VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Simrik Sharma (1BF24CS293)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Aug-2025 to Jan-2026

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 **SIMRIK SHARMA (1BF24CS293),** who is bonafide student of **B.M.S. College of Engineering.** It is in partial fulfilment 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.

Dr. Seema Patil Associate Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE

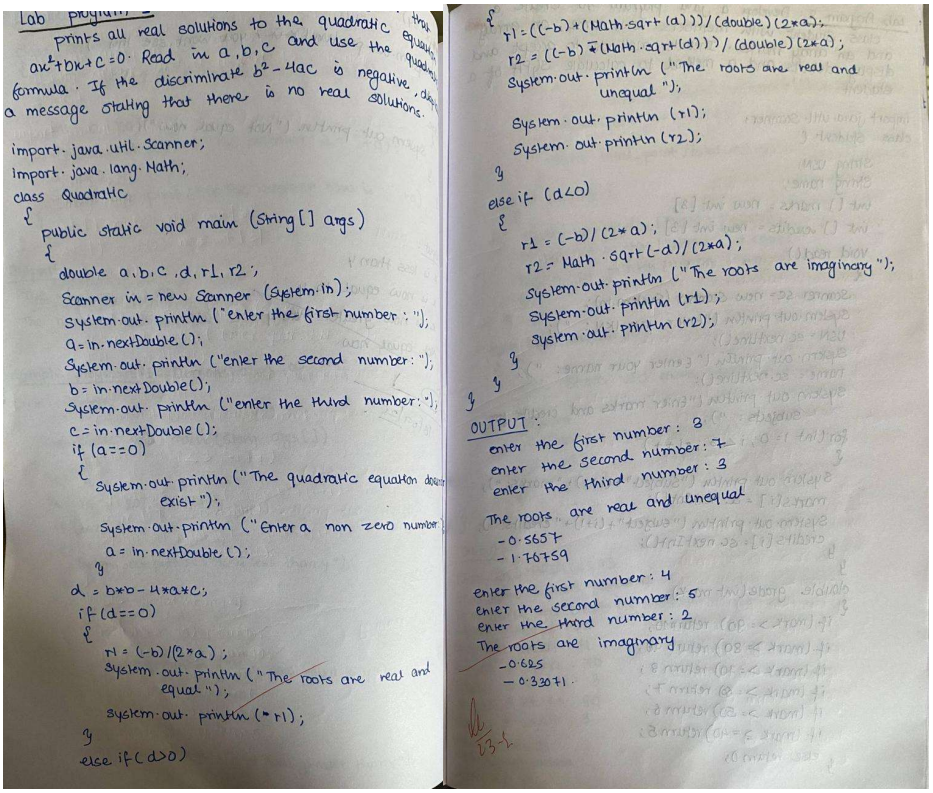
Index

		d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.	
6	18/11/25	Create a package CIE which has two classes - Personal and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Personal. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.	22
7	25/11/25	Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age=father's age.	25
8	09/12/25	Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds	27
9	09/12/25	Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.	29
10	09/12/25	Demonstrate Inter process Communication and deadlock.	31

