VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum -590014, Karnataka.

LAB REPORT

on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Dyuthi (1BM23CS097)

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 **Dyuthi (1BM23CS097),** 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.

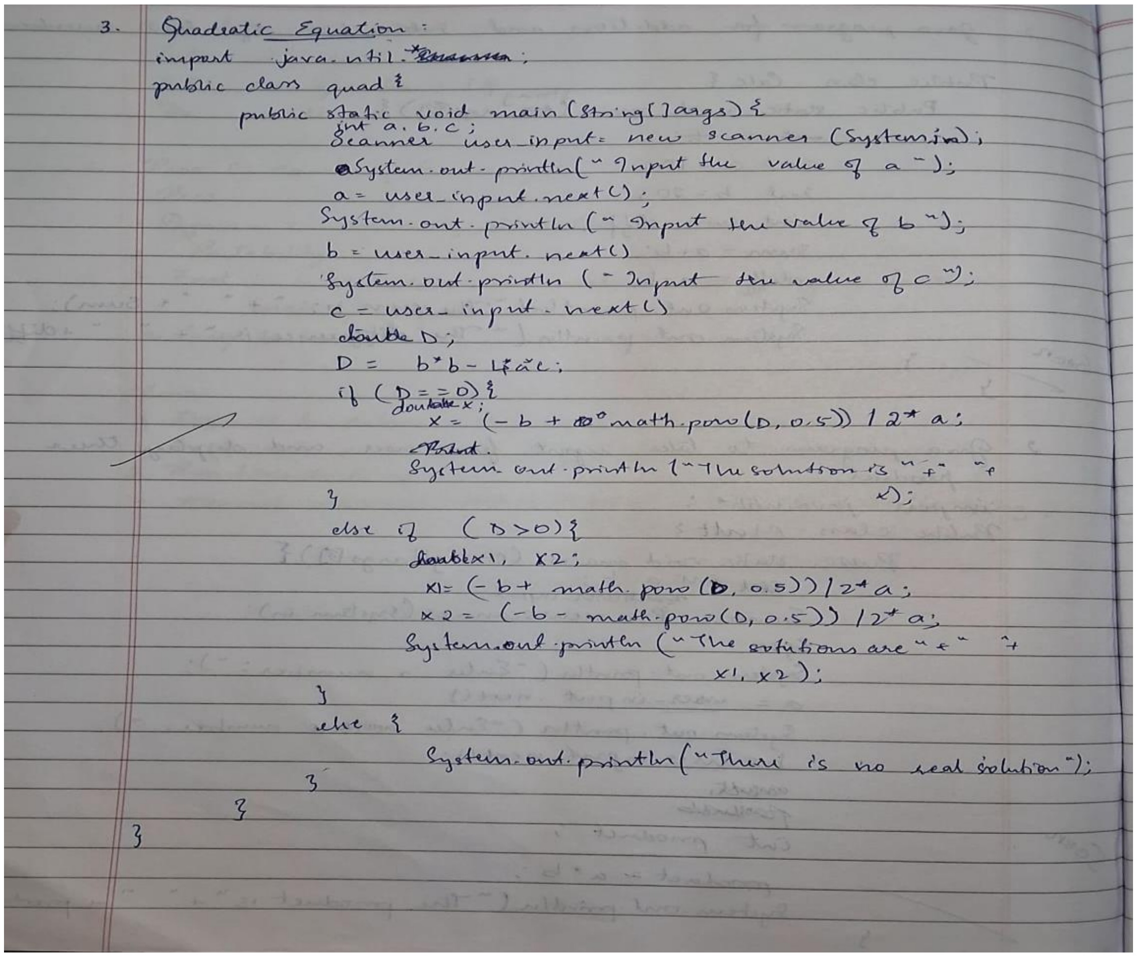
Lab faculty Swathi Sridharan Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

GitHub Link:
https://github.com/Dyuthiprasad/java-lab/tree/main

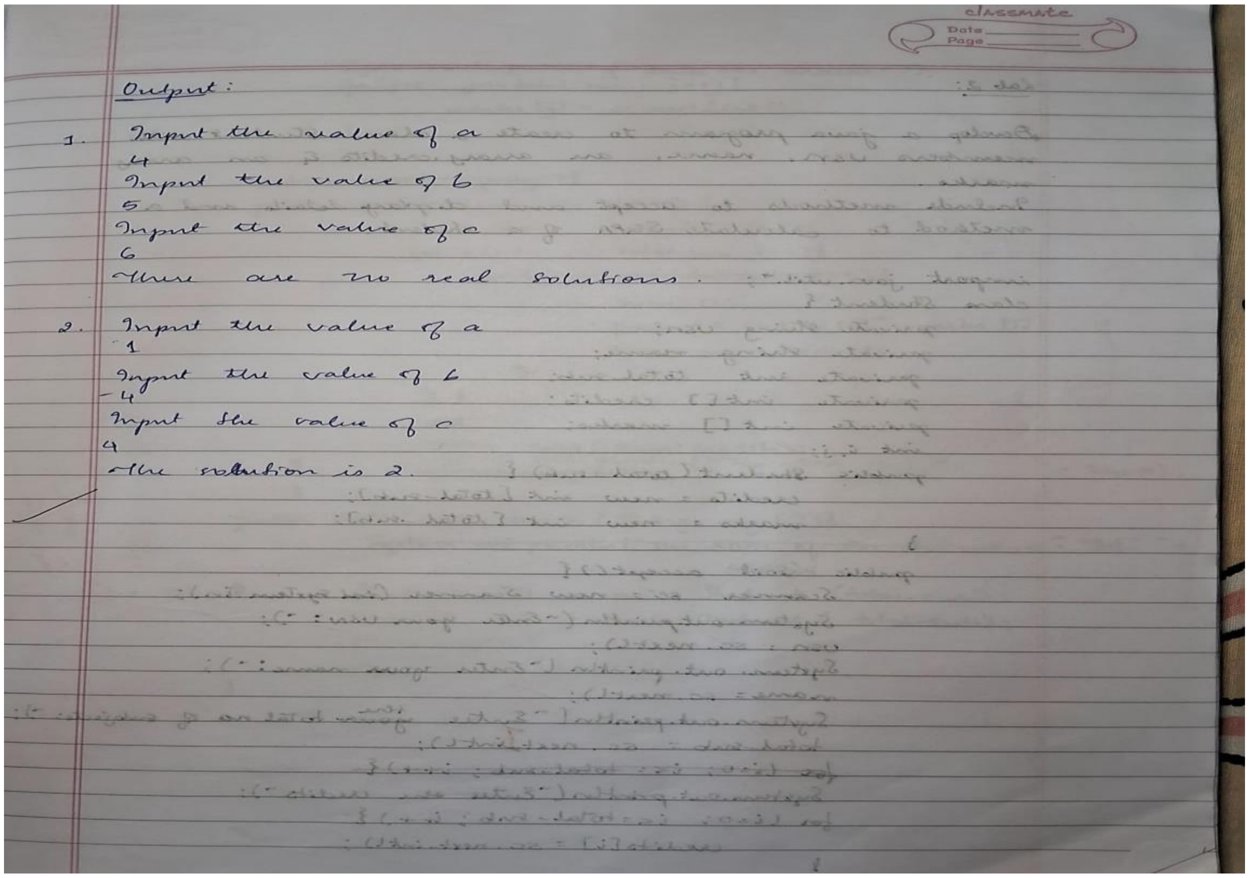
<u>Program 1</u>
Implement Quadratic Equation

Algorithm:



3. Quadratic Equation :

```
import   java.util.*[illegible];
public class  quad {
        public static void main(String[]args){
                int a.b.c;
                Scanner  user_input= new  scanner (System.in);
                System.out.println(" Input the  value of a ");
                a= user_input.next();
               System.out.println (" Input the value of b ");
                b = user_input. next()
               System.out.println (" Input the value of c ");
                c = user_input. next()
                double D;
                D =    b*b - 4*a*c;
                if (D==0){
                double x;
                      x = (-b + math.pow(D,0.5)) / 2* a;
                      ~~Point~~.
                      System.out.println ("The solution is " + "  "+
                                                              x);
                }
                else if    (D>0){
                      double x1,  x2;
                       x1= (-b+ math.pow(D, 0.5))/2* a;
                       x2 =   (-b - math.pow(D,0.5)) /2* a;
                      System.out.println ("The solutions are " + "  "+
                                                   x1, x2);
               }
               else {
                       System.out.println("There is no real solution");
               }
        }
}
```



Output:

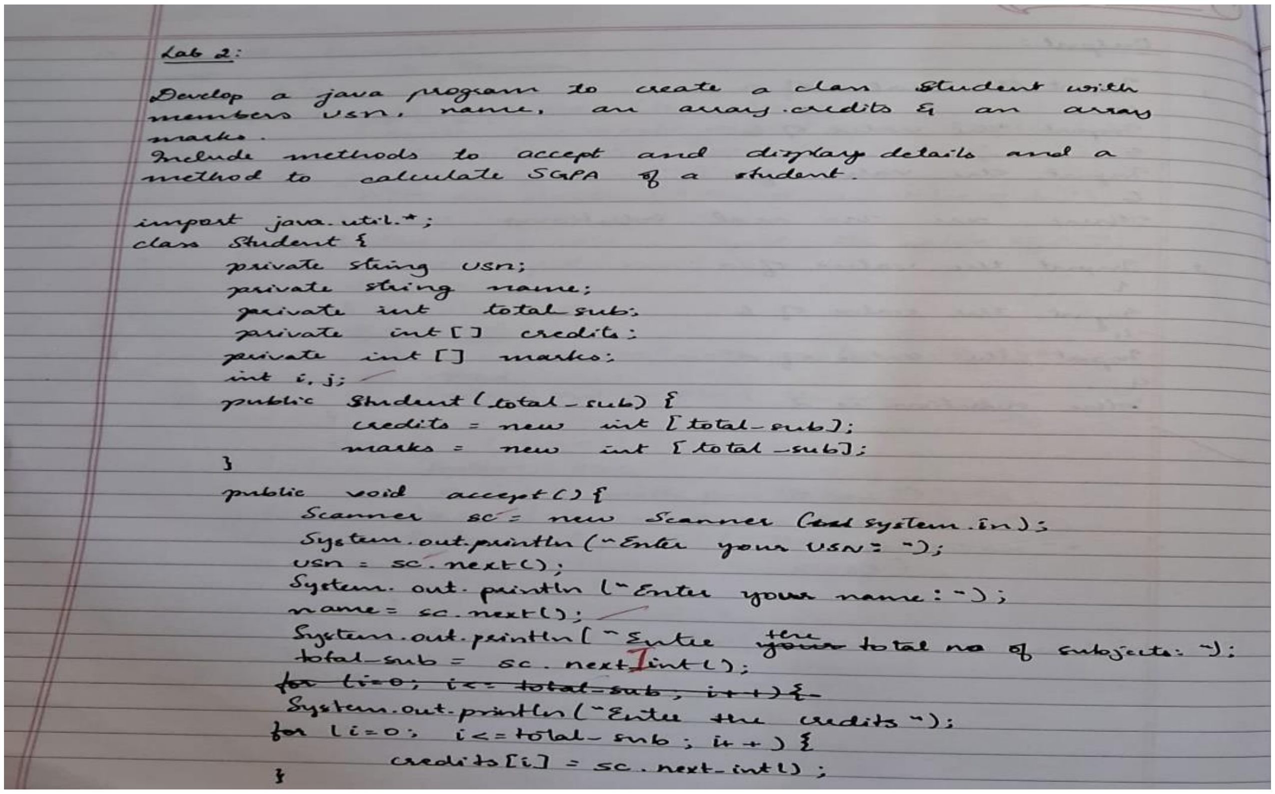
```
1. Input the value of a
   4
   Input the value of b
   5
   Input the value of c
   6
   There are no real solutions.

2. Input the value of a
   1
   Input the value of b
   -4
   Input the value of c
   4
   The solution is 2.
```

Code:

```java
import java.util.Scanner;

public class quad {
    public static void main(String[] args) {
        int a, b, c;
        Scanner sc = new Scanner(System.in);
        System.out.println("Input the value of a:");
        a = sc.nextInt();
        System.out.println("Input the value of b:");
        b = sc.nextInt();
        System.out.println("Input the value of c:");
        c = sc.nextInt();

        double D;
        D = b * b - 4 * a * c;

        if (D >= 0) {
            double x;
            x = (-b + Math.pow(D, 0.5)) / 2 * a;
            System.out.println("The solution is " + x);
        }
        else if (D > 0) {
            double x1, x2;
            x1 = (-b + Math.pow(D, 0.5)) / (2 * a);
            x2 = (-b - Math.pow(D, 0.5)) / (2 * a);
            System.out.println("The solutions are " + x1 + ", " + x2);
        }
        else {
            System.out.println("There is no real solution");
        }
    }
}
```

PROGRAM 2:

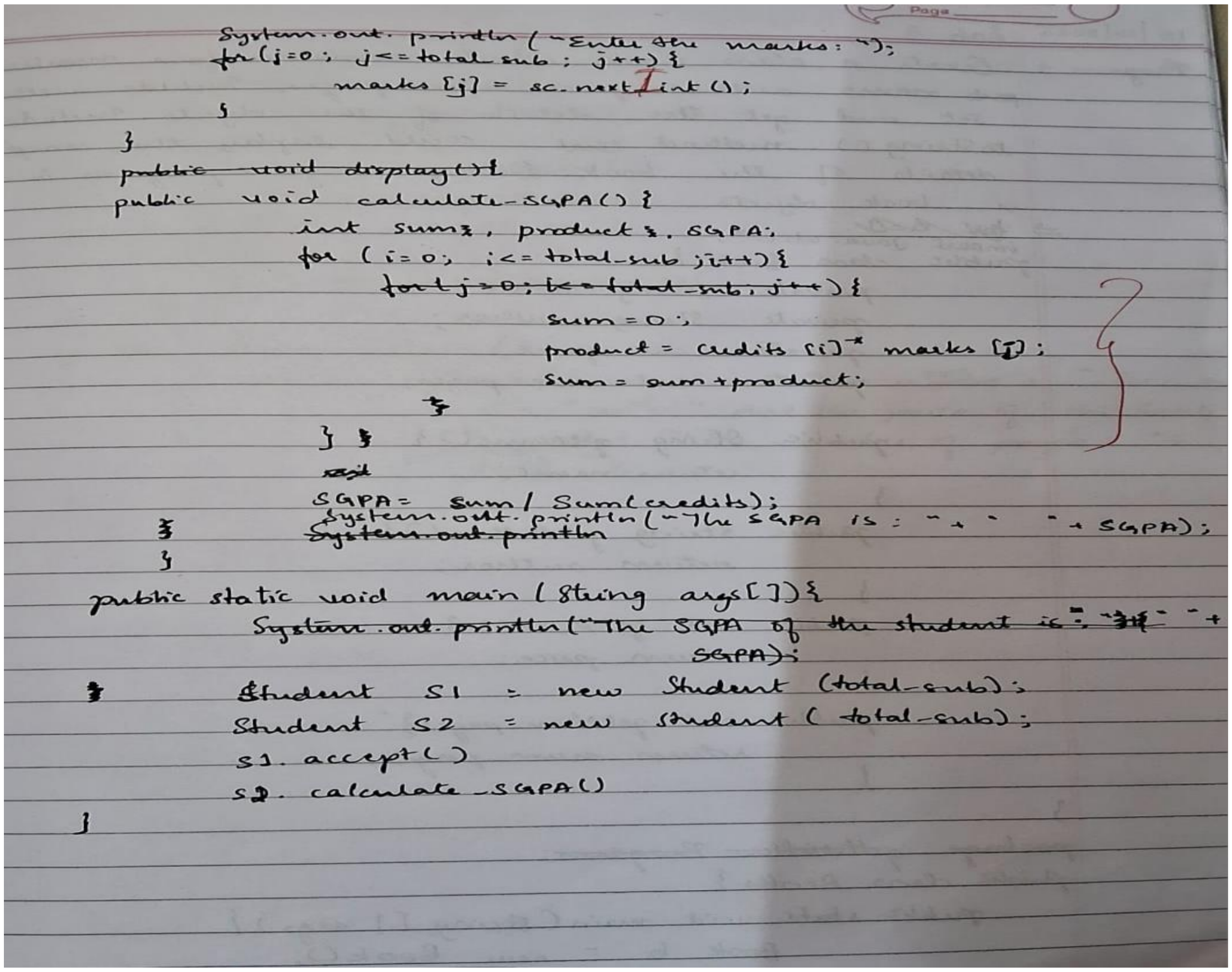
SGPA calculator



Lab 2:

