
Enter the CIE marks of3th subject
42
Enter the CIE marks of4th subject
43
Enter the CIE marks of5th subject
44
Enter the2th student's SEE marks

Enter the SEE marks of1th subject
90
Enter the SEE marks of2th subject
78
Enter the SEE marks of3th subject
89
Enter the SEE marks of4th subject
92
Enter the SEE marks of5th subject
91
The details of the2th student is
Name: Gopi_Bahu
USN: 1BM23CS101
Semester: 3
The final marks of the2th student is
The final marks of the student is

The marks of the1th subject is
85.0
The marks of the2th subject is
80.0
The marks of the3th subject is
86.5
The marks of the4th subject is
89.0
The marks of the5th subject is
89.5
```

Program 7

Wrong Age Exception

Algorithm:

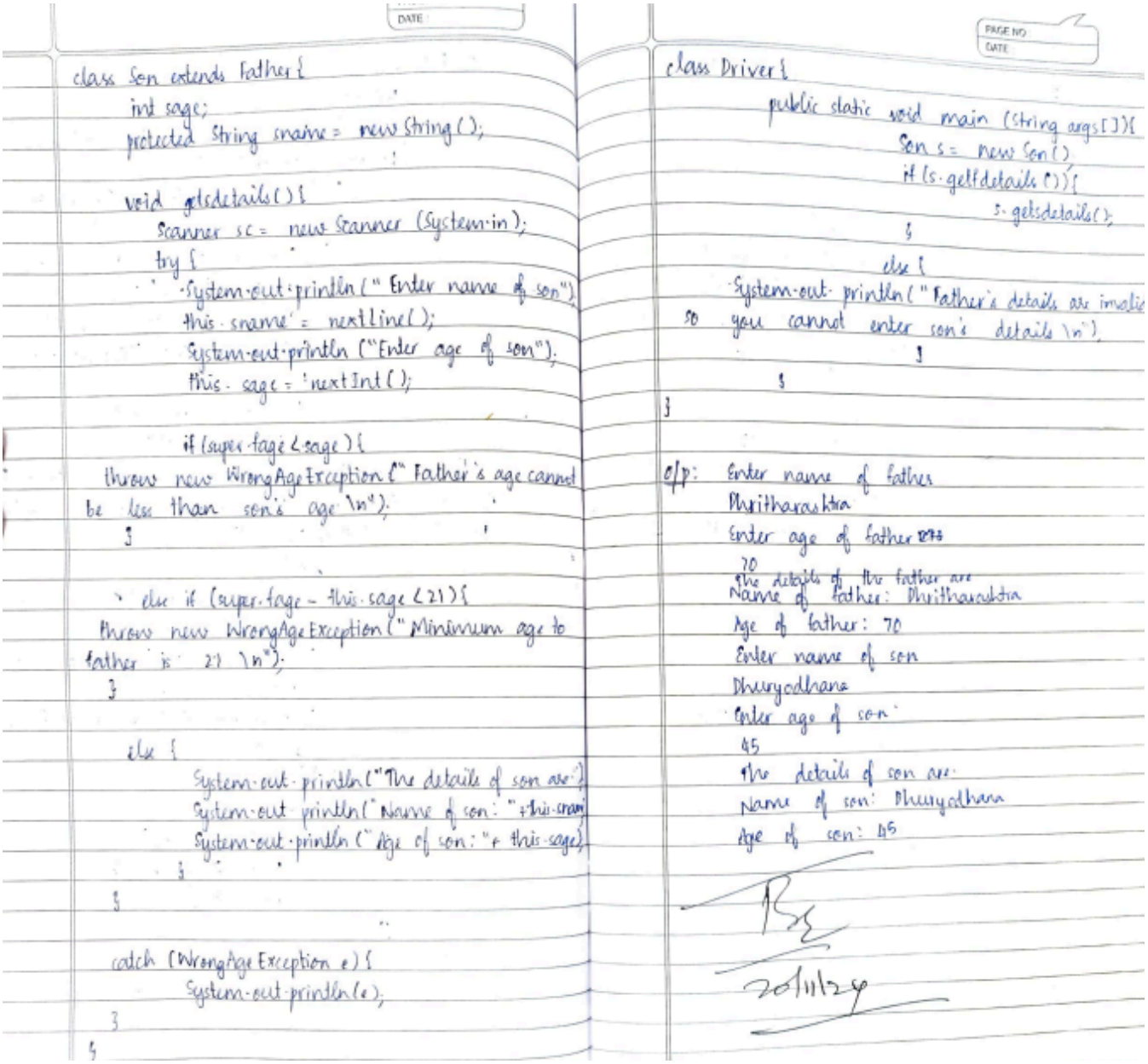


Week-7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and a derived class called as "son" which extends the base class. In father's 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 an exception if son's age >= father's age.