Github Link:
https://github.com/Simrik-Sharma/Java-1BF24CS293

Program 1

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



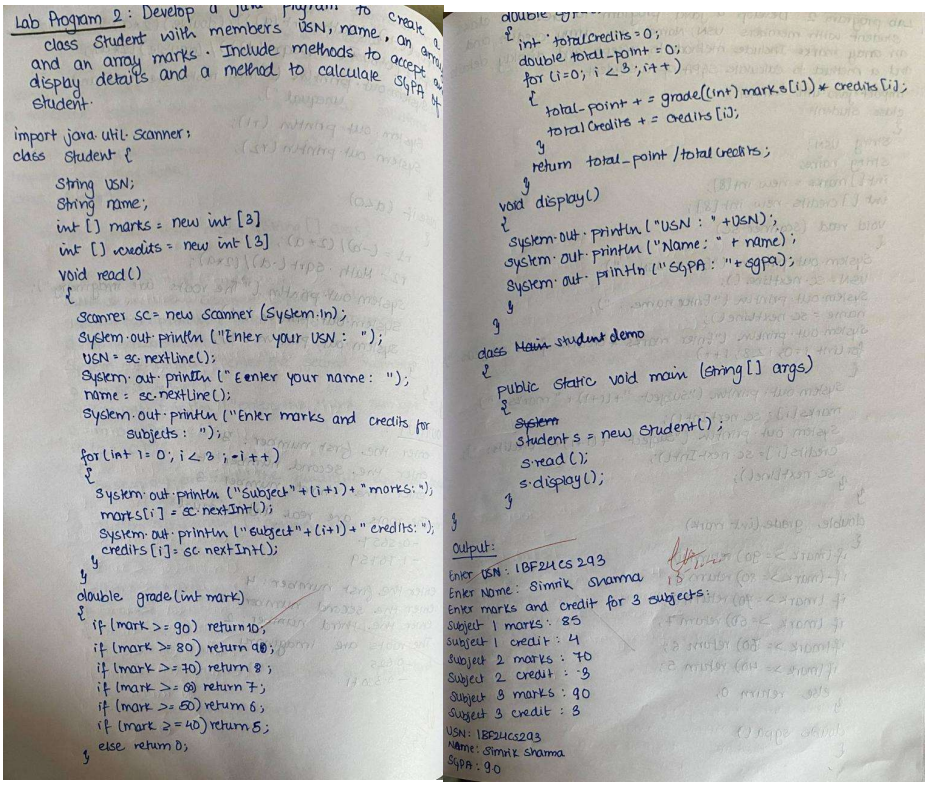
Code:

```
import java.util.Scanner;
import java.lang.Math;
class Quadratic
{
   public static void main (String []args)
   {
```

```
        double a, b, c, d, r1, r2;
        Scanner in = new Scanner (System.in);
        System.out.println ("Enter the first number: ");
        a = in.nextDouble();
        System.out.println ("Enter the second number: ");
        b = in.nextDouble();
        System.out.println ("Enter the third number: ");
        c = in.nextDouble();
        if (a==0)
        {
           System.out.println ("the quadratic equation does not exist");
           System.out.println ("Enter the a non zero number: ");
           a = in.nextDouble();
        }
     d = b*b - 4*a*c;
     if (d==0)
     {
        r1 = (-b)/(2*a);
        System.out.println ("The roots are real and equal ");
        System.out.println (r1);
     }
     else if (d>0)
     {
        r1 = ((-b) + (Math.sqrt(d)))/(double)(2*a);
        r2 = ((-b) - (Math.sqrt(d)))/(double)(2*a);
        System.out.println ("The roots are real and unequal ");
        System.out.println (r1);
        System.out.println (r2);
     }
     else if (d<0)
     {
        r1 = (-b)/(2*a);
        r2 = Math.sqrt(-d)/(2*a);
        System.out.println ("The roots are imaginary");
        System.out.println (r1);
        System.out.println (r2);
     }
     }
}
```

Program 2

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



Code:

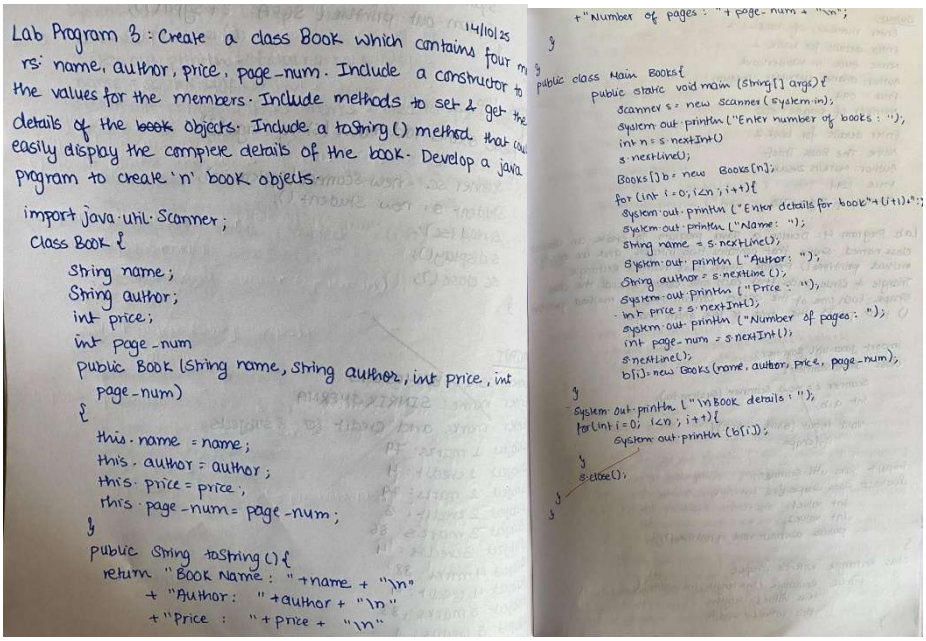
```
import java.util.Scanner;
class Student
{
   String USN;
   String name;
   int[] marks = new int[8];
   int[] credits = new int[8];
   void read(Scanner sc)
   {
      System.out.println("Enter USN:");
```

```
        USN = sc.nextLine();
        System.out.println("Enter Name:");
        name = sc.nextLine();
        System.out.println("Enter marks and credits for 8 subjects:");
        for (int i = 0; i < 8; i++)
        {
            System.out.println("Subject " + (i + 1) + " marks:");
            marks[i] = sc.nextInt();
            System.out.println("Subject " + (i + 1) + " credits:");
            credits[i] = sc.nextInt();
            sc.nextLine();
        }
    }
    double gradePoint(int mark)
    {
        if (mark >= 90) return 10;
        else if (mark >= 80) return 9;
        else if (mark >= 70) return 8;
        else if (mark >= 60) return 7;
        else if (mark >= 50) return 6;
        else if (mark >= 40) return 5;
        else return 0;
    }
    double sgpa()
    {
        int totalCredits = 0;
        double totalPoints = 0;
        for (int i = 0; i < 8; i++)
        {
            totalPoints += gradePoint(marks[i]) * credits[i];
            totalCredits += credits[i];
        }
        return totalPoints / totalCredits;
    }
    void display()
    {
        System.out.println("USN: " + USN);
        System.out.println("Name: " + name);
        System.out.printf("SGPA: %.2f\n", sgpa());
    }
}
```