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

```
import java.util.*;
class Student {
        private string usn;
        private string name;
        private int total sub;
        private int [] credits;
        private int [] marks;
        int i, j;
        public Student (total_sub) {
                credits = new int [total_sub];
                marks = new int [total_sub];
        }
        public void accept() {
            Scanner sc = new Scanner (~~[illegible]~~ system.in);
            System.out.println ("Enter your USN: ");
            usn = sc.next();
            System.out.println ("Enter your name: ");
            name = sc.next();
            System.out.println ("Enter the ~~your~~ total no of subjects: ");
            total_sub = sc.nextInt();
           ~~for (i=0; i<= total_sub; i++){~~
            System.out.println ("Enter the credits ");
           for (i=0; i<=total_sub; i++) {
                credits[i] = sc.next_int();
           }
```



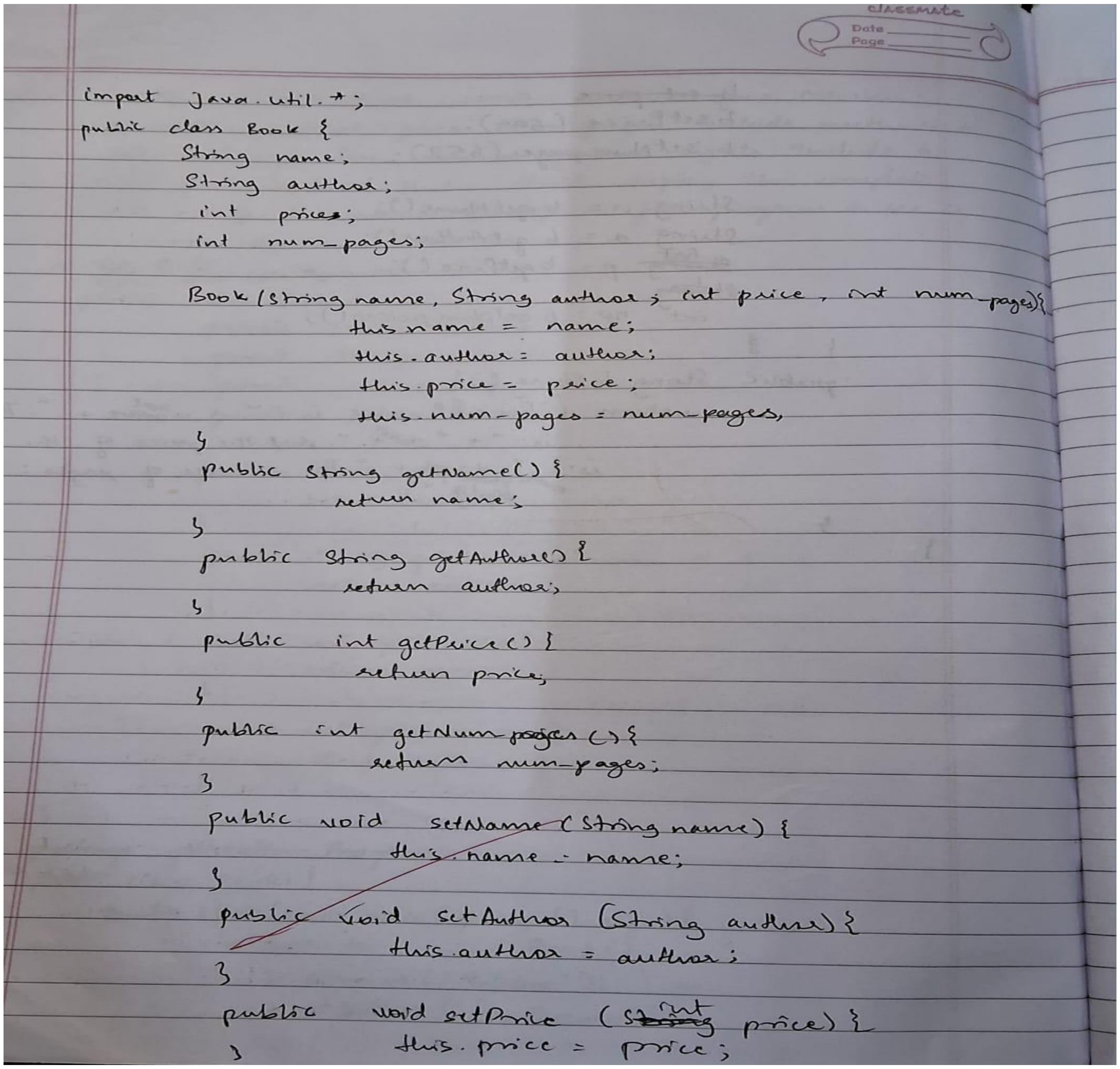
```
        System.out.println("Enter the marks: ");
        for(j=0; j<=total sub ; j++) {
                marks [j] = sc.nextInt();
        }
    }
    ~~public void display(){~~
    public void calculate_SGPA() {
            int sum, product, SGPA;
            for ( i=0;  i<= total_sub ;i++){
                 ~~for(j=0; i<=total_sub; j++){~~
                          sum =0;
                          product = credits [i]* marks [j];
                          sum = sum+product;
                    }
            } }
            ~~[illegible]~~
            SGPA= sum/ Sum(credits);
            System.out.println("The SGPA is : "+ "   "+ SGPA);
    }       ~~System.out.println~~
    }
public static void main ( String args[]){
        ~~System.out.println("The SGPA of the student is : " " +~~
                                        ~~SGPA);~~
}       Student   S1   = new  Student (total_sub);
        Student   S2   = new  Student ( total_sub);
        s1. accept( )
        s2. calculate_SGPA()
}
```

CODE:

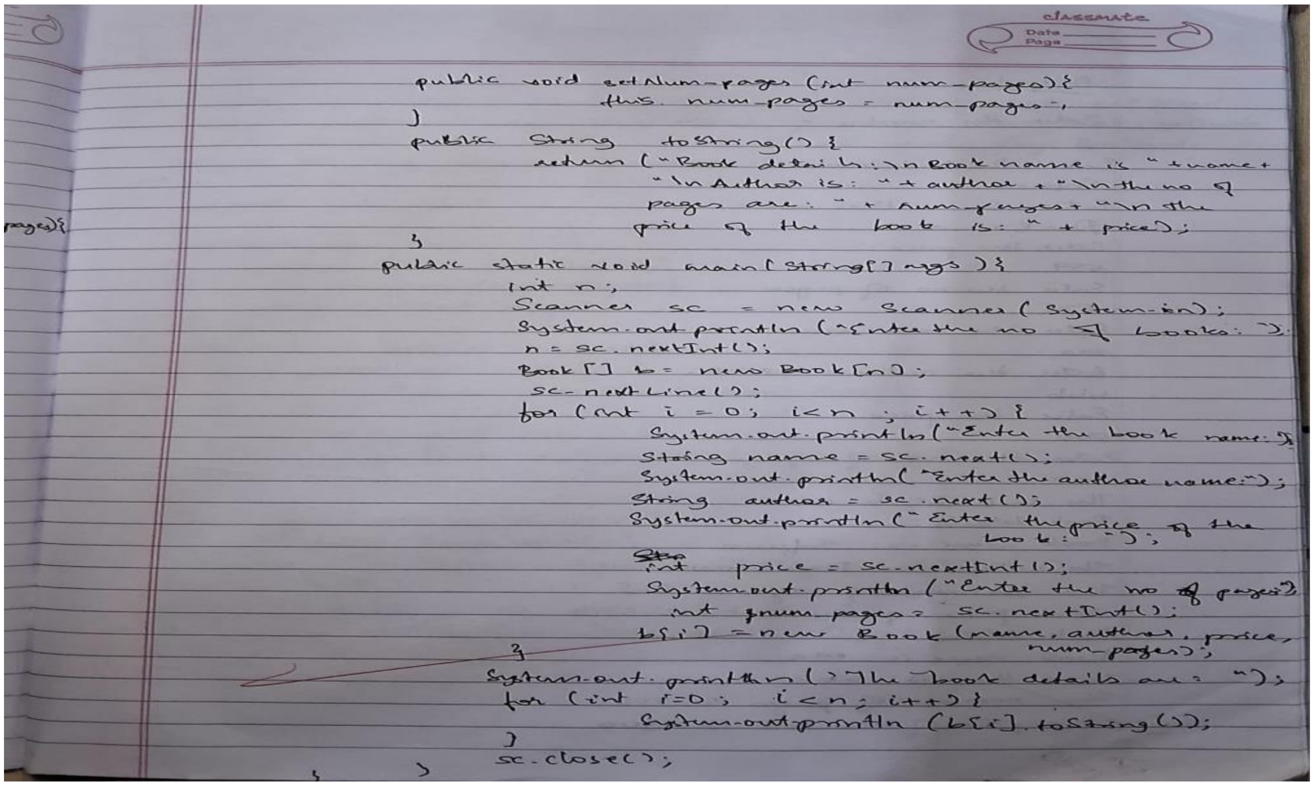
```
import java.util.Scanner;

class Student {
    // Class members
    private String usn;
    private String name;
    private int[] credits;
    private int[] marks;
    private int numSubjects;

    // Method to accept student details
    public void acceptDetails() {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter USN: ");
        usn = sc.nextLine();

        System.out.print("Enter Name: ");
        name = sc.nextLine();

        System.out.print("Enter the number of subjects: ");
        numSubjects = sc.nextInt();

        credits = new int[numSubjects];
        marks = new int[numSubjects];

        System.out.println("Enter the credits and marks for each subject:");
        for (int i = 0; i < numSubjects; i++) {
            System.out.print("Subject " + (i + 1) + " - Credits: ");
            credits[i] = sc.nextInt();

            System.out.print("Subject " + (i + 1) + " - Marks: ");
            marks[i] = sc.nextInt();
        }
    }
    public void displayDetails() {
        System.out.println("\n--- Student Details ---");
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Subject-wise Credits and Marks:");
        for (int i = 0; i < numSubjects; i++) {
            System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ", Marks: " + marks[i]);
        }
    }
    public double calculateSGPA() {
        int totalCredits = 0;
        double totalGradePoints = 0.0;

        for (int i = 0; i < numSubjects; i++) {
            int gradePoint = calculateGradePoint(marks[i]);
            totalGradePoints += gradePoint * credits[i];
            totalCredits += credits[i];
        }
        if (totalCredits == 0) {
            return 0.0; // Avoid division by zero
        }
        return totalGradePoints / totalCredits;
    }
    private int calculateGradePoint(int marks) {
        if (marks >= 90) return 10;
        else if (marks >= 80) return 9;
        else if (marks >= 70) return 8;
        else if (marks >= 60) return 7;
        else if (marks >= 50) return 6;
        else if (marks >= 40) return 5;
        else return 0;
    }
    public static void main(String[] args) {
        Student student = new Student();
        student.acceptDetails();
        student.displayDetails();
        double sgpa = student.calculateSGPA();
        System.out.printf("\nSGPA: %.2f\n", sgpa);
    }
}
```

PROGRAM 3:

Implementing toString() method



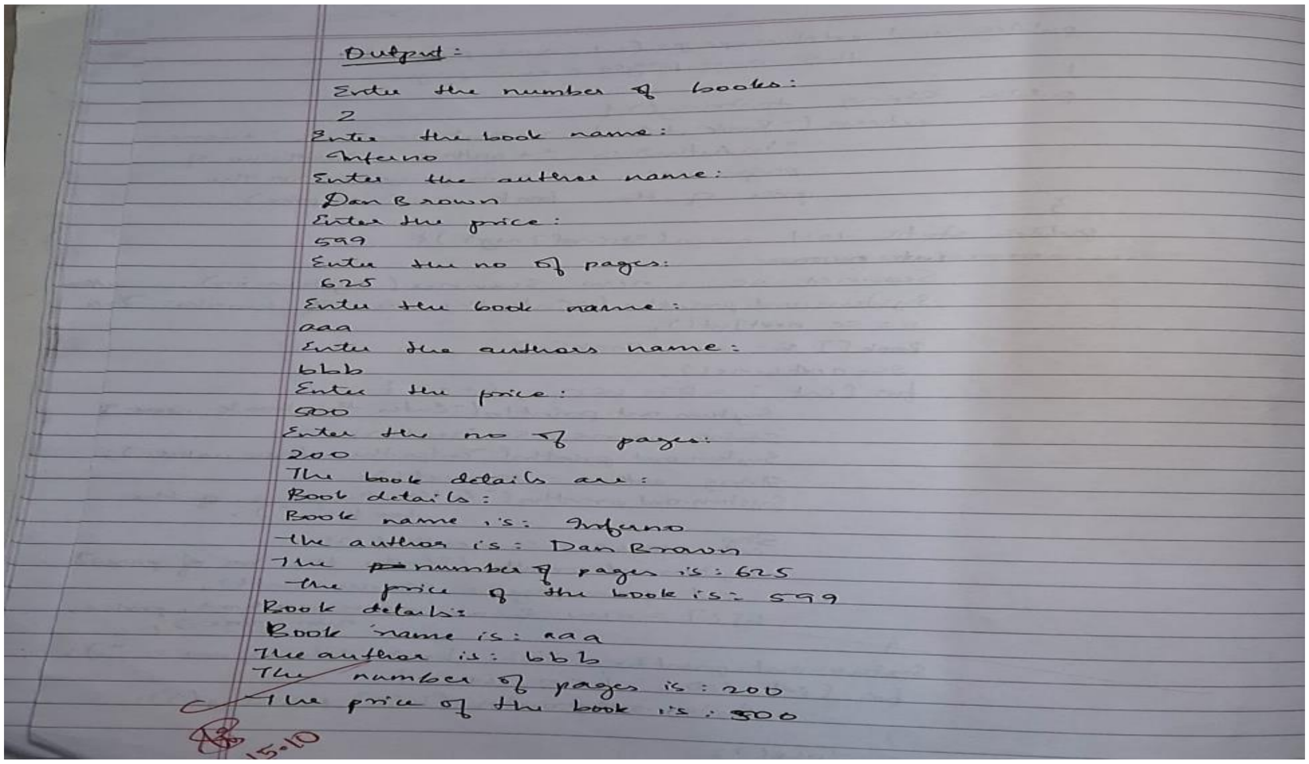
```
import java.util.*;
public class Book {
        String name;
        String author;
         int price;
         int num_pages;

        Book(String name, String author; int price, int num_pages){
                    this.name = name;
                    this.author = author;
                    this.price = price;
                    this.num_pages = num_pages,
        }
        public String getName() {
                   return name;
        }
        public String getAuthor() {
                   return author;
        }
        public int getPrice() {
                   return price;
        }
        public int getNum_pages() {
                   return num_pages;
        }
        public void setName (String name) {
                     this.name = name;
        }
        public void setAuthor (String author){
                     this.author = author;
        }
        public void setPrice (~~String~~ int price) {
        }            this.price = price;
```



classmate
Date
Page

pages){

```
        public void setNum-pages (int num_pages){
                this. num_pages = num_pages;
        }
        public String toString() {
                return ("Book details:\n Book name is "+name+
                        "\n Author is: "+ author + "\n the no of
                        pages are: " + num_pages+ "\n the
                        price of the book is: " + price);
        }
    public static void main(String[] args ){
            int n;
            Scanner sc = new Scanner ( System.in);
            System.out.println ("Enter the no of books: ");
            n = sc.nextInt();
            Book [] b = new Book[n];
            sc.nextLine();
            for (int i = 0; i<n ; i++) {
                    System.out.println("Enter the book name:");
                    String name = sc.next();
                    System.out.println("Enter the author name:");
                    String author = sc.next();
                    System.out.println(" Enter the price of the
                                                book: ");
                    ~~Str~~
                    int price = sc.nextInt();
                    System.out.println ("Enter the no of pages");
                    int num_pages = sc.nextInt();
                    b[i] = new Book (name, author, price,
                                             num_pages);
            }
          System.out.println ("The book details are: ");
          for (int i=0 ; i<n; i++) {
                    System.out.println (b[i].toString());
          }
          sc.close();
    }           }
```