```
import java.util.Scanner;

class WrongAgeException {
    int ag;
    public WrongAgeException(String message, int ag){
        super(message);
        this.ag = ag;
    }

    @Override
    public String toString(){
        return "Invalid age: " + ag + "\n" +
            getMessage();
    }
}
```

```
class Father{
    int fage;
    protected String fname = new String();
    boolean getfdetails(){
        Scanner sc = new Scanner(System.in);
        try {
            System.out.println("Enter the name
            of the father");
            this.fname = sc.nextLine();
            System.out.println("Enter the age of
            the father");
            this.fage = sc.nextInt();
            if (this.fage<21){
    throw new WrongAgeException("Enter a valid age!!
    The age that you have entered is not enough to
    become a father \n", this.fage);
            else {
                System.out.println("The details of the
                father are ");
                System.out.println("Name of the father"
                + this.fname);
                System.out.println("Age of the father"
                + this.fage);
            return true;
        }
        catch(WrongAgeException e){
            System.out.println(e);
            return false;
        }
    }
}
```



```
class Son extends Father{
    int sage;
    protected String sname = new String();

    void getsdetails(){
      Scanner sc = new Scanner (System.in);
      try {
        System.out.println(" Enter name of son");
        this.sname = nextLine();
        System.out.println("Enter age of son");
        this.sage = nextInt();

          if (super.fage < sage){
    throw new WrongAgeException(" Father's age cannot
be less than son's age \n");
      }

       else if (super.fage - this.sage < 21){
    throw new WrongAgeException(" Minimum age to
father is 21 \n");
      }

        else {
            System.out.println("The details of son are");
            System.out.println("Name of son: "+this.sname
            System.out.println("Age of son: "+ this.sage);
          }
      }

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

```
class Driver{
    public static void main (String args[]){
        Son s = new Son();
        if (s.getfdetails()){
            s.getsdetails();
        }
        else {
      System.out.println(" Father's details are invalid
so you cannot enter son's details \n");
        }
    }
}
```

```
o/p: Enter name of father
Phritharashtra
Enter age of father
70
The details of the father are
Name of father: Phritharashtra
Age of father: 70
Enter name of son
Dhuryodhana
Enter age of son
45
The details of son are:
Name of son: Dhuryodhana
Age of son: 45
```

20/11/24

Code:

```
import java.util.Scanner;

class WrongAgeExcpetion extends Exception{
  int ag;
  public WrongAgeExcpetion(String message, int ag){
  super(message);
  this.ag = ag;
  }

  @Override
```

```
    public String toString(){
        return "Invalid age: "+ ag +"\n"+getMessage();
    }

}

class Father{
    int fage;
    protected String fname = new String();
    boolean getfdetails(){
        Scanner sc = new Scanner(System.in);
        try{
            System.out.println("Enter the name of the father");
            this.fname = sc.nextLine();
            System.out.println("Enter the age of the father");
            this.fage = sc.nextInt();
            if(this.fage<21){
                throw new WrongAgeExcpetion("Enter a valid age!! Minimum age of
father must be 21 \n", this.fage);
            }
            else{
                System.out.println("The details of the father are");
                System.out.println("Name of the father: "+this.fname);
                System.out.println("Age of the father: "+this.fage);
                return true;
            }
        }
        catch (WrongAgeExcpetion e){
            System.out.println(e);
            return false;
        }
    }
}

class Son extends Father{
    int sage;
    protected String sname = new String();
    void getsdetails(){
        Scanner sc = new Scanner(System.in);
        try{
            System.out.println("Enter the name of the son");
            this.sname = sc.nextLine();
            System.out.println("Enter the age of the son");
            this.sage = sc.nextInt();
            if(super.fage<this.sage){
                throw new WrongAgeExcpetion("Enter a valid age!! Son's age cannot
be greater than father's age\n", this.sage);
            }
```

```
                else if(super.fage-this.sage<=21){
                        throw new WrongAgeExcpetion("Enter a valid age!! Age difference of
son and father must be atleast 21 years!! \n", this.sage);

                }
                else{
                        System.out.println("The details of the son are");
                        System.out.println("Name of the son: "+this.sname);
                        System.out.println("Age of the son: "+this.sage);
                }

        }
        catch (WrongAgeExcpetion e){
                System.out.println(e);
        }
    }

}


class Driver{
    public static void main(String args[]){
    System.out.println("Name: Amith R\nUSN: 1BM23CS028\n");
    Son s = new Son();
    if(s.getfdetails()){
            s.getsdetails();
    }
    else{
            System.out.println("Father's details are invalid; so you cannot enter son's details \n");
```