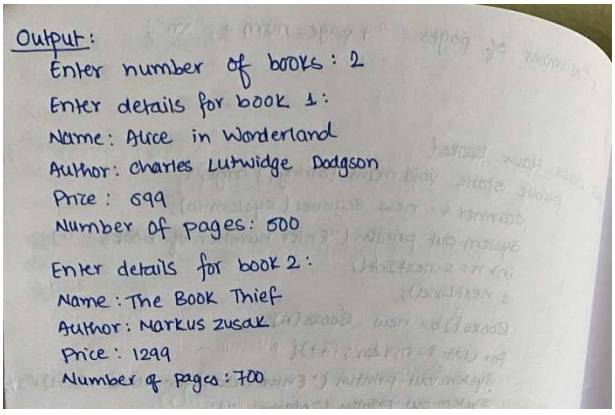
```
class gpaa
{
   public static void main(String args[])
   {
      Scanner sc = new Scanner(System.in);
      Student s1 = new Student();
      System.out.println("Enter details for Student 1:");
      s1.read(sc);
      System.out.println("Details for Student 1:");
      s1.display();
      System.out.println();
      Student s2 = new Student();
      System.out.println("Enter details for Student 2:");
      s2.read(sc);
      System.out.println("Details for Student 2:");
      s2.display();
      sc.close();
   }
}
```

Program 3

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



```
14/10/25
Lab Program 3 : Create a class Book which contains four m
rs: name, author, price, page_num. Include a constructor to
the values for the members. Include methods to set & get the
details of the book objects. Include a toString() method that cou
easily display the complete details of the book. Develop a java
program to create 'n' book objects.

import java.util.Scanner;
Class Book {
    String name;
    String author;
    int price;
    int page_num
    public Book (String name, String author, int price, int
      page_num)
    {
      this.name = name;
      this.author = author;
      this.price = price;
      this.page_num = page_num;
    }
    public String toString(){
      return "Book Name : " +name + "\n"
          + "Author :  " +author + "\n"
          +"Price :  " + price +  "\n"
          +"Number of pages : " + page_num + "\n";
    }
public class Main Books{
    public static void main (String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter number of books : ");
        int n = s.nextInt()
        s.nextLine();
        Books[] b = new Books[n];
        for(int i=0;i<n;i++){
          System.out.println("Enter details for book"+(i+1)+":");
          System.out.println("Name: ");
          String name = s.nextLine();
          System.out.println("Author: ");
          String author = s.nextLine();
          System.out.println("Price : ");
          int price = s.nextInt();
          System.out.println("Number of pages: ");
          int page_num = s.nextInt();
          s.nextLine();
          b[i]= new Books(name, author, price, page_num);
    }
    System.out.println("\n Book details : ");
    for(int i=0; i<n; i++){
        System.out.println(b[i]);
    }
    s.close();
}
}
```



```
Output:
Enter number of books : 2
Enter details for book 1:
Name: Alice in Wonderland
Author: Charles Lutwidge Dodgson
Price : 599
Number of pages: 500
Enter details for book 2:
Name : The Book Thief
Author: Markus Zusak
Price : 1299
Number of pages : 700
```