Output:

```
Enter the number of books:
2
Enter the book name:
Inferno
Enter the author name:
Dan Brown
Enter the price:
599
Enter the no of pages:
625
Enter the book name:
aaa
Enter the authors name:
bbb
Enter the price:
500
Enter the no of pages:
200
The book details are:
Book details:
Book name is: Inferno
the author is: Dan Brown
The ~~p~~ number of pages is: 625
the price of the book is: 599
Book details:
Book name is: aaa
The author is: bbb
The number of pages is: 200
The price of the book is: 500
```

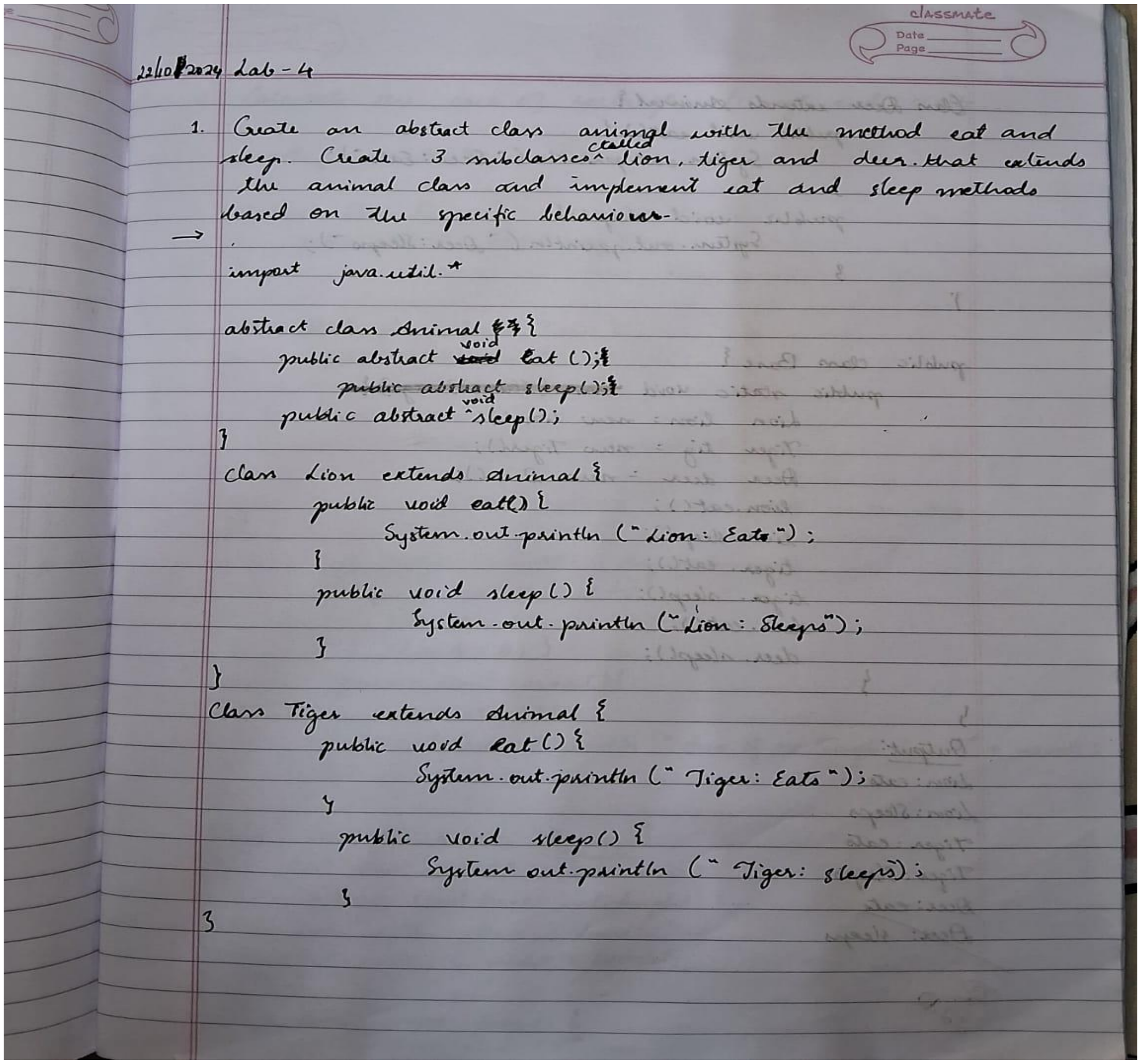
[illegible] 15.10

CODE:

```
import java.util.Scanner;

class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;

    public Book(String name, String author, double price, int numPages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public void setNumPages(int numPages) {
        this.numPages = numPages;
    }

    public String getName() {
        return name;
    }

    public String getAuthor() {
        return author;
    }

    public double getPrice() {
        return price;
    }

    public int getNumPages() {
        return numPages;
    }
    public String toString() {
        return "Book Name: " + name + "\nAuthor: " + author + "\nPrice: " + price + "\nNumber of
Pages: " + numPages;
    }
}

public class BookDetails {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();
        scanner.nextLine();

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for Book " + (i+1));

            System.out.print("Enter book name: ");
            String name = scanner.nextLine();

            System.out.print("Enter author name: ");
            String author = scanner.nextLine();

            System.out.print("Enter price: ");
            double price = scanner.nextDouble();

            System.out.print("Enter number of pages: ");
            int numPages = scanner.nextInt();
            scanner.nextLine();
            books[i] = new Book(name, author, price, numPages);
        }

        System.out.println("\nBook Details:");
        for (int i = 0; i < n; i++) {
            System.out.println("\nDetails of Book " + (i+1) + ":");
            System.out.println(books[i].toString());
        }

        scanner.close();
    }
}
```

PROGRAM 4:

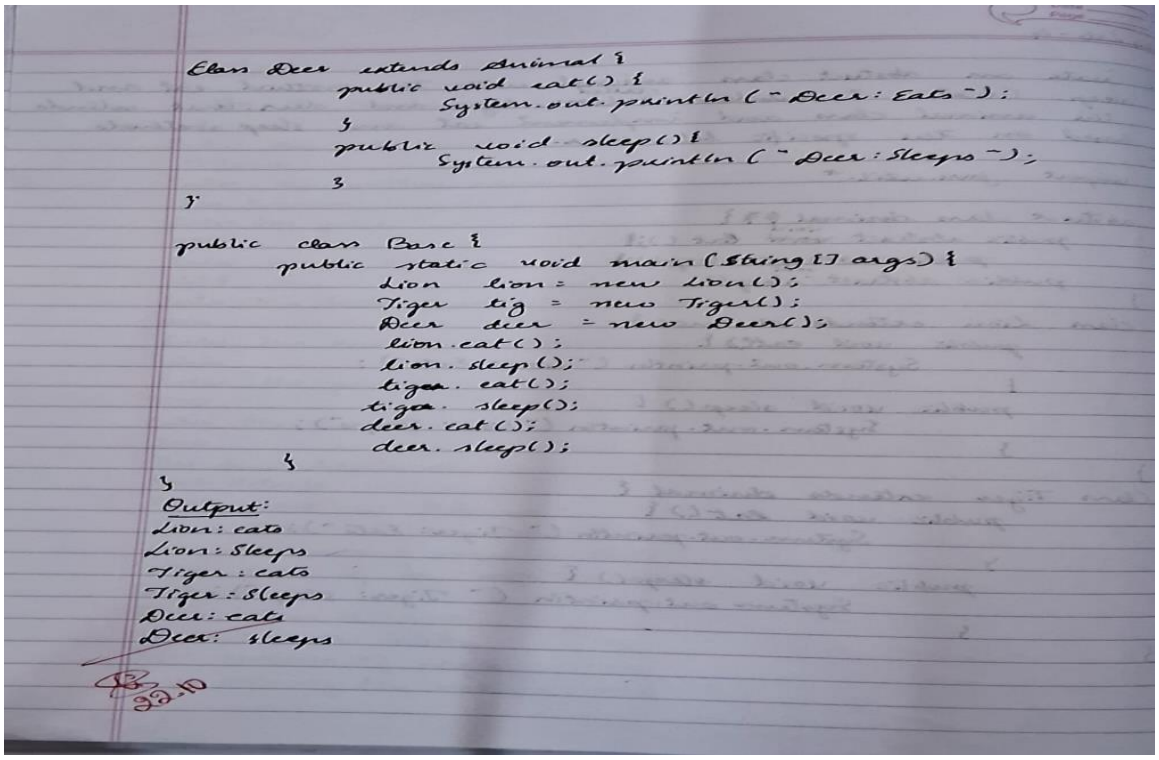
Abstract classes (Animal program):



22/10/2024 Lab - 4

1. Create an abstract class animal with the method eat and sleep. Create 3 subclasses called lion, tiger and deer. that extends the animal class and implement eat and sleep methods based on the specific behaviou~~rs~~-

```
import java.util.*

abstract class Animal {
    public abstract void eat ();
        ~~public abstract sleep();~~
    public abstract void sleep();
}
Class Lion extends Animal {
        public void eat() {
                System.out.println ("Lion: Eats");
        }
        public void sleep() {
                System.out.println ("Lion: Sleeps");
        }
}
Class Tiger extends Animal {
        public void Eat() {
                System.out.println (" Tiger: Eats");
        }
        public void sleep() {
                System.out.println (" Tiger: sleeps);
        }
}
```



```
class Deer extends Animal {
        public void eat() {
                System.out.println ("Deer: Eats");
        }
        public void sleep() {
                System.out.println ("Deer: Sleeps");
        }
};

public class Base {
        public static void main(String[] args) {
                Lion lion = new Lion();
                Tiger tig = new Tiger();
                Deer deer = new Deer();
                lion.eat();
                lion. sleep();
                tiger. eat();
                tiger. sleep();
                deer. eat();
                deer. sleep();
        }
}
```

Output:

```
Lion: eats
Lion: Sleeps
Tiger : eats
Tiger : Sleeps
Deer: eats
Deer: sleeps
```

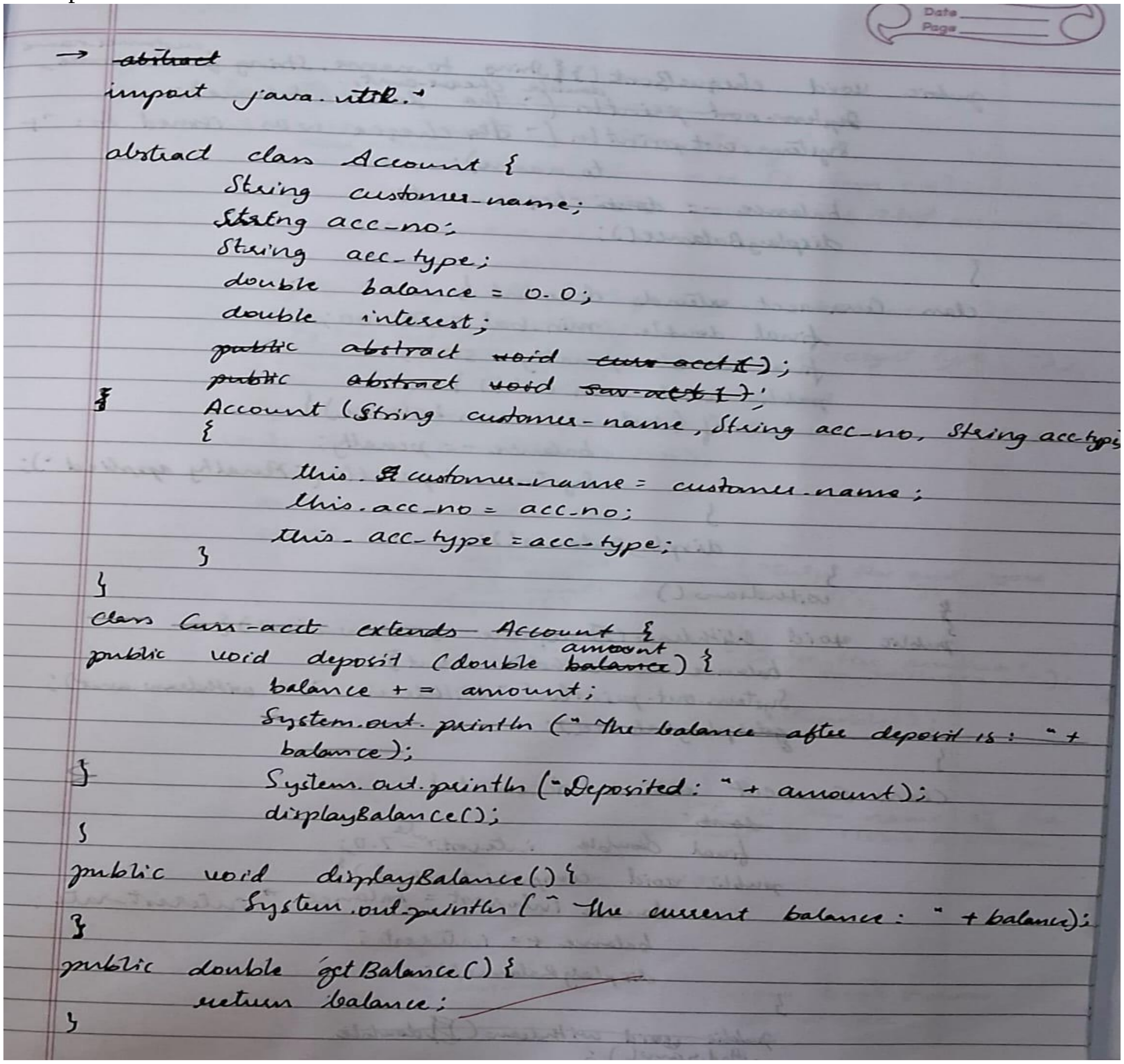
[illegible] 22.10

CODE:

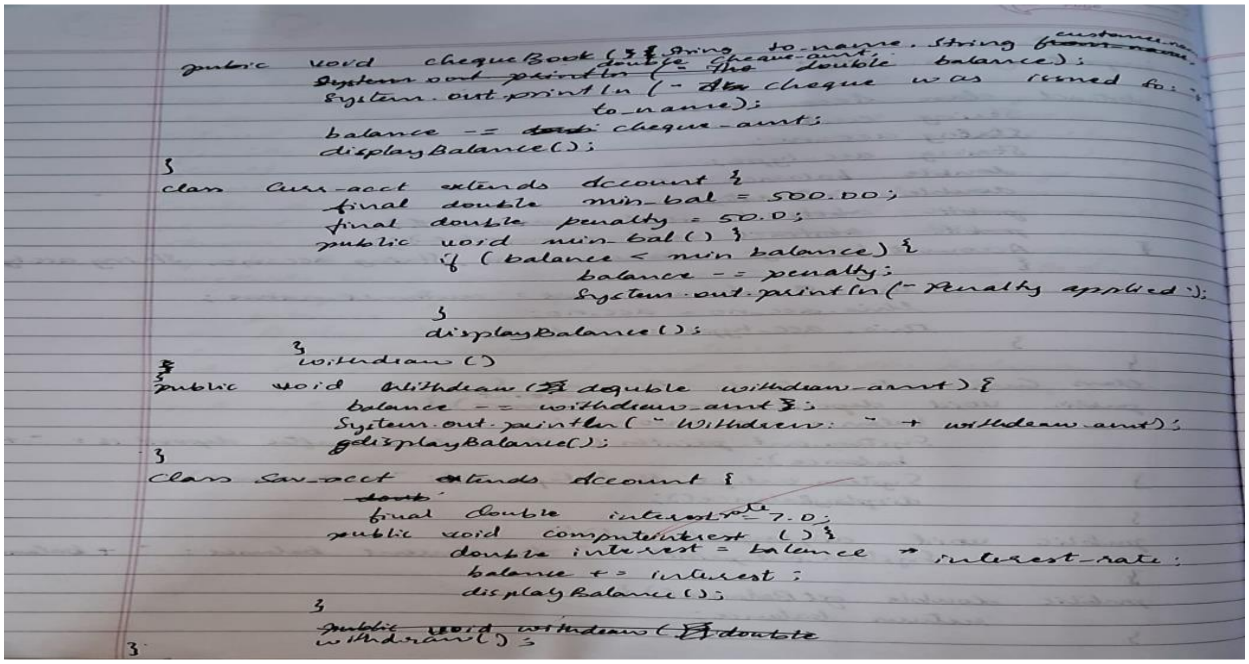
```
import java.util.*;

abstract class Animal {
    public abstract void eat();
    public abstract void sleep();
}

class Lion extends Animal {
    public void eat() {
        System.out.println("Lion: Eats");
    }

    public void sleep() {
        System.out.println("Lion: Sleeps");
    }
}

class Tiger extends Animal {
    public void eat() {
        System.out.println("Tiger: Eats");
    }
    public void sleep() {
        System.out.println("Tiger: Sleeps");
    }
}
Class Deer extends Animal {
    public void eat() {
        System.out.println("Deer: Eats");
    }

    public void sleep() {
        System.out.println("Deer: Sleeps");
    }
}

public class Base {
    public static void main(String[] args) {
        Lion lion = new Lion();
        Tiger tig = new Tiger();
        Deer deer = new Deer();

        lion.eat();
        lion.sleep();
        tig.eat();
        tig.sleep();
        deer.eat();
        deer.sleep();
    }
}
```

PROGRAM 5:

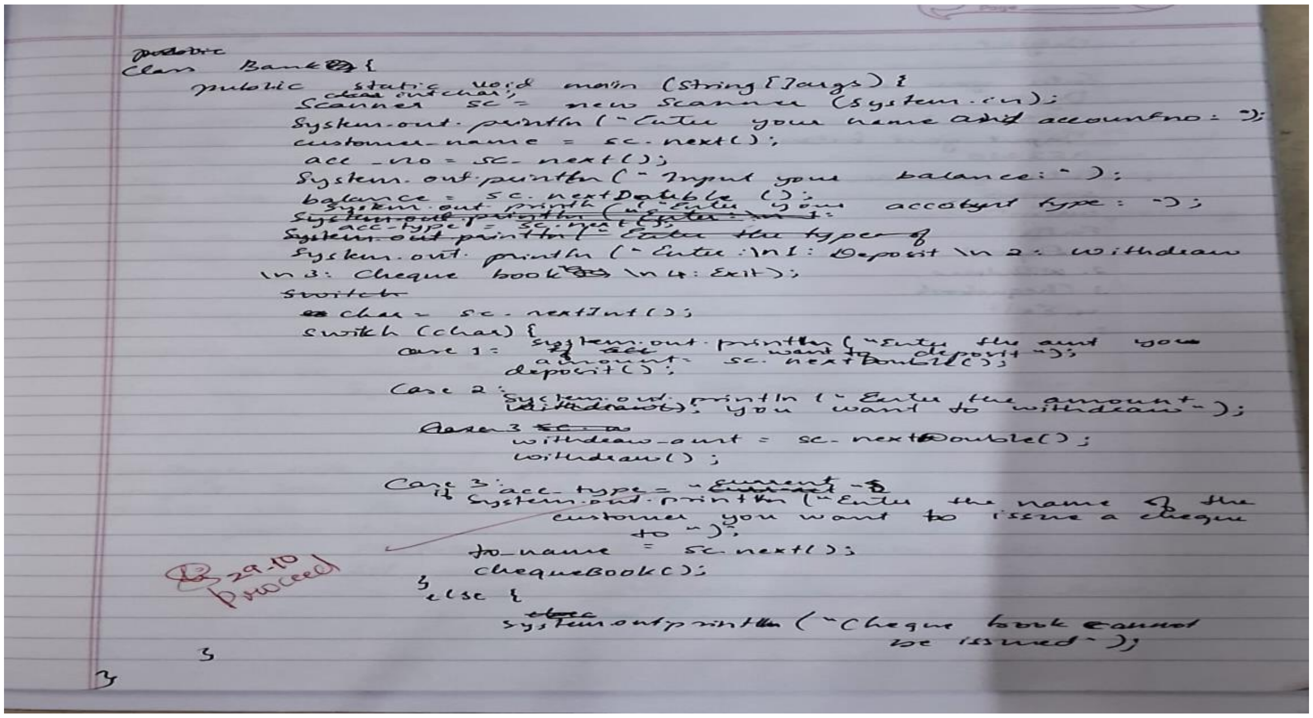
Bank problem:



```
→ ~~abstract~~
import java.util.*

abstract class Account {
        String customer_name;
        String acc_no;
        String acc_type;
        double balance = 0.0;
        double interest;
        ~~public abstract void curr acct()~~;
        ~~public abstract void sav-acct()~~;
~~}~~       Account (String customer_name, String acc_no, String acc_type
        {
            this.customer_name = customer_name;
            this.acc_no = acc_no;
            this.acc_type = acc_type;
        }
}
class Curr-acct extends Account {
public void deposit (double ~~balance~~ amount) {
            balance += amount;
            ~~System.out.println ("The balance after deposit is: " +~~
            ~~balance);~~
}           System.out.println ("Deposited: " + amount);
            displayBalance();
}
public void displayBalance() {
        System.out.println (" The current balance: " + balance);
}
public double getBalance() {
        return balance;
}
```

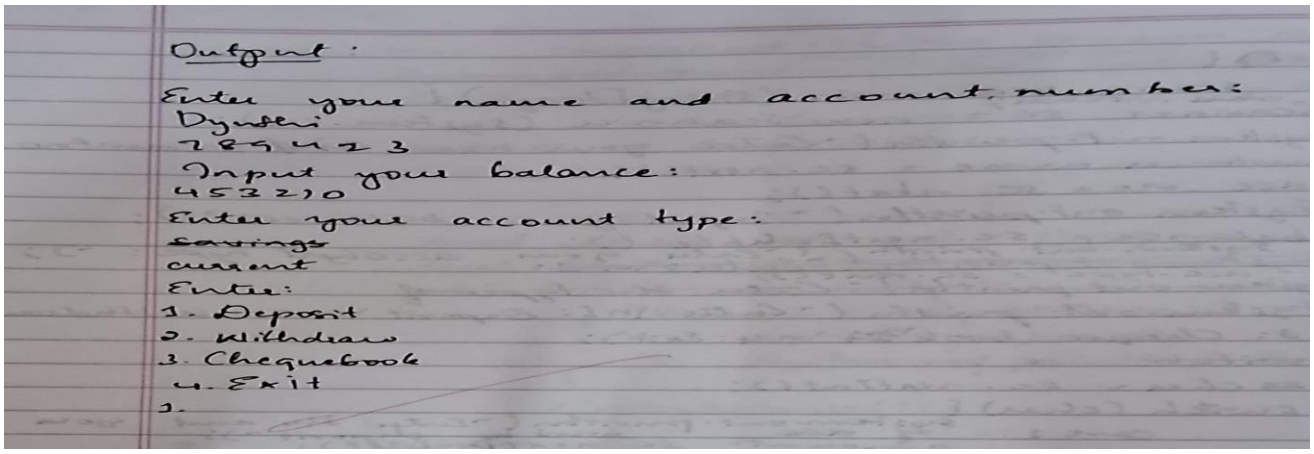


```
public   void   chequeBook (~~{~~ String   to_name , String  ~~from_name~~ customer_name,
                            double cheque_amt,
       ~~System.out.println ("the   double   balance);~~
       System.out.println ("~~the~~ Cheque   was   issued to: " +
                                to_name);
       balance -= ~~doub~~ cheque_amt;
       displayBalance();
}
class   Curr_acct   extends   Account {
        final   double   min_bal = 500.00;
        final   double   penalty = 50.0;
        public   void   min_bal () {
                if ( balance < min balance) {
                        balance -= penalty;
                        System.out.println("Penalty applied");
                }
                displayBalance();
        }
        withdraw ()
}
public   void   Withdraw (~~{~~ double   withdraw_amt) {
        balance  -=  withdraw_amt ~~}~~;
        System.out.println( "Withdrew: " + withdraw_amt);
        ~~g~~displayBalance();
}
Class   Sav_acct   extends   Account {
        ~~doub~~
        final   double   interest_rate = 7.0;
        public   void   computeInterest  () {
                double  interest = balance  *  interest_rate;
                balance += interest;
                displayBalance();
        }
        ~~public   void   withdraw (~~ ~~{~~ ~~double~~
        withdraw();
}
```



```
~~public~~
Class   Bank ~~{}~~ {
    public   static   void   main (String[]args) {
                 class int char;
             Scanner   sc =  new Scanner  (System.in);
             System.out.println ("Enter   your   name and accountno: ");
             customer_name  =  sc.next();
             acc_no = sc.next();
             System.out.println ("Input your   balance: " );
             balance =  sc.nextDouble ();
             ~~System.out.println (" Enter :\n 1:~~
                 System.out.println ("Enter your   account type: ");
                 acc_type = sc.next();
             ~~System.out.println (" Enter  the type of~~
             System.out. println (" Enter :\n 1: Deposit \n 2: Withdraw
         \n 3: Cheque   book \n 4: Exit");
             ~~Switch~~
             char =  sc.nextInt ();
             switch (char) {
                     case 1:   system.out.println ("Enter the amt  you
                               ~~if acc~~  want to deposit");
                               amount = sc.nextDouble();
                               deposit();
                     Case 2:   System.out.println ("Enter the amount
                               withdrawn); you  want to withdraw");
                     ~~Case 3~~ ~~sc.n~~
                               withdraw_amt  =  sc.nextDouble();
                               withdraw();
                     Case 3: acc_type = "~~current~~ current";
                         if  System.out.println ("Enter  the name of the
                                 customer you want to issue a cheque
                                       to ");
                          to_name  =  sc.next();
                          chequeBook();
                     }
                     else {
                          system.out.println ("Cheque book cannot
                                                be issued");
          }
}
```

29-10
Proceed



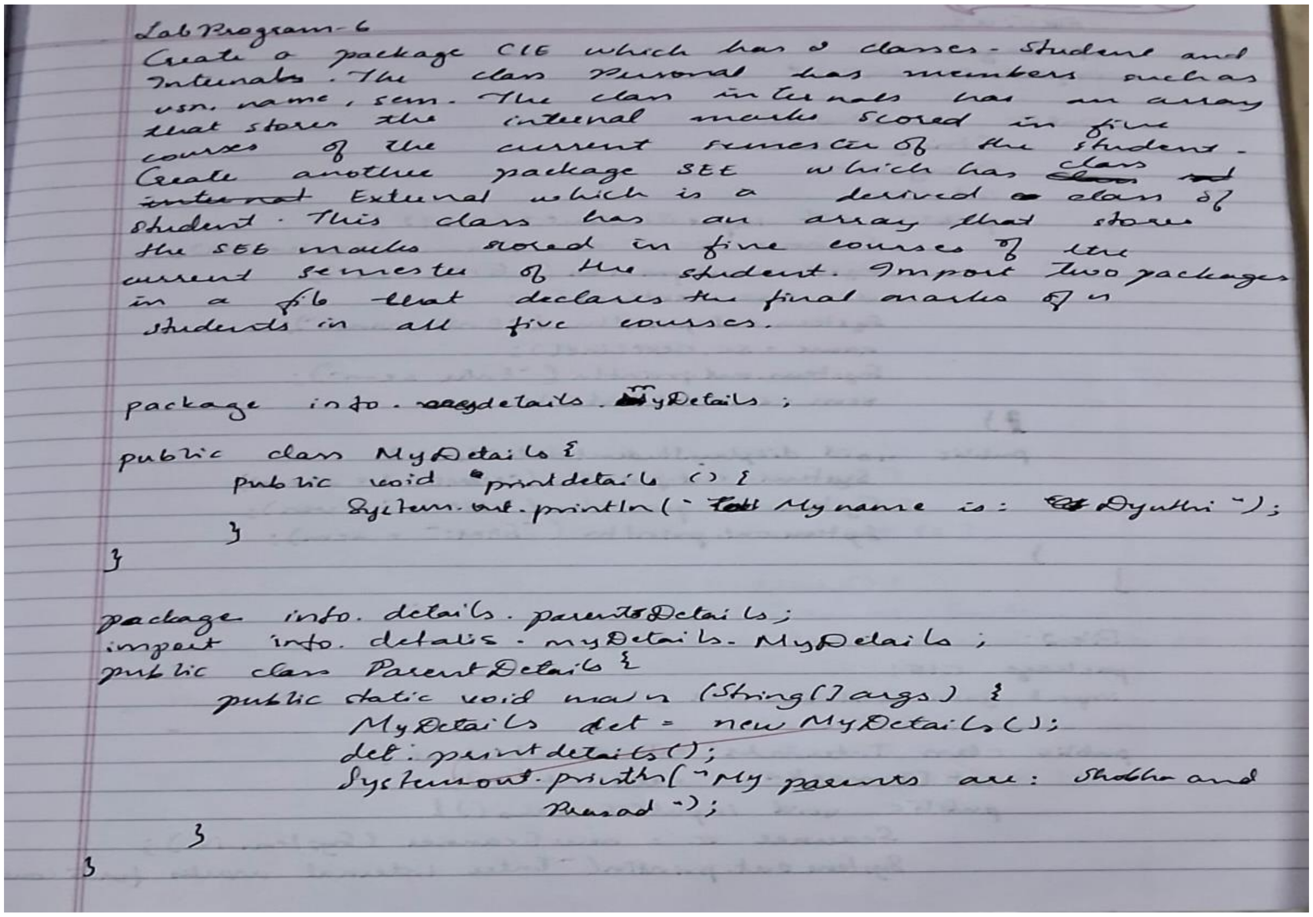
```
Output:

Enter your name and account number:
Dyuthi
789423
Input your balance:
453210
Enter your account type:
~~savings~~
current
Enter:
1. Deposit
2. Withdraw
3. Chequebook
4. Exit
1.
```