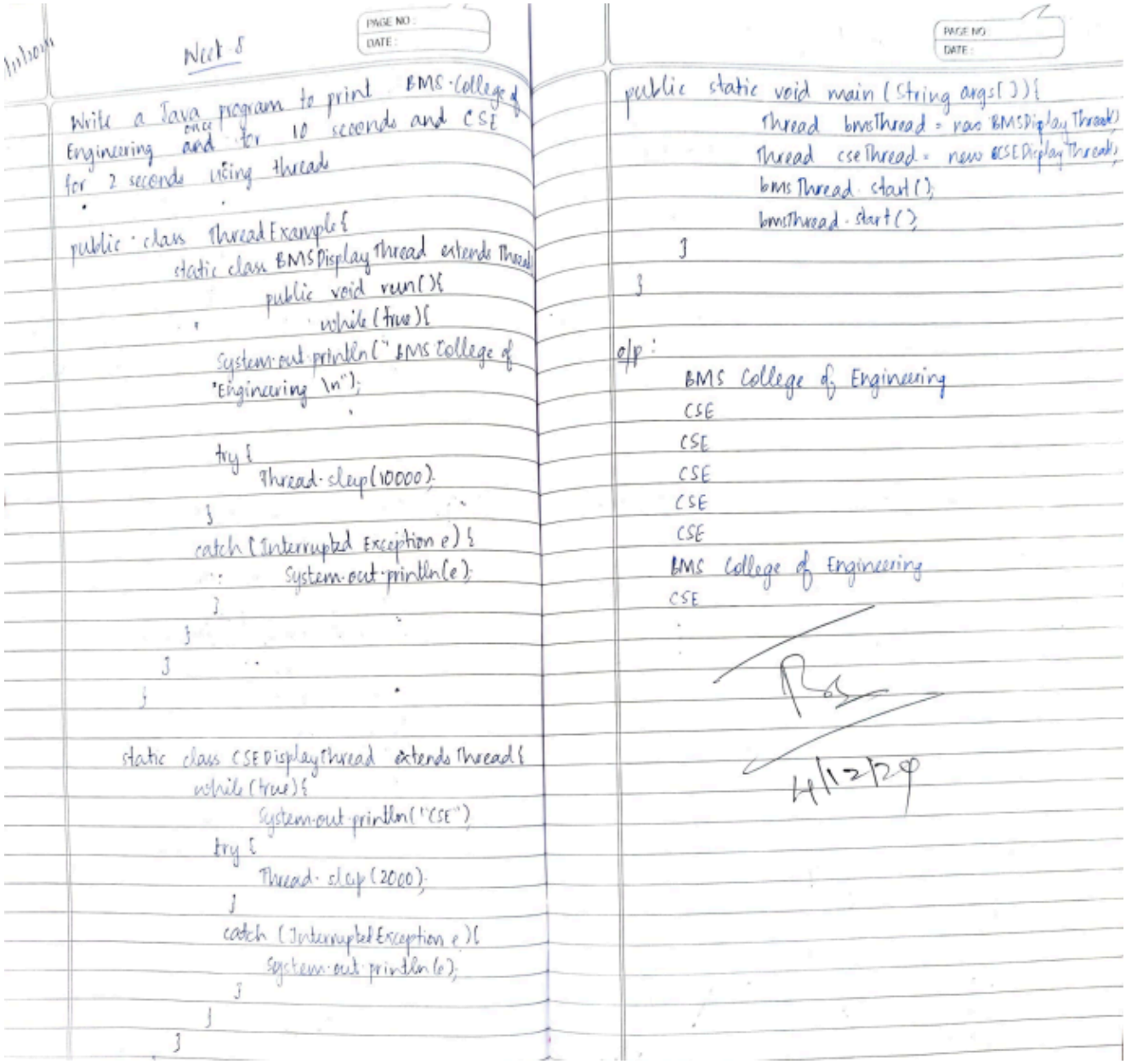
```
D:\23CS028>java Driver
Name: Amith R
USN: 1BM23CS028

Enter the name of the father
ABC
Enter the age of the father
30
The details of the father are
Name of the father: ABC
Age of the father: 30
Enter the name of the son
XYZ
Enter the age of the son
32
Invalid age: 32
Enter a valid age!! Son's age cannot be greater than father's age
```

```
D:\23CS028>java Driver
Name: Amith R
USN: 1BM23CS028

Enter the name of the father
Dhritharashtra
Enter the age of the father
70
The details of the father are
Name of the father: Dhritharashtra
Age of the father: 70
Enter the name of the son
Dhuryodhana
Enter the age of the son
45
The details of the son are
Name of the son: Dhuryodhana
Age of the son: 45

D:\23CS028>java Driver
Name: Amith R
USN: 1BM23CS028

Enter the name of the father
Pandu
Enter the age of the father
67
The details of the father are
Name of the father: Pandu
Age of the father: 67
Enter the name of the son
Karna
Enter the age of the son
49
Invalid age: 49
Enter a valid age!! Age difference of son and father must be atleast 21 years!!
```