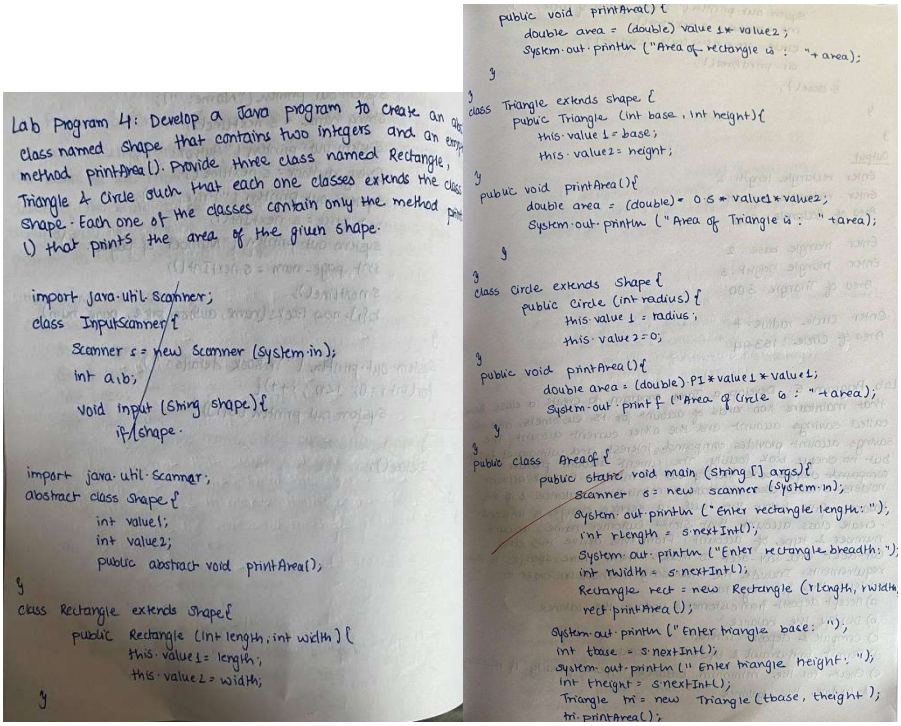
Code:

```
import java.util.Scanner;
class Books {
   String name;
   String author;
   int price;
   int numPages;
   Books(String name, String author, int price, int numPages) {
      this.name = name;
      this.author = author;
      this.price = price;
      this.numPages = numPages;
   }
   public String toString() {
      String name, author, price, numPages;
      name = "Book name: " + this.name + "\n";
      author = "Author name: " + this.author + "\n";
      price = "Price: " + this.price + "\n";
      numPages = "Number of pages: " + this.numPages + "\n";
      return name + author + price + numPages;
   }
}
public class Main {
   public static void main(String args[]) {
      Scanner s = new Scanner(System.in);
      int n, price, numPages;
      String name, author;
      System.out.print("Enter number of books: ");
      n = s.nextInt();
      Books[] b = new Books[n];
      for (int i = 0; i < n; i++) {
         System.out.println("\nEnter details of Book " + (i + 1) + ":");
         System.out.print("Enter book name: ");
         name = s.next();
         System.out.print("Enter author name: ");
         author = s.next();
         System.out.print("Enter price: ");
         price = s.nextInt();
         System.out.print("Enter number of pages: ");
         numPages = s.nextInt();
         b[i] = new Books(name, author, price, numPages);
```

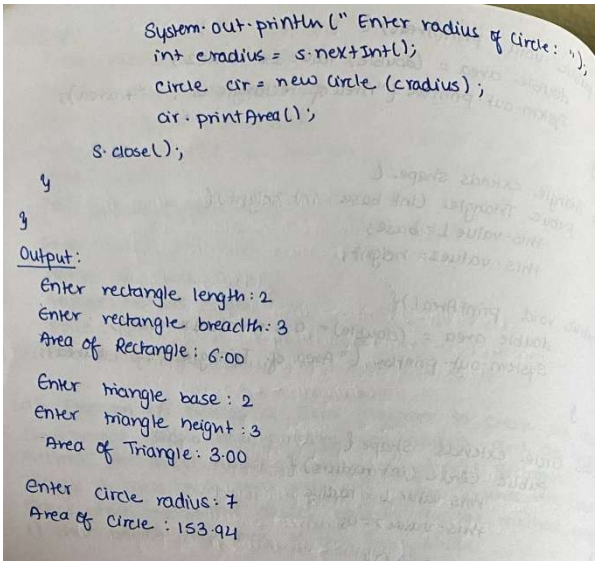
```
        }
        System.out.println("\n--- Book Details ---");
        for (int i = 0; i < n; i++) {
            System.out.println("\nBook " + (i + 1) + " Details:");
            System.out.println(b[i].toString());
        }
        s.close();
    }
}
```