CODE:

```
import java.util.*;

abstract class Account {
    String customer_name;
    String acc_no;
    int acc_type;
    double balance = 0.0;


    Account(String customer_name, String acc_no, int acc_type) {
        this.customer_name = customer_name;
        this.acc_no = acc_no;
        this.acc_type = acc_type;
    }

    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
        displayBalance();
    }

    public void displayBalance() {
        System.out.println("The current balance is: " + balance);
    }

    public double getBalance() {
        return balance;
    }
    public void chequeBook(String to_name, double cheque_amount) {
        System.out.println("A cheque was issued to: " + to_name);
        balance -= cheque_amount;
        displayBalance();
    }
}

class Curr_acct extends Account {
    final double min_bal = 500.0;
    final double penalty = 50.0;

    Curr_acct(String customer_name, String acc_no, int acc_type) {
        super(customer_name, acc_no, acc_type);
    }

    public void checkMinBal() {
        if (balance < min_bal) {
            balance -= penalty;
            System.out.println("Penalty of " + penalty + " is issued due to low balance.");
        }
        displayBalance();
    }

    public void withdraw(double withdraw_amt) {
        if (withdraw_amt <= balance) {
            balance -= withdraw_amt;
            System.out.println("Withdrew: " + withdraw_amt);
        } else {
            System.out.println("Insufficient funds for withdrawal.");
        }
        displayBalance();
    }
}

class Sav_acct extends Account {
    final double interest_rate = 7.0;

    Sav_acct(String customer_name, String acc_no, int acc_type) {
        super(customer_name, acc_no, acc_type);
    }

    public void computeInterest() {
        double interest = balance * (interest_rate / 100);
        balance += interest;
        System.out.println("Interest added: " + interest);
        displayBalance();
    }

    public void withdraw(double withdraw_amt) {
        if (withdraw_amt <= balance) {
            balance -= withdraw_amt;
            System.out.println("Withdrew: " + withdraw_amt);
        } else {
            System.out.println("Insufficient funds for withdrawal.");
        }
        displayBalance();
    }
}

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


        System.out.println("Enter your name and account no: ");
        String customer_name = sc.next();
        String acc_no = sc.next();
        System.out.println("Input your initial balance: ");
        double balance = sc.nextDouble();

        System.out.println("Enter your account type:\nFor savings account enter 1\nFor current account
enter 2");
        int acc_type = sc.nextInt();

        Account account = null;
        if (acc_type == 1) {
            account = new Sav_acct(customer_name, acc_no, acc_type);
        } else if (acc_type == 2) {
            account = new Curr_acct(customer_name, acc_no, acc_type);
        } else {
            System.out.println("Invalid account type.");

        }

        account.deposit(balance);
        int ch;
        System.out.println("Enter:\n1: Deposit\n2: Withdraw\n3: Cheque book\n4: View Balance\n5:
Compute Interest (Savings)");
        ch = sc.nextInt();

        switch (ch) {
            case 1:
                System.out.println("Enter the amount you want to deposit: ");
                double depositAmount = sc.nextDouble();
                account.deposit(depositAmount);
                break;

            case 2:
                System.out.println("Enter the amount you want to withdraw: ");
                double withdrawAmount = sc.nextDouble();
                if (account instanceof Curr_acct) {
                    ((Curr_acct) account).withdraw(withdrawAmount);
                } else if (account instanceof Sav_acct) {
                    ((Sav_acct) account).withdraw(withdrawAmount);
                }
                break;

            case 3:
                System.out.println("Enter the payee name for the cheque: ");
                String to_name = sc.next();
                System.out.println("Enter the cheque amount: ");
                double chequeAmount = sc.nextDouble();
                account.chequeBook(to_name, chequeAmount);
                break;

            case 4:
                account.displayBalance();
                break;

            case 5:
                if (account instanceof Sav_acct) {
                    ((Sav_acct) account).computeInterest();
                } else {
                    System.out.println("Interest is only applicable to savings accounts.");
                }
                break;

            default:
                System.out.println("Invalid option.");
                break;
        }

        sc.close();
    }
}
```

PROGRAM 6:

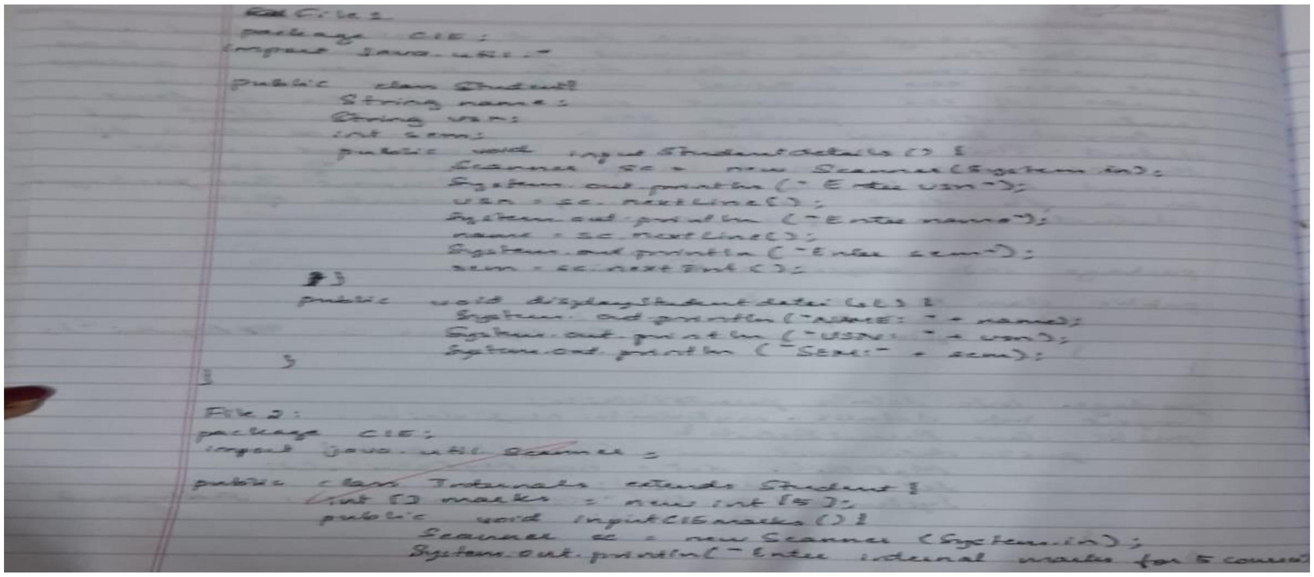
Packages



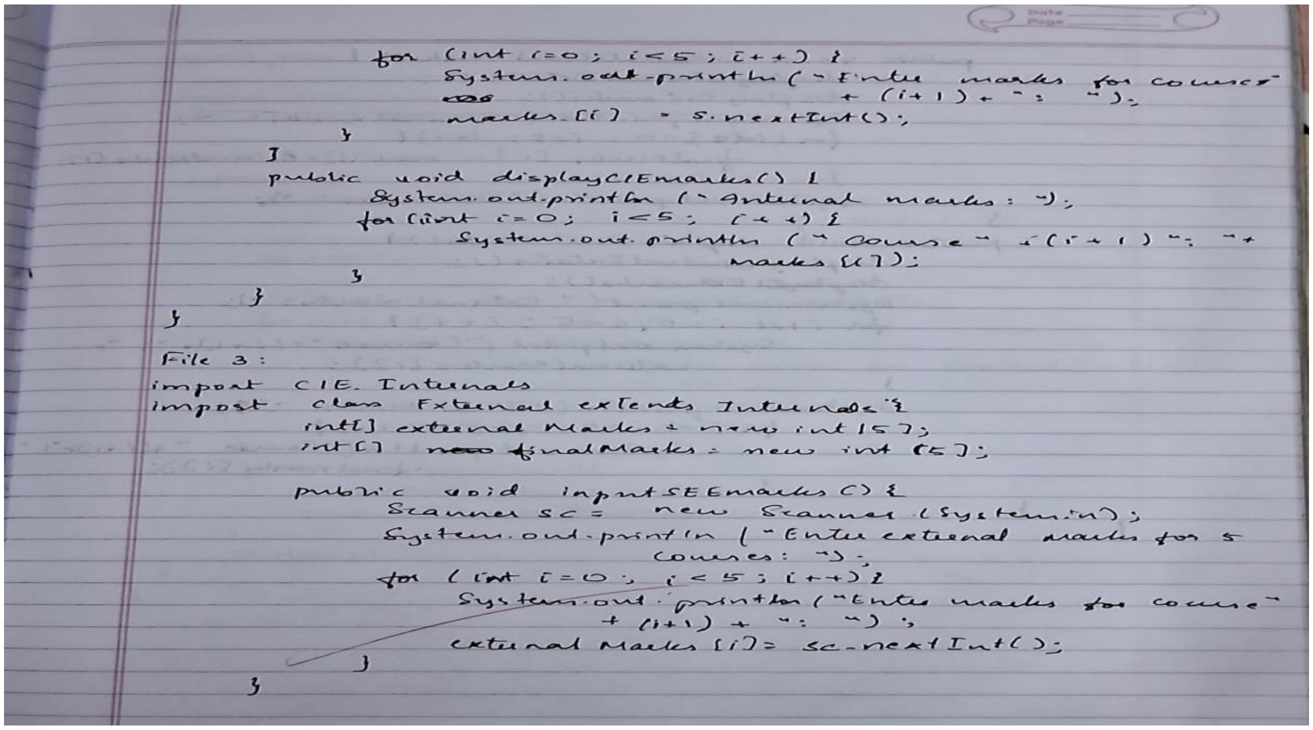
Lab Program-6

Create a package CIE which has 2 classes - Student and Internals. The class Personal has members such as usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has ~~class~~ class ~~and internal~~ 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 two packages in a file that declares the final marks of n students in all five courses.

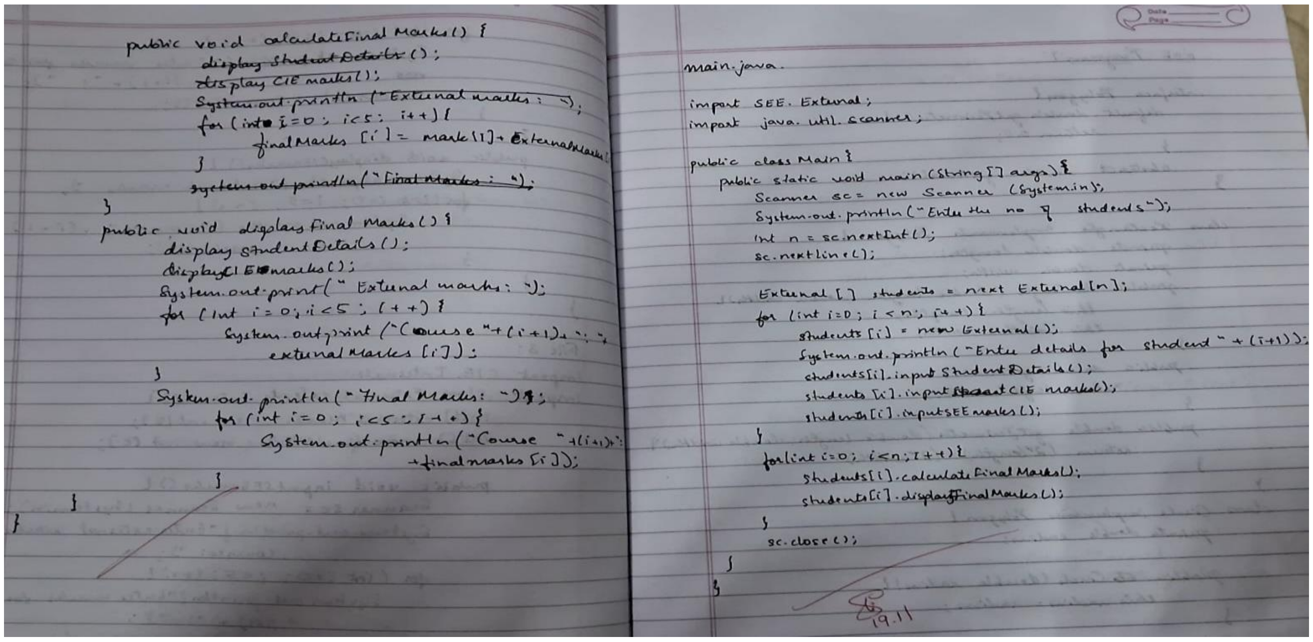
```
package info.~~mydetails~~details.MyDetails ;

public class MyDetails {
        public void printdetails () {
                System.out.println("My name is : Dyuthi");
        }
}

package info.details.parentsDetails;
import info.details.myDetails.MyDetails ;
public class ParentDetails {
        public static void main (String[] args) {
                MyDetails det = new MyDetails();
                det.printdetails();
                Systemout.println("My parents are : Shobha and
                        Prasad");
        }
}
```



```
File 1
package CIE ;
import java.util.*

public class Student{
        String name;
        String usn;
        int sem;
        public void inputStudentdetails () {
                Scanner sc = new Scanner(System.in);
                System.out.println(" Enter usn");
                usn = sc.nextLine();
                System.out.println ("Enter name");
                name = sc.nextLine();
                System.out.println ("Enter sem");
                sem = sc.nextInt ();
        }
        public void displayStudentdetails() {
                System.out.println("NAME: " + name);
                System.out.println ("USN: " + usn);
                System.out.println ("SEM:" + sem);
        }
}

File 2:
package CIE;
import java.util.Scanner ;

public class Internals extends Student {
        int [] marks = new int [5];
        public void inputCIEmarks () {
                Scanner sc = new Scanner (System.in);
                System.out.println("Enter internal marks for 5 courses
```



```
        for (int i=0; i<5; i++) {
            System.out.println("Enter marks for course"
                                        + (i+1) + ": ");
            marks [i] = s.nextInt();
        }
    }
    public void displayCIEmarks() {
        System.out.println ("Internal marks: ");
        for (int i=0; i<5; i++) {
            System.out.println ("Course " + (i+1) ": " +
                                  marks [i]);
        }
    }
}
File 3:
import CIE. Internals
import class External extends Internals {
        int[] external Marks = new int [5];
        int [] ~~new~~ finalMarks = new int [5];

        public void inputSEEmarks () {
            Scanner sc = new Scanner (System.in);
            System.out.println ("Enter external marks for 5
                                  courses: ");
            for ( int i=0; i<5; i++) {
                System.out.println ("Enter marks for course"
                          + (i+1) + ": ");
                external Marks [i]= sc.nextInt();
            }
        }
```



```
    public void calculateFinal Marks() {
            ~~display StudentDetails ();~~
            ~~display CIE marks();~~
            ~~System.out.println ("External marks: ");~~
            for (int i=0; i<5; i++) {
                finalMarks [i] = mark [i] + ExternalMarks [i];
            }
            ~~system.out.println("Final Marks: ");~~
    }
    public void displayFinal Marks() {
        display StudentDetails ();
        displayCIEmarks();
        System.out.print(" External marks: ");
        for (int i=0; i<5; i++) {
            System.out.print ("Course "+ (i+1)+ ": "+
                  externalMarks [i]);
        }
        System.out.println("Final Marks: ");
            for (int i=0; i<5; i++) {
                System.out.println("Course "+(i+1)+":
                                  +finalmarks [i]);
            }
    }
}
```

```
main.java.

import SEE. External;
import java. util. scanner;

public class Main {
    public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter the no of students");
        int n = sc.nextInt();
        sc.nextLine();

        External [] students = next External [n];
        for (int i=0; i<n; i++) {
            students [i] = new External ();
            System.out.println ("Enter details for student " + (i+1));
            students[i].input StudentDetails();
            students [i].input ~~[illegible]~~ CIE marks();
            students[i].inputSEEmarks();
        }
        for(int i=0; i<n; i++) {
            students[i].calculate Final Marks();
            students[i].displayFinalMarks();
        }
        sc.close();
    }
}
```