Program 8

Multi Threading

Algorithm:



Week 8

Write a Java program to print BMS College of Engineering once and for 10 seconds and CSE for 2 seconds using thread

```
public class ThreadExample{
    static class BMSDisplayThread extends Thread{
        public void run(){
            while(true){
                System.out.println("BMS College of Engineering \n");
                try{
                    Thread.sleep(10000);
                }
                catch(InterruptedException e){
                    System.out.println(e);
                }
            }
        }
    }

    static class CSEDisplayThread extends Thread{
        while(true){
            System.out.println("CSE");
            try{
                Thread.sleep(2000);
            }
            catch(InterruptedException e){
                System.out.println(e);
            }
        }
    }
}
```

```
public static void main(String args[]){
        Thread bmsThread = new BMSDisplayThread();
        Thread cseThread = new CSEDisplayThread();
        bmsThread.start();
        bmsThread.start();
    }
}
```

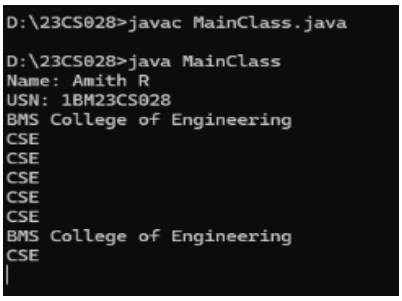
o/p:

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

4/12/24

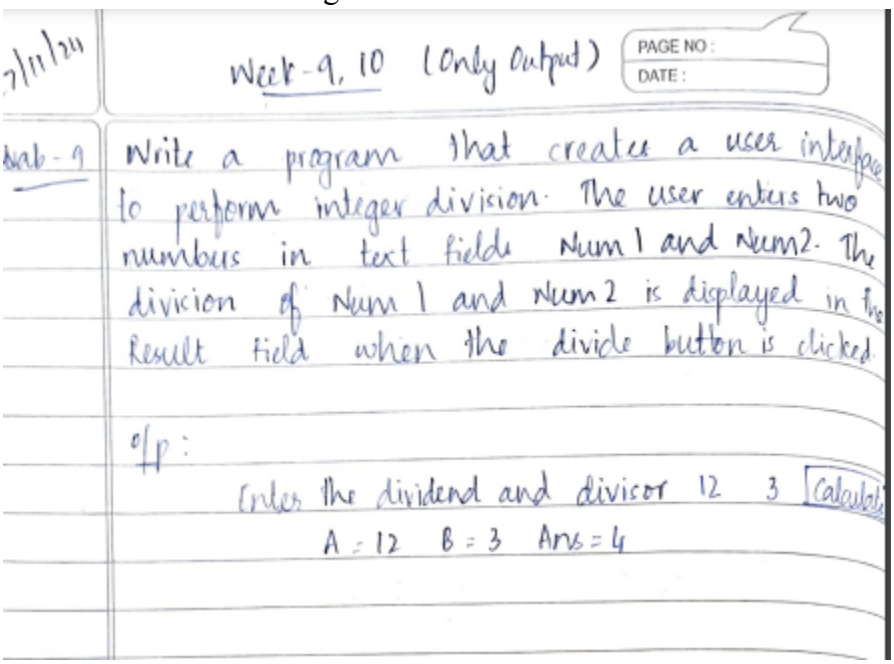
Code:

```
public class MainClass {

    public static class BMSCollegeThread extends Thread {
        @Override
        public void run() {
            while (true) {
                try {
                    System.out.println("BMS College of Engineering");
                    Thread.sleep(10000);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }
        }
    }

    public static class CSEThread extends Thread {
        @Override
        public void run() {
            while (true) {
                try {
                    System.out.println("CSE");
                    Thread.sleep(2000);
                } catch (InterruptedException e) {
                    System.out.println(e.getMessage());
                }
            }
        }
    }

    public static void main(String[] args) {
            System.out.println("Name: Amith R");
            System.out.println("USN: 1BM23CS028");
        BMSCollegeThread bmsThread = new BMSCollegeThread();
        CSEThread cseThread = new CSEThread();
        bmsThread.start();
        cseThread.start();
    }
}
```