Program 4

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



```
Lab Program 4: Develop a Java program to create an ab
class named Shape that contains two integers and an emp
method printArea(). Provide three class named Rectangle,
Triangle & Circle such that each one classes extends the class
Shape. Each one of the classes contain only the method print
() that prints the area of the given shape.

import java.util.Scanner;
class InputScanner{
    Scanner s = new Scanner (System.in);
    int a,b;
    void input (String shape){
        if(shape.

import java.util.Scanner;
abstract class Shape{
        int value1;
        int value2;
        public abstract void printArea();
}
class Rectangle extends Shape{
    public Rectangle (int length, int width){
            this.value1 = length;
            this.value2 = width;
    }
    public void printArea(){
        double area = (double) value1 * value2;
        System.out.println ("Area of rectangle is : "+area);
    }
}
class Triangle extends Shape{
    public Triangle (int base, int height){
        this.value1 = base;
        this.value2 = height;
    }
public void printArea(){
        double area = (double) 0.5 * value1 * value2;
        System.out.println ("Area of Triangle is : "+area);
    }
}
class Circle extends Shape{
    public Circle (int radius){
        this.value1 = radius;
        this.value2 = 0;
    }
public void printArea(){
        double area = (double) PI * value1 * value1;
        System.out.printf ("Area of Circle is : "+area);
    }
}
public class AreaOf{
    public static void main (String[] args){
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter rectangle length: ");
        int rlength = s.nextInt();
        System.out.println ("Enter rectangle breadth: ");
        int rwidth = s.nextInt();
        Rectangle rect = new Rectangle (rlength, rwidth
        rect.printArea();
    System.out.println (" Enter triangle base: ");
    int tbase = s.nextInt();
    System.out.println (" Enter triangle height: ");
    int theight = s.nextInt();
    Triangle tri = new Triangle (tbase, theight);
    tri.printArea();
```



```
        System.out.println (" Enter radius of Circle: ");
        int cradius = s.nextInt();
        Circle cir = new Circle (cradius);
        cir.printArea();
    s.close();
  }
}
```

Output:

```
Enter rectangle length: 2
Enter rectangle breadth: 3
Area of Rectangle: 6.00

Enter triangle base: 2
Enter triangle height: 3
Area of Triangle: 3.00

Enter Circle radius: 7
Area of Circle: 153.94
```

Code:

```
import java.util.Scanner;
abstract class Shape {
   int value1;
   int value2;
   public abstract void printArea();
}
class Rectangle extends Shape {
   public Rectangle(int length, int width) {
      this.value1 = length;
      this.value2 = width;
   }
   public void printArea() {
      // Area = Length * Width
      double area = (double)value1 * value2;
      System.out.printf("Area of Rectangle (L=%d, W=%d): %.2f\n", value1, value2, area);
   }
}
class Triangle extends Shape {
   public Triangle(int base, int height) {
      this.value1 = base;
      this.value2 = height;
   }
   public void printArea() {
      // Area = 0.5 * Base * Height
      double area = 0.5 * value1 * value2;
      System.out.printf("Area of Triangle (B=%d, H=%d): %.2f\n", value1, value2, area);
   }
}
class Circle extends Shape {
   private static final double PI = Math.PI;
   public Circle(int radius) {
      this.value1 = radius;
      this.value2 = 0;
   }
   public void printArea() {
      double area = PI * value1 * value1;
      System.out.printf("Area of Circle (Radius=%d): %.2f\n", value1, area);
   }
}
public class shapearea {
```

```
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println("--- Shape Area Calculation Program ---");
        System.out.print("Enter Rectangle Length: ");
        int rLength = scanner.nextInt();
        System.out.print("Enter Rectangle Width: ");
        int rWidth = scanner.nextInt();
        Rectangle rect = new Rectangle(rLength, rWidth);
        rect.printArea();
        System.out.print("\nEnter Triangle Base: ");
        int tBase = scanner.nextInt();
        System.out.print("Enter Triangle Height: ");
        int tHeight = scanner.nextInt();
        Triangle tri = new Triangle(tBase, tHeight);
        tri.printArea();
        System.out.print("\nEnter Circle Radius: ");
        int cRadius = scanner.nextInt();
        Circle circ = new Circle(cRadius);
        circ.printArea();

        scanner.close();

    }
}
```