[illegible] 19.11

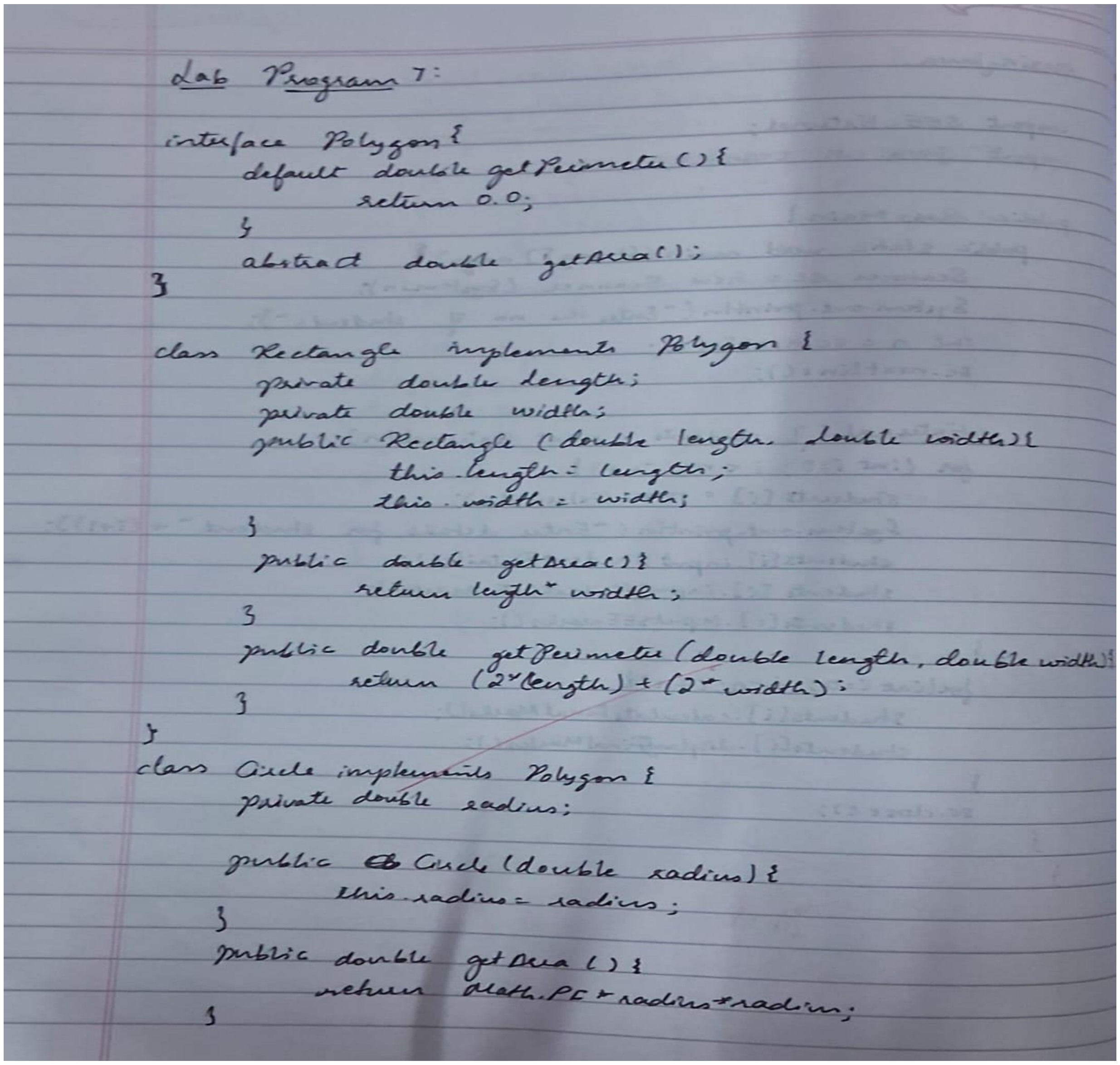
Code:

```
package CIE;

import java.util.Scanner;

public class Student {
    protected String usn;
    protected String name;
    protected int sem;

    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter USN: ");
        usn = s.nextLine();
        System.out.println("Enter Name: ");
        name = s.nextLine();
        System.out.println("Enter Semester: ");
        sem = s.nextInt();
    }

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

package SEE;

import CIE.Internals;
import java.util.Scanner;

public class External extends Internals {
    protected int[] externalMarks = new int[5];
    protected int[] finalMarks = new int[5];

    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter External Marks for 5 Courses: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter marks for course " + (i + 1) + ": ");
            externalMarks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + externalMarks[i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        displayCIEmarks();
        System.out.println("External Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + externalMarks[i]);
        }
        System.out.println("Final Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}
```

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
        sc.nextLine();

        External[] students = new External[n];

        for (int i = 0; i < n; i++) {
            students[i] = new External();
            System.out.println("Enter details for student " + (i + 1));
            students[i].inputStudentDetails();
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
        }

        for (int i = 0; i < n; i++) {
            students[i].calculateFinalMarks();
            students[i].displayFinalMarks();
        }
        sc.close();
    }
}
```

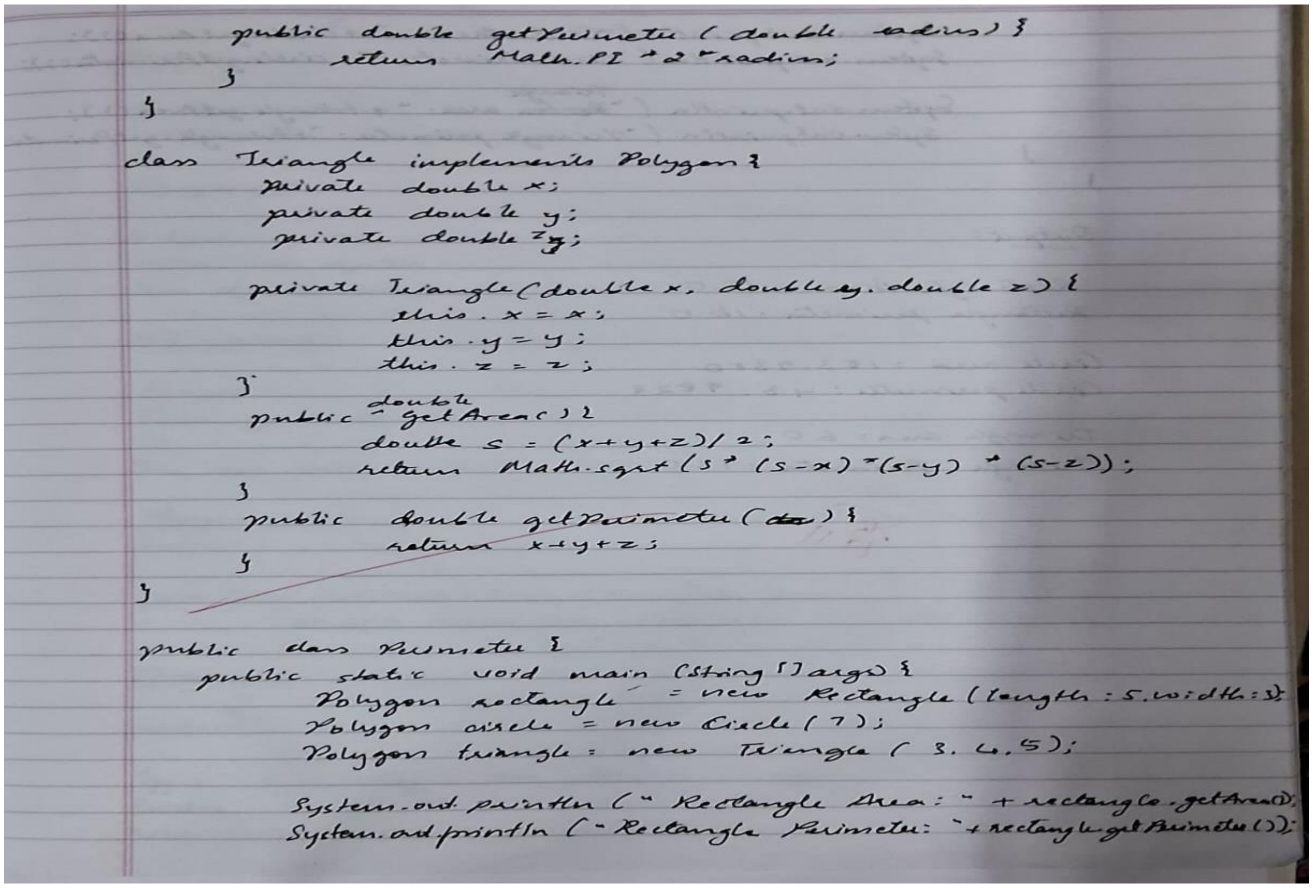
PROGRAM 7:

Interfaces

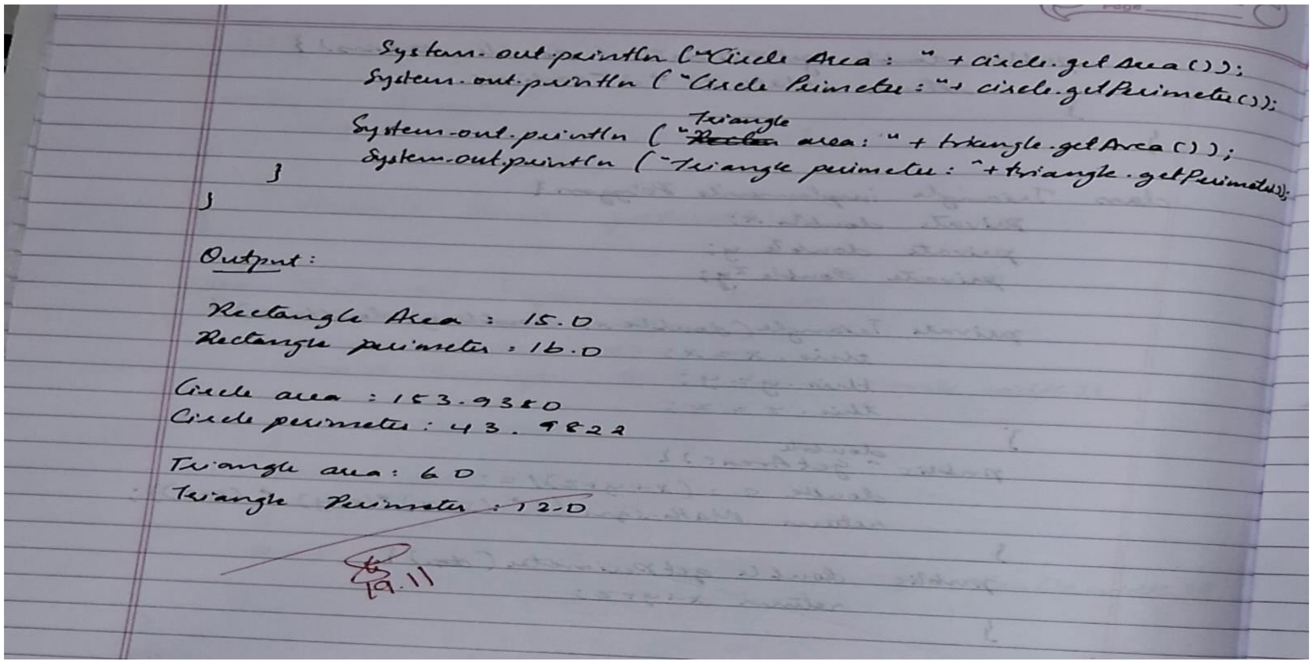


Lab Program 7:

```
interface Polygon{
        default double getPerimeter(){
                return 0.0;
        }
        abstract double getArea();
}

class Rectangle implements Polygon {
        private double length;
        private double width;
        public Rectangle (double length, double width){
                this.length = length;
                this.width = width;
        }
        public double getArea(){
                return length* width;
        }
        public double getPerimeter(double length, double width){
                return (2*length) + (2* width);
        }
}
class Circle implements Polygon {
        private double radius;

        public ~~C~~ Circle (double radius){
                this.radius = radius;
        }
        public double getArea (){
                return Math.PI * radius*radius;
        }
```



```
        public double getPerimeter ( double radius) {
              return   Math.PI * 2 * radius;
        }
  }

class  Triangle  implements Polygon {
          private  double x;
          private  double y;
          private  double z~~y~~;

          private Triangle(double x, double y, double z) {
                   this . x = x;
                   this . y = y;
                   this . z = z;
          }
                  double
          public ^ getArea() {
                   double s = (x+y+z)/2;
                   return  Math.sqrt (s * (s-x) * (s-y) * (s-z));
          }
          public   double getPerimeter(~~do~~) {
                   return   x+y+z;
          }
  }

public   class Perimeter {
    public static  void main (String [] args) {
           Polygon rectangle  = new  Rectangle (length : 5, width : 3);
           Polygon circle = new Circle (7);
           Polygon triangle =  new   Triangle ( 3, 4, 5);

           System.out.println (" Rectangle Area: " + rectangle.getArea());
           System.out.println (" Rectangle Perimeter: " + rectangle.getPerimeter());
```



```
                 System.out.println ("Circle Area :  " + circle.getArea());
                 System.out.println (" Circle Perimeter : "+ circle.getPerimeter());

                                      Triangle
                 System.out.println (" ~~Rectan~~ area: " + triangle.getArea());
                 System.out.println (" Triangle perimeter: " + triangle.getPerimeter());
          }
    }
```

Output:

```
Rectangle Area : 15.0
Rectangle perimeter : 16.0

Circle area : 153.9380
Circle perimeter : 43.9822

Triangle area : 6.0
Triangle Perimeter : 12.0
```

19.11

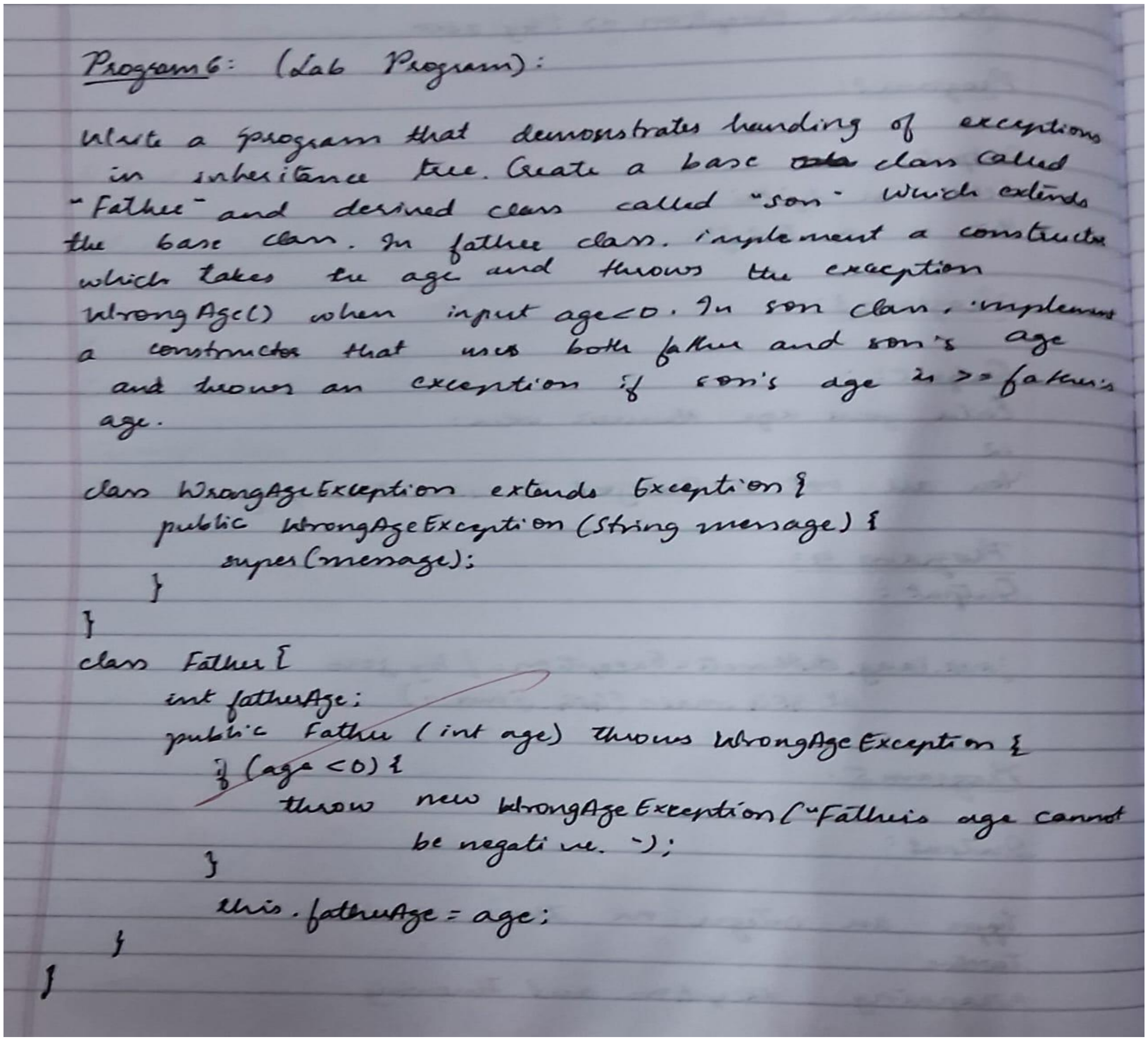
CODE:

```
interface Polygon {
   default double getPerimeter(){
      return 0.0;
   }
   abstract double getArea();
}

class Rectangle implements Polygon {
   private double length;
   private double width;

   public Rectangle(double length, double width) {
      this.length = length;
      this.width = width;
   }
   public double getArea() {
      return length * width;
   }
   public double getPerimeter(){
      return (2*length)+(2*width);
   }


}

class Circle implements Polygon {
   private double radius;

   public Circle(double radius) {
      this.radius = radius;
   }

   public double getArea() {
      return Math.PI * radius * radius;
   }
   public double getPerimeter(){
      return Math.PI*2*radius;
   }


}
class Triangle implements Polygon {
   private double x;
   private double y;
   private double z;
```

```
    public Triangle(double x, double y, double z) {
        this.x = x;
        this.y = y;
        this.z = z;
    }


    public double getArea() {
        double s = (x + y + z) / 2;
        return Math.sqrt(s * (s - x) * (s - y) * (s - z));
    }
    public double getPerimeter(){
        return x+y+z;
    }

}

public class Perimeter{
    public static void main(String[] args) {
        Polygon rectangle = new Rectangle(5, 3);
        Polygon circle = new Circle(7);
        Polygon triangle = new Triangle(3, 4, 5);

        System.out.println("Rectangle Area: " + rectangle.getArea());
        System.out.println("Rectangle Perimeter: " + rectangle.getPerimeter());

        System.out.println("Circle Area: " + circle.getArea());
        System.out.println("Circle Perimeter: " + circle.getPerimeter());

        System.out.println("Triangle Area: " + triangle.getArea());
        System.out.println("Triangle Perimeter: " + triangle.getPerimeter());
    }
}
```

PROGRAM 8:

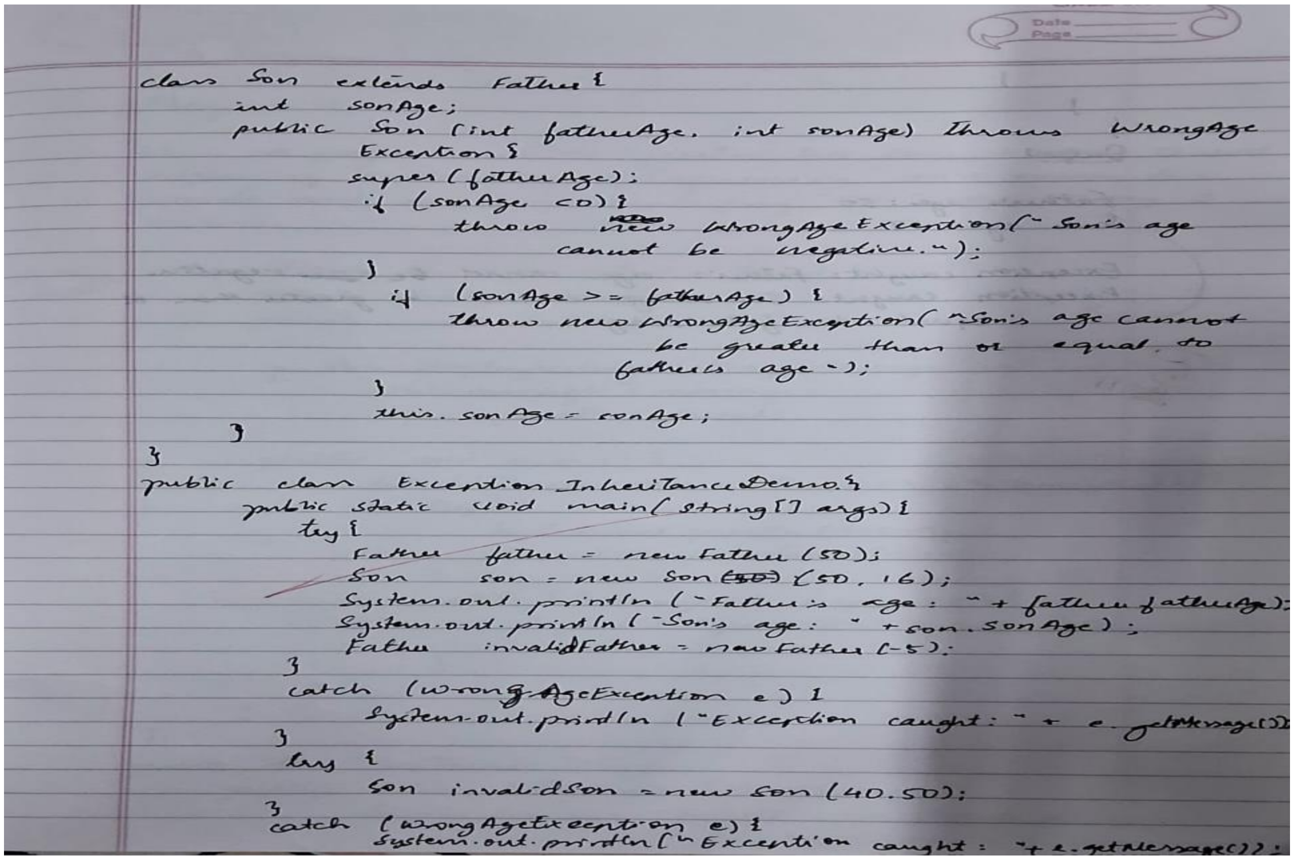
Exception handling



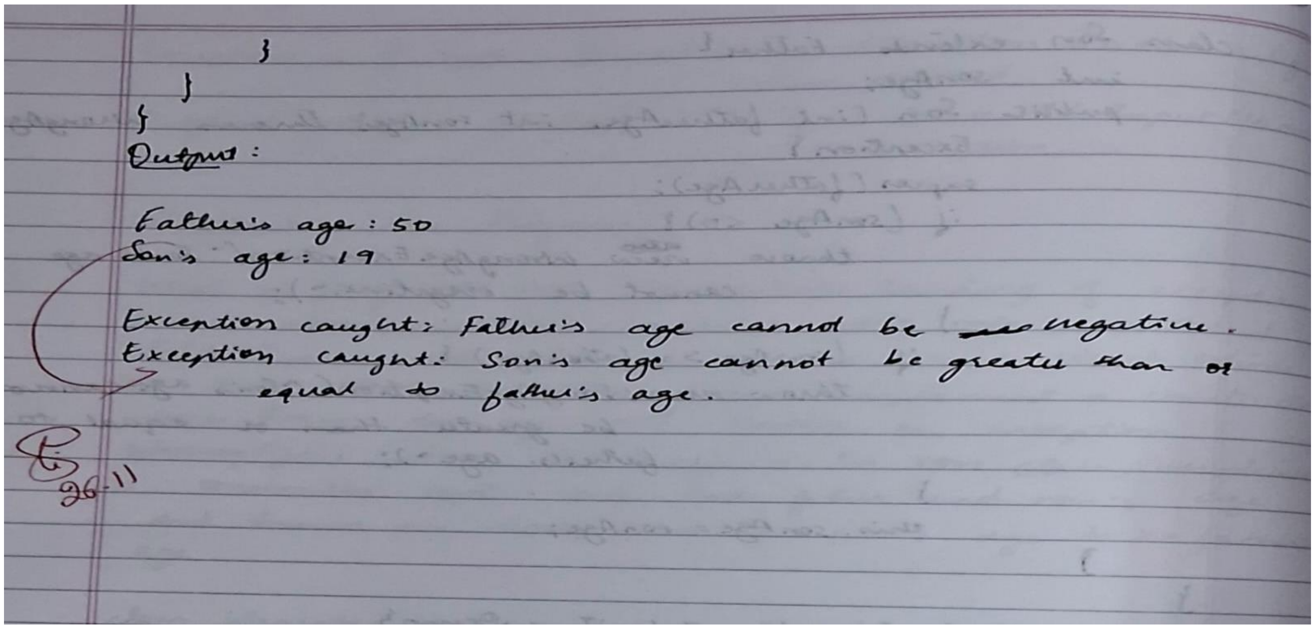
Program 6: (Lab Program):

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

```
class WrongAgeException extends Exception {
    public WrongAgeException (String message) {
        super(message);
    }
}
class Father {
    int fatherAge;
    public Father (int age) throws WrongAgeException {
        if (age <0) {
            throw new WrongAgeException ("Father's age cannot
                                          be negative.");
        }
        this.fatherAge = age;
    }
}
```



```
class Son extends Father {
    int sonAge;
    public Son (int fatherAge, int sonAge) Throws WrongAge
            Exception {
            super (fatherAge);
            if (sonAge <0) {
                    throw new WrongAgeException(" Son's age
                            cannot be negative.");
            }
             if  (sonAge >= fatherAge) {
                    throw new WrongAgeException( "Son's age cannot
                                be greater than or equal to
                            father's age ");
            }
            this. sonAge = sonAge;
    }
}
public class ExceptionInheritanceDemo {
    public static void main( String[] args) {
        try {
            Father   father = new Father (50);
            Son      son = new Son ~~(50)~~ (50, 16);
            System.out.println (" Father's  age : " + father.fatherAge);
            System.out.println ( "Son's age: " + son.sonAge) ;
            Father    invalidFather = new Father (-5);
        }
        catch (WrongAgeException e) {
            System.out.println ( "Exception caught: " + e.getMessage())
        }
        try {
            Son  invalidSon = new Son (40.50);
        }
        catch (WrongAgeException e) {
            System.out.println (" Exception caught: "+ e.getMessage());
```



```
        }
    }
}
```

<u>**Output**</u> :

```
Father's age : 50
Son's age: 19

Exception caught: Father's age cannot be ~~[illegible]~~ negative.
Exception caught: Son's age cannot be greater than or
        equal to father's age.
```

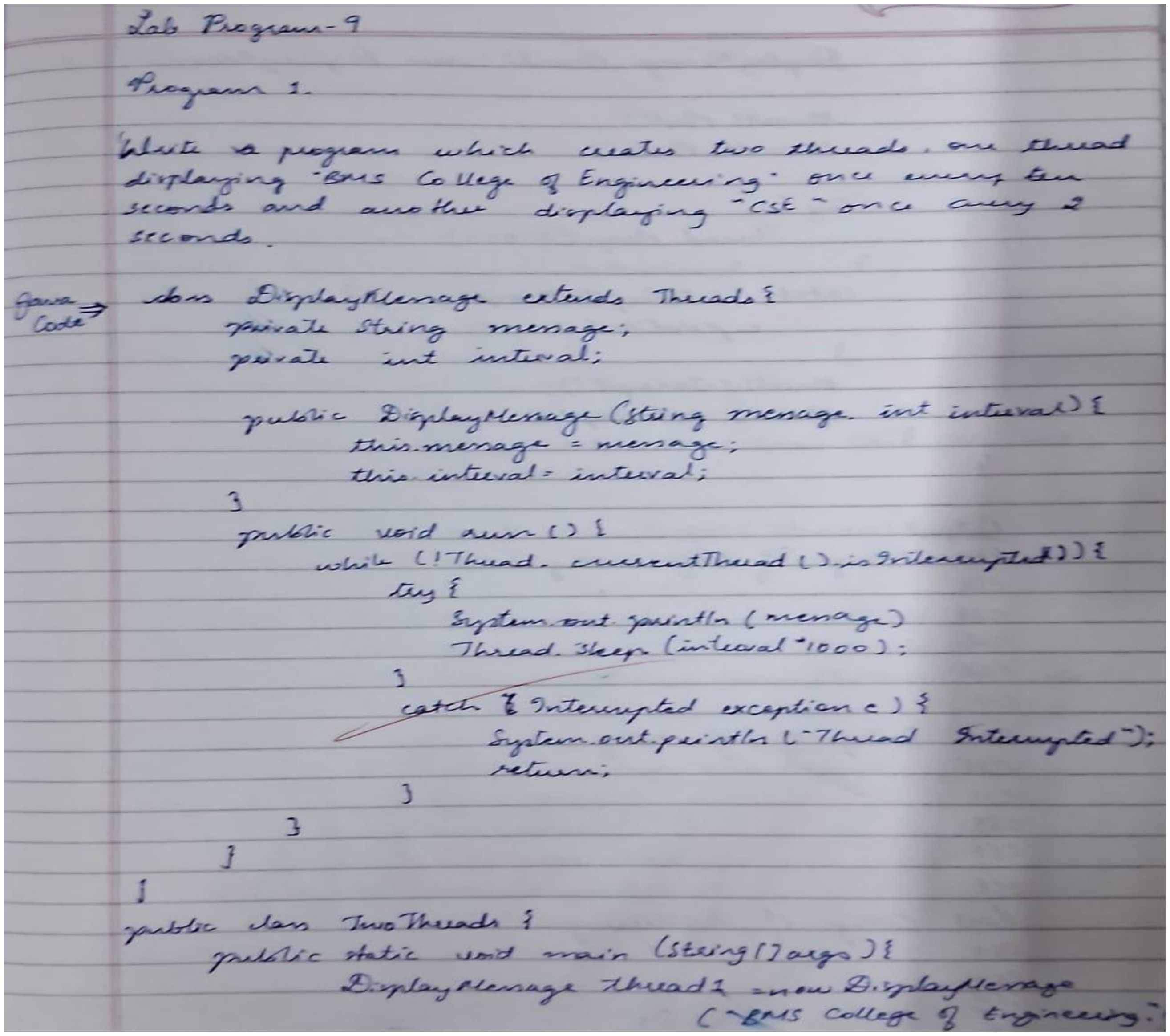
26-11

CODE:

```
class WrongAgeException extends Exception {
    public WrongAgeException(String message) {
        super(message);
    }
}
class Father {
    int fatherAge;
    public Father(int age) throws WrongAgeException {
        if (age < 0) {
            throw new WrongAgeException("Father's age cannot be negative.");
        }
        this.fatherAge = age;
    }
}
class Son extends Father {
    int sonAge;

    public Son(int fatherAge, int sonAge) throws WrongAgeException {
        super(fatherAge);
        if (sonAge < 0) {
            throw new WrongAgeException("Son's age cannot be negative.");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAgeException("Son's age cannot be greater than or equal to father's age.");
        }
        this.sonAge = sonAge;
    }
}

public class ExceptionInheritanceDemo {
    public static void main(String[] args) {
        try {
            Father father = new Father(40);
            Son son = new Son(40, 18);
            System.out.println("Father's age: " + father.fatherAge);
            System.out.println("Son's age: " + son.sonAge);
            Father invalidFather = new Father(-5);
        } catch (WrongAgeException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }

        try {
            Son invalidSon = new Son(40, 50);
        } catch (WrongAgeException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

PROGRAM 9:

Two threads



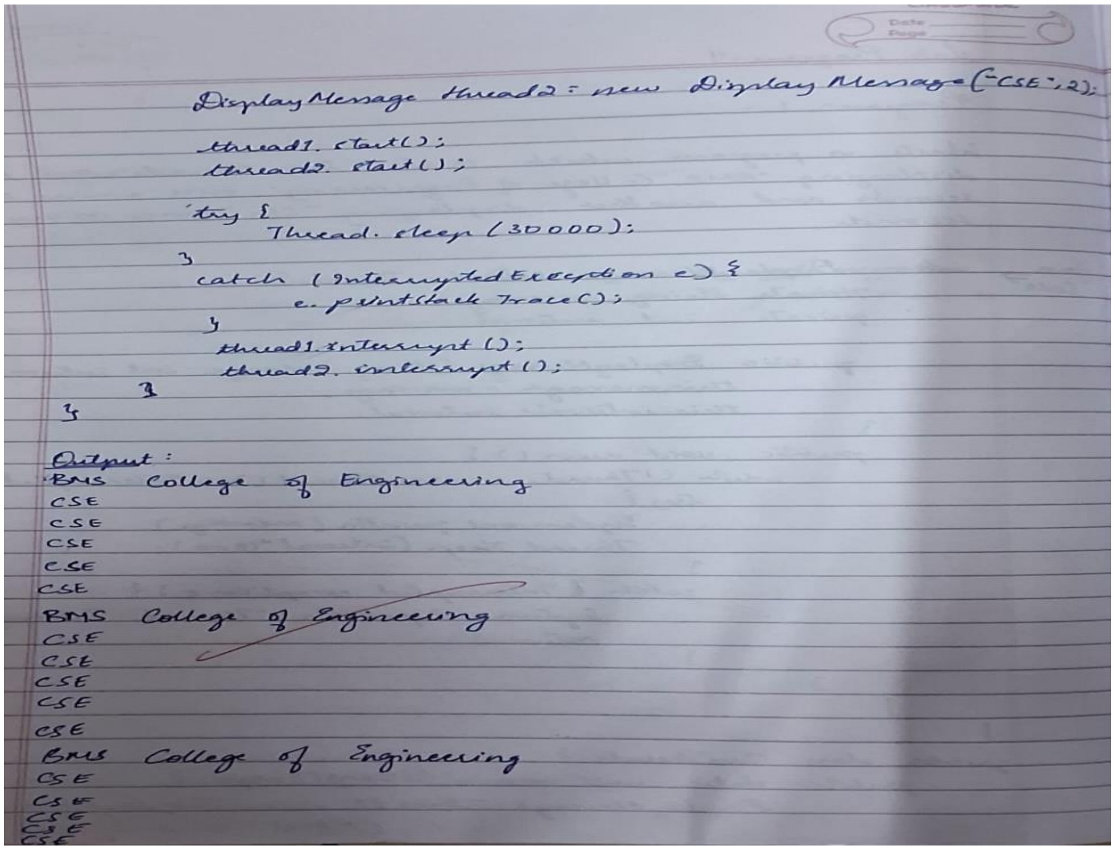
Lab Program-9

Program 1.

Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every 2 seconds.

Java Code ⇒

```
class DisplayMessage extends Threads {
    private String message;
    private int interval;

    public DisplayMessage (String message int interval) {
        this.message = message;
        this.interval = interval;
    }
    public void run () {
        while (!Thread. currentThread ().isInterrupted()) {
            try {
                System.out.println (message)
                Thread. sleep (interval*1000);
            }
            catch { InterruptedException e ) {
                System.out.println ("Thread Interrupted");
                return;
            }
        }
    }
}
public class TwoThreads {
    public static void main (String[] args) {
        DisplayMessage thread1 = new DisplayMessage
                    ("BMS College of Engineering".
```



```
        DisplayMessage thread2 = new DisplayMessage("CSE",2);

        thread1. start();
        thread2. start();

        try {
            Thread. sleep (30000);
        }
        catch (InterruptedException e) {
            e. printStackTrace();
        }
        thread1.interrupt ();
        thread2. interrupt ();
    }
}
```

Output :

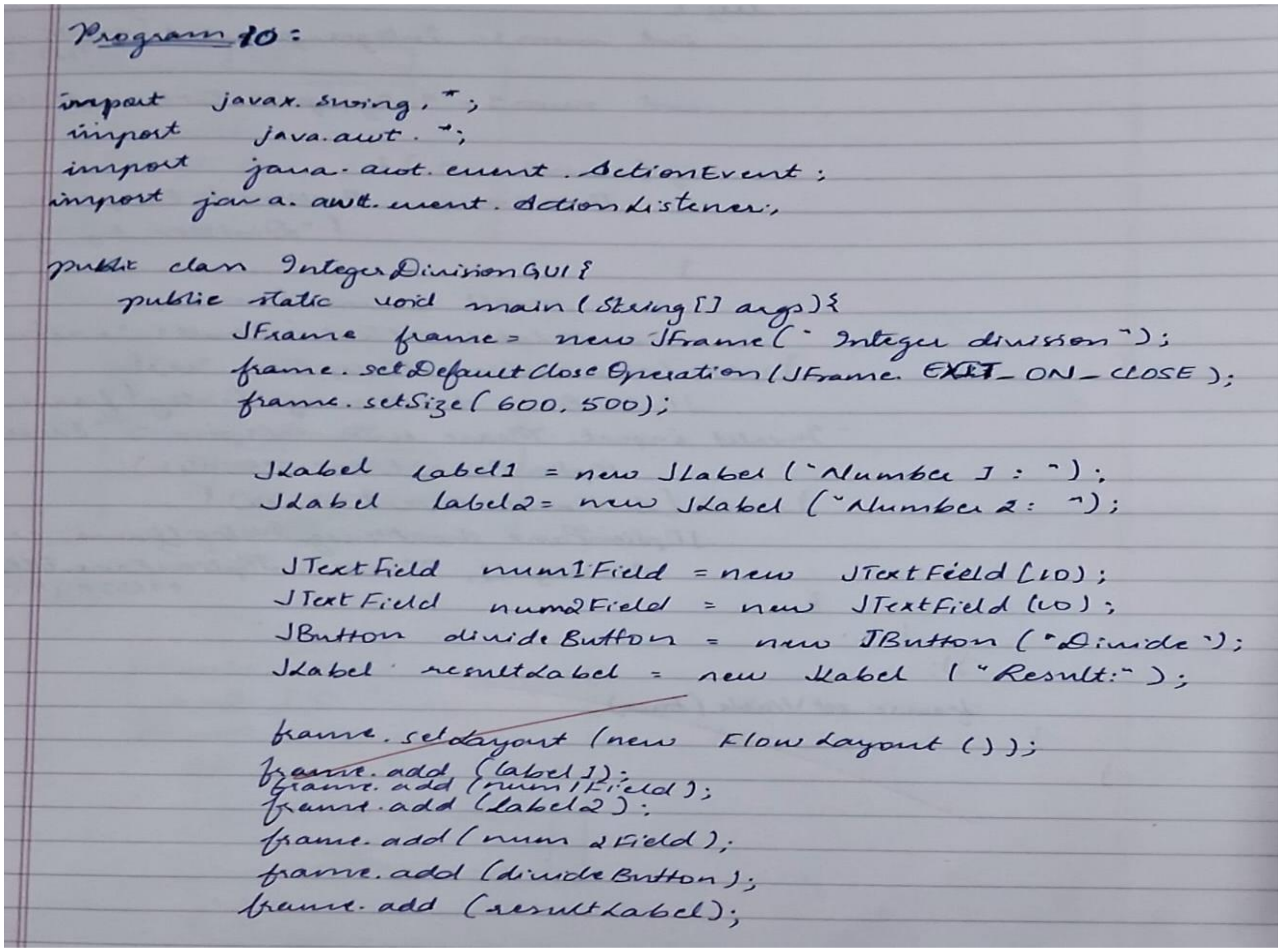
```
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
```

CODE:

```
class DisplayMessage extends Thread {
    private String message;
    private int interval;

    public DisplayMessage(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }


    public void run() {
        while (true) {
            try {
                System.out.println(message);
                Thread.sleep(interval * 1000);
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted");
            }
        }
    }
}

public class TwoThreads {
    public static void main(String[] args) {
        DisplayMessage thread1 = new DisplayMessage("BMS College of Engineering", 10);
        DisplayMessage thread2 = new DisplayMessage("CSE", 2);

        thread1.start();
        thread2.start();
    }
}
```

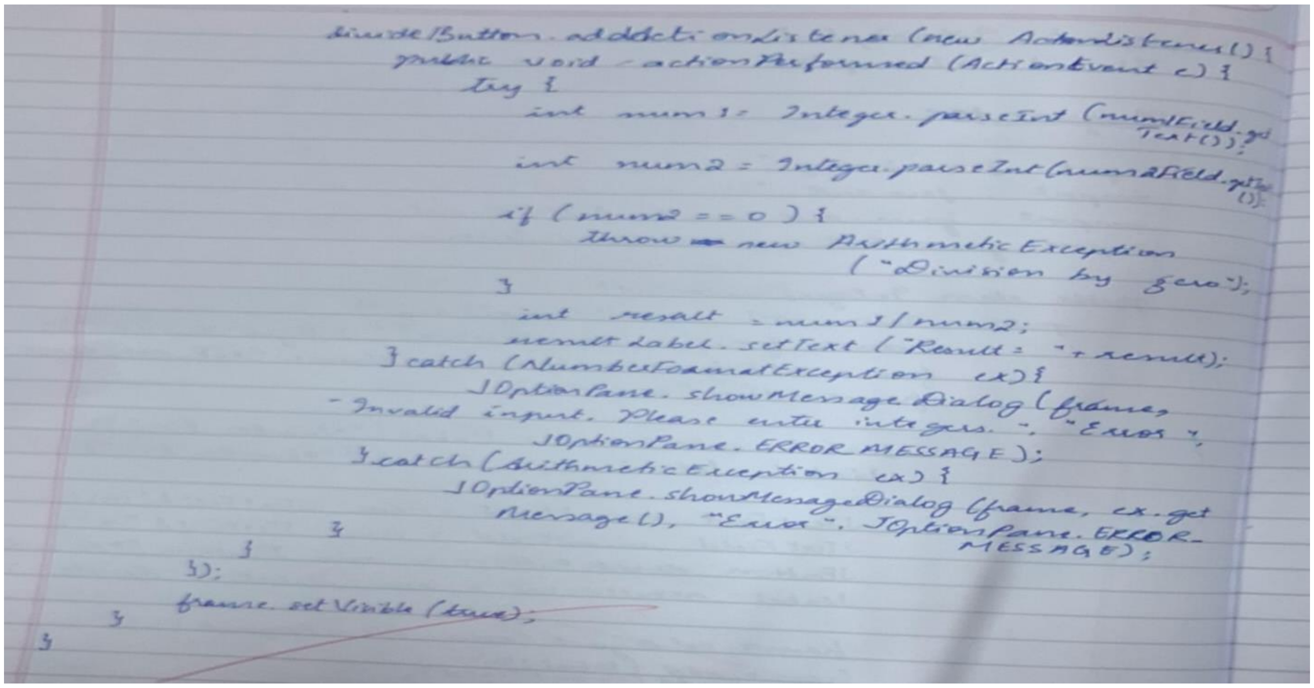
PROGRAM 10:

Integer division GUI

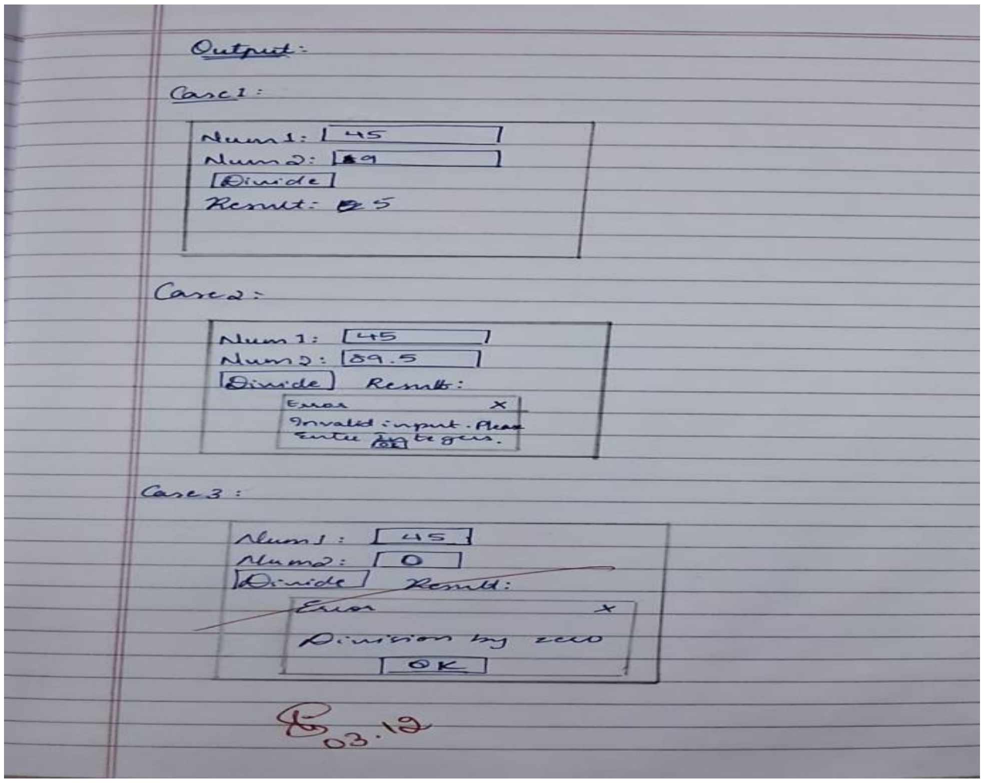


Program 10:

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

public class IntegerDivisionGUI{
    public static void main(String[] args){
        JFrame frame= new JFrame("Integer division");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(600, 500);

        JLabel label1 = new JLabel("Number 1: ");
        JLabel label2= new JLabel("Number 2: ");

        JTextField num1Field = new JTextField(10);
        JTextField num2Field = new JTextField(10);
        JButton divideButton = new JButton("Divide");
        JLabel resultLabel = new JLabel("Result:");

        frame.setLayout(new FlowLayout());
        frame.add(label1);
        frame.add(num1Field);
        frame.add(label2);
        frame.add(num2Field);
        frame.add(divideButton);
        frame.add(resultLabel);
```



```
divideButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());
            if (num2 == 0) {
                throw new ArithmeticException
                            ("Division by zero");
            }
            int result = num1/num2;
            resultLabel.setText("Result: " + result);
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(frame,
"Invalid input. Please enter integers.", "Error",
            JOptionPane.ERROR_MESSAGE);
        } catch (ArithmeticException ex) {
            JOptionPane.showMessageDialog(frame, ex.get
            Message(), "Error", JOptionPane.ERROR_
                                MESSAGE);
        }
    }
});
frame.setVisible(true);
}
}
```



Output:

Case1:

```
Num1: [45      ]
Num2: [9       ]
[Divide]
Result: 5
```

Case2:

```
Num1: [45      ]
Num2: [89.5    ]
[Divide]  Result:
   Error                 x
   Invalid input. Please
   enter integers.
```

Case 3:

```
Num1: [45]
Num2: [0 ]
[Divide]  Result:
   Error                 x
   Division by zero
        [OK]
```

[illegible] 03.12

CODE:

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

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

        JFrame frame = new JFrame("Integer Division");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);

        JLabel label1 = new JLabel("Num1:");
        JLabel label2 = new JLabel("Num2:");
        JTextField num1Field = new JTextField(10);
        JTextField num2Field = new JTextField(10);
        JButton divideButton = new JButton("Divide");
        JLabel resultLabel = new JLabel("Result: ");

        frame.setLayout(new FlowLayout());
        frame.add(label1);
        frame.add(num1Field);
        frame.add(label2);
        frame.add(num2Field);
        frame.add(divideButton);
        frame.add(resultLabel);

        divideButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                try {
                    int num1 = Integer.parseInt(num1Field.getText());
                    int num2 = Integer.parseInt(num2Field.getText());

                    if (num2 == 0) {
                        throw new ArithmeticException("Division by zero");
                    }

                    int result = num1 / num2;
                    resultLabel.setText("Result: " + result);
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(frame, "Invalid input. Please enter integers.", "Error",
JOptionPane.ERROR_MESSAGE);
                } catch (ArithmeticException ex) {
                    JOptionPane.showMessageDialog(frame, ex.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
```

```
                }
            }
        });
        frame.setVisible(true);
    }
}
```