D:\23CS028>javac MainClass.java
D:\23CS028>java MainClass
Name: Amith R
USN: 1BM23CS028
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE

Program 9

Division of numbers using GUI:

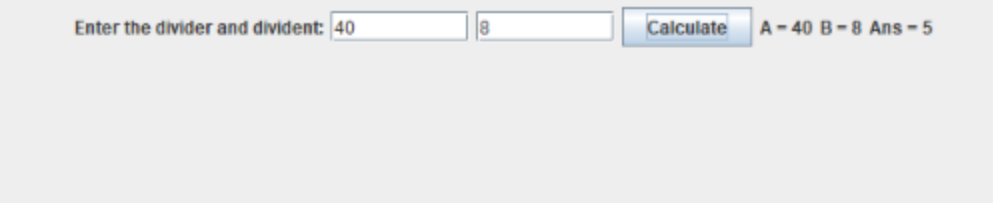


Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
SwingDemo(){
// create jframe container
JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 150);
jfrm.setLayout(new FlowLayout());
// to terminate on close
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
JLabel jlab = new JLabel("Enter the divider and divident:");
// add text field for both numbers
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
// calc button
JButton button = new JButton("Calculate");
// labels
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();

JLabel anslab = new JLabel();
// add in order :)
```

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

err.setText("Enter Only Integers!");
}
catch(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("B should be NON zero!");
}
}
});
// display frame
jfrm.setVisible(true);
}
public static void main(String args[]){
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable(){
public void run(){
new SwingDemo();
```

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

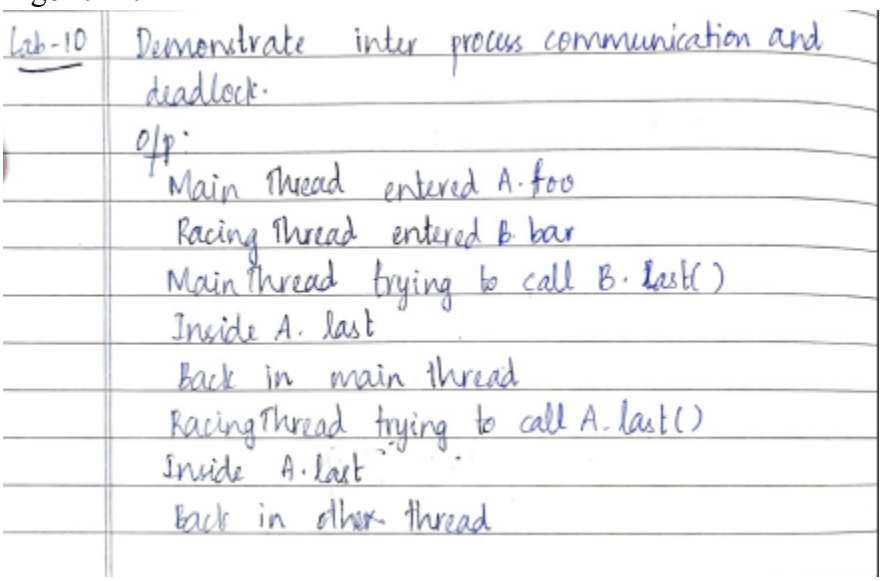

Enter the divider and divident:
40
8
Calculate
A = 40 B = 8 Ans = 5

Program 10

IPC and Deadlock

Deadlock:

Algorithm:

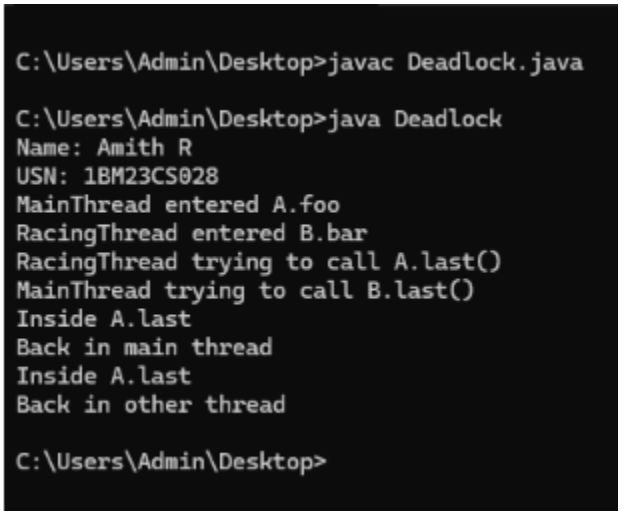


Code:

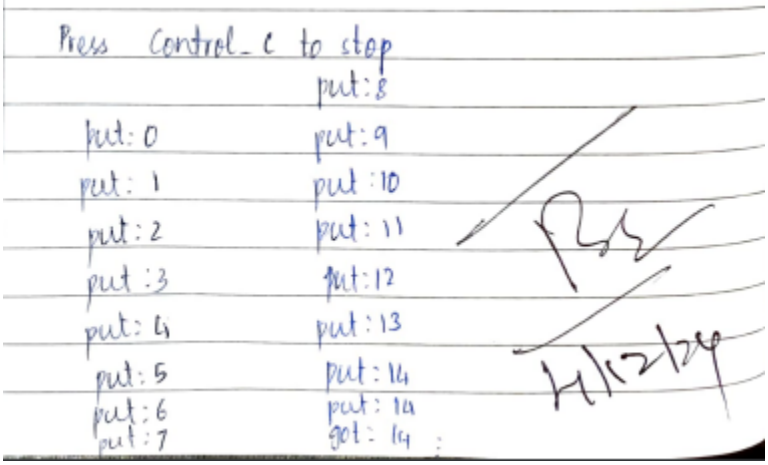
```
class A {

synchronized void foo(B b) {

String name =
Thread.currentThread().getName();
System.out.println("Name: Amith R\nUSN: 1BM23CS028");
System.out.println(name + " entered A.foo");

try {

Thread.sleep(1000);

} catch(Exception e) {

System.out.println("A Interrupted");

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

b.last();

}

void last() {
```

```
System.out.println("Inside A.last");

}

}

class B {

synchronized void bar(A a) {

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

System.out.println(name + " entered B.bar");

try {

Thread.sleep(1000);

} catch(Exception e) {

System.out.println("B Interrupted");

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

a.last();

}

void last() {

System.out.println("Inside A.last");

}

}

class Deadlock implements Runnable
{

A a = new A();

B b = new B();

Deadlock() {
Thread.currentThread().setName("MainThread");
```

```
Thread t = new Thread(this,"RacingThread");
t.start();
a.foo(b); // get lock on a in this thread.
System.out.println("Back in main thread");
}
public void run() {
b.bar(a); // get lock on b in other thread.
System.out.println("Back in other thread");
}
public static void main(String args[]) {
new Deadlock();
}
}
```



IPC:
Algorithm:



Code:

```
class Q {

int n;

boolean valueSet = false;

synchronized int get() {

while(!valueSet)

try {

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

wait();

} catch(InterruptedException e) {

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

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

valueSet = false;

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

notify();
```

```
return n;
}
synchronized void put(int n) {
while(valueSet)
try {
System.out.println("\nProducer waiting\n");
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}
class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
```

```
int i = 0;

while(i<15) {

q.put(i++);

}

}

}
class Consumer implements Runnable {

Q q;

Consumer(Q q) {

this.q = q;

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

}

public void run() {

int i=0;

while(i<15) {

int r=q.get();

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

i++;

}

}

}

class PCFixed{
public static void main(String args[]) {
System.out.println("Name: Amith");
System.out.println("USN:1BM23CS028");
Q q = new Q();
```

```
new Producer(q);

new Consumer(q);

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

}

}
```

```
C:\Users\Admin\Desktop>javac PCFixed.java

C:\Users\Admin\Desktop>java PCFixed
Name: Amith
USN:1BM23CS028
Press Control-C to stop.
Put: 0

Intimate Consumer


Producer waiting

Got: 0

Intimate Producer

Put: 1

Intimate Consumer


Producer waiting

consumed:0
Got: 1

Intimate Producer

consumed:1
Put: 2

Intimate Consumer


Producer waiting

Got: 2

Intimate Producer

consumed:2
Put: 3

Intimate Consumer


Producer waiting

Got: 3
```