Program 5

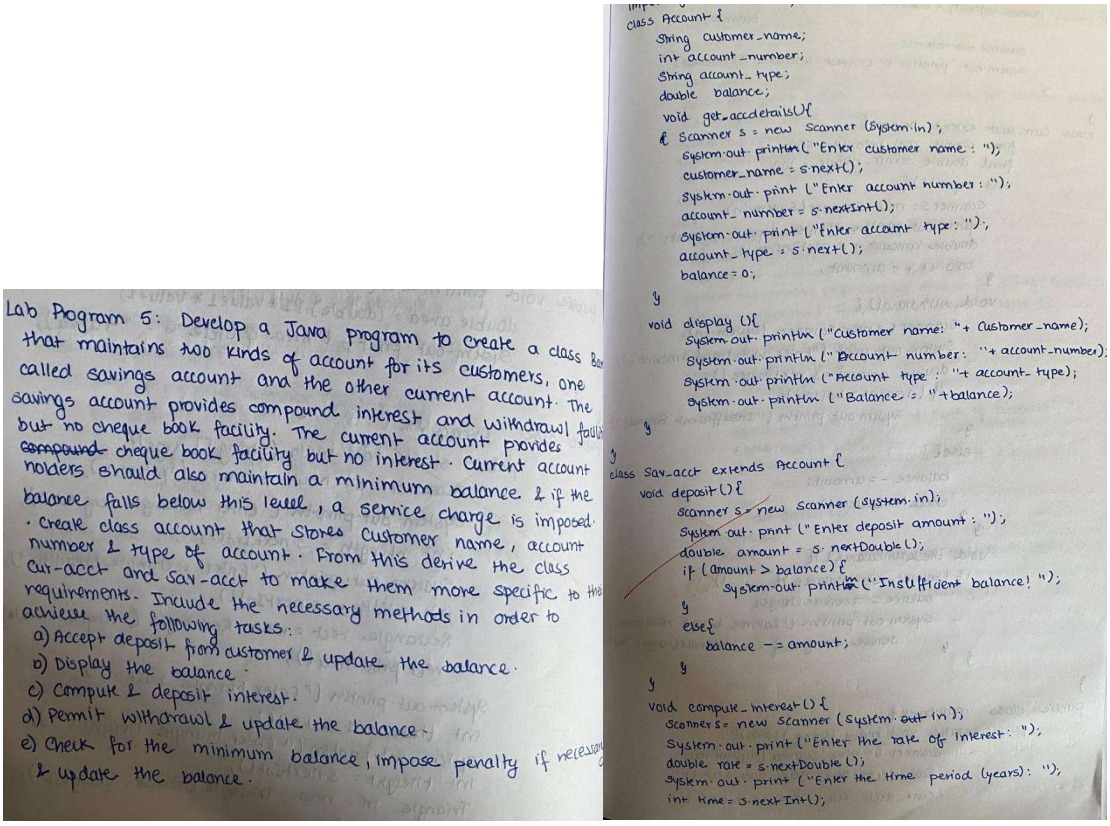
Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book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

a) Accept deposit from customer and update the balance.

b) Display the balance.

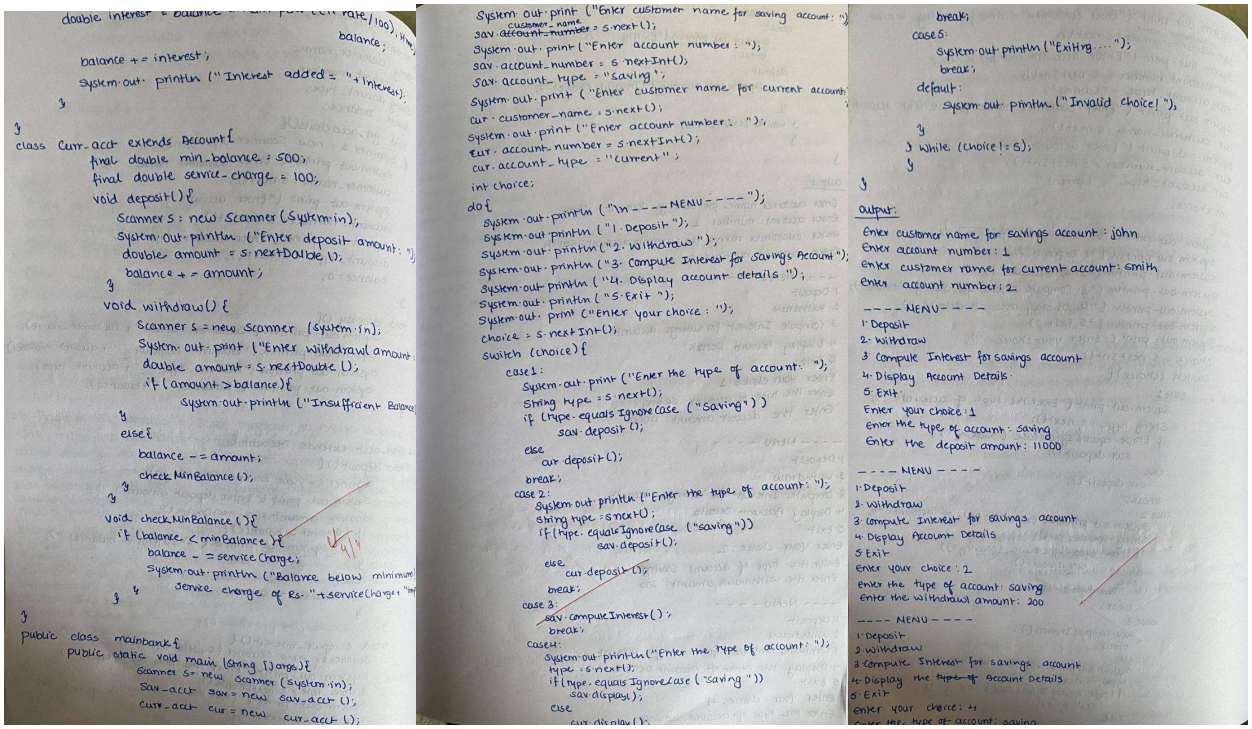
c) Compute and deposit interest

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



Lab Program 5: Develop a Java program to create a class Ba
that maintains two kinds of account for its customers, one
called savings account and the other current account. The
savings account provides compound interest and withdrawal facili
but no cheque book facility. The current account provides
compound cheque book facility but no interest. Current account
holders should also maintain a minimum balance & if the
balance falls below this level, a service charge is imposed.
Create class account that stores customer name, account
number & type of account. From this derive the class
Cur-acct and Sav-acct to make them more specific to the
requirements. Include the necessary methods in order to
achieve the following tasks:
a) Accept deposit from customer & update the balance.
b) Display the balance.
c) Compute & deposit interest.
d) Permit withdrawal & update the balance.
e) Check for the minimum balance, impose penalty if necessary
& update the balance.

```
class Account {
    String customer_name;
    int account_number;
    String account_type;
    double balance;
    void get_accdetails(){
    Scanner s = new Scanner (System.in);
        System.out.println("Enter customer name : ");
        customer_name = s.next();
        System.out.print ("Enter account number : ");
        account_number = s.nextInt();
        System.out.print ("Enter account type : ");
        account_type = s.next();
        balance = 0;
    }
    void display (){
        System.out.println ("Customer name: "+ customer_name);
        System.out.println ("Account number: "+ account_number);
        System.out.println ("Account type : "+ account_type);
        System.out.println ("Balance = "+balance);
    }
}
class Sav_acct extends Account {
    void deposit(){
        Scanner s = new Scanner (System.in);
        System.out.print ("Enter deposit amount : ");
        double amount = s.nextDouble();
        if (amount > balance){
            System.out.println("Insufficient balance! ");
        }
        else{
            balance -= amount;
        }
    }
    Void compute_interest(){
        Scanner s= new Scanner (System.out.in);
        System.out.print ("Enter the rate of interest: ");
        double rate = s.nextDouble();
        System.out.print ("Enter the time period (years): ");
        int time = s.nextInt();
```



```
        double interest = balance * rate/100;
                                                balance;

        balance += interest;
        System.out.println("Interest added = "+interest);
    }
}
class Curr_acct extends Account{
        final double min_balance = 500;
        final double service_charge = 100;
        void deposit(){
            Scanner s = new Scanner(System.in);
            System.out.println("Enter deposit amount: ");
            double amount = s.nextDouble();
            balance += amount;
        }
        void withdraw() {
            Scanner s = new Scanner (System.in);
            System.out.print("Enter withdrawal amount:
            double amount = s.nextDouble();
            if (amount > balance){
                System.out.println("Insufficient Balance
            }
            else{
                balance -= amount;
                checkMinBalance();
            }
        }
        void checkMinBalance(){
            if (balance < minBalance){
                balance -= serviceCharge;
                System.out.println("Balance below minimum
                    service charge of Rs. "+serviceCharge+"
            }
}
public class mainbank{
    public static void main(String[]args){
        Scanner s= new Scanner(System.in);
        Sav_acct sav = new Sav_acct();
        Curr_acct cur = new Curr_acct();
```

```
System.out.print("Enter customer name for saving account: ");
sav.account_number = s.next();
System.out.print("Enter account number: ");
sav.account_number = s.nextInt();
sav.account_type = "saving";
System.out.print("Enter customer name for current account
cur.customer_name = s.next();
System.out.print("Enter account number: ");
cur.account_number = s.nextInt();
cur.account_type = "current";

int choice;

do{
    System.out.println("\n----MENU----");
    System.out.println("1. Deposit");
    System.out.println("2. Withdraw");
    System.out.println("3. Compute Interest for Savings Account");
    System.out.println("4. Display account details");
    System.out.println("5. Exit");
    System.out.print("Enter your choice: ");
    choice = s.nextInt();
    switch (choice){
        case1:
            System.out.print("Enter the type of account: ");
            String type = s.next();
            if (type.equalsIgnoreCase("Saving"))
                sav.deposit();
            else
                cur.deposit();
            break;
        case 2:
            System.out.println("Enter the type of account: ");
            String type = s.next();
            if(type.equalsIgnoreCase("saving"))
                sav.deposit();
            else
                cur.deposit();
            break;
        case 3:
            sav.computeInterest();
            break;
        case4:
            System.out.println("Enter the type of account: ");
            type = s.next();
            if(type.equalsIgnoreCase("saving"))
                sav.display();
            else
                cur.display();
            break;
        case 5:
            System.out.println("Exiting....");
            break;
        default:
            System.out.println("Invalid choice!");
    }
} while (choice!=5);
}
```

output:

```
Enter customer name for savings account : john
Enter account number : 1
Enter customer name for current account: smith
Enter account number: 2

----MENU----
1. Deposit
2. Withdraw
3. Compute Interest for savings account
4. Display Account Details
5. Exit
Enter your choice: 1
Enter the type of account: saving
Enter the deposit amount: 11000

----MENU----
1. Deposit
2. Withdraw
3. Compute Interest for savings account
4. Display Account Details
5. Exit
Enter your choice: 2
enter the type of account: saving
enter the withdrawal amount: 200

----MENU----
1. Deposit
2. Withdraw
3. Compute Interest for savings account
4. Display the type of Account Details
5. Exit
Enter your choice: 4
Enter the type of account: saving
Account number: 1
Type of account: saving
Balance: 10800.0

----MENU----
1. Deposit
2. Withdraw
3. Compute Interest for savings account
4. Display Account Details
5. Exit
Enter your choice: 1
Enter type of account: current
enter deposit amount: 2305

----MENU----
1. Deposit
2. Withdraw
3. Compute Interest for Savings account
4. Display Account Details
5. Exit
enter your choice 5
exiting....
```

Code:

```
import java.util.Scanner;
class Account {
   String customerName;
   int accountNumber;
   String accountType;
   double balance;
   void getAccountDetails()
   {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter customer name: ");
      customerName = s.next();
      System.out.print("Enter account Number: ");
      accountNumber = s.nextInt();
      System.out.print("Enter type of account (saving/current): ");
      accountType = s.next();
      balance = 0;
   }
   void display() {
      System.out.println("Customer name: " + customerName);
      System.out.println("Account number: " + accountNumber);
      System.out.println("Type of Account: " + accountType);
      System.out.println("Balance = " + balance);
   }
}
class Sav_acct extends Account {
   void deposit()
   {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter the deposit amount: ");
      double amount = s.nextDouble();
      balance += amount;
   }
   void withdraw()
   {
      Scanner s = new Scanner(System.in);
      System.out.print("Enter the withdrawal amount: ");
      double amount = s.nextDouble();
      if (amount > balance) {
         System.out.println("Insufficient balance!");
      } else {
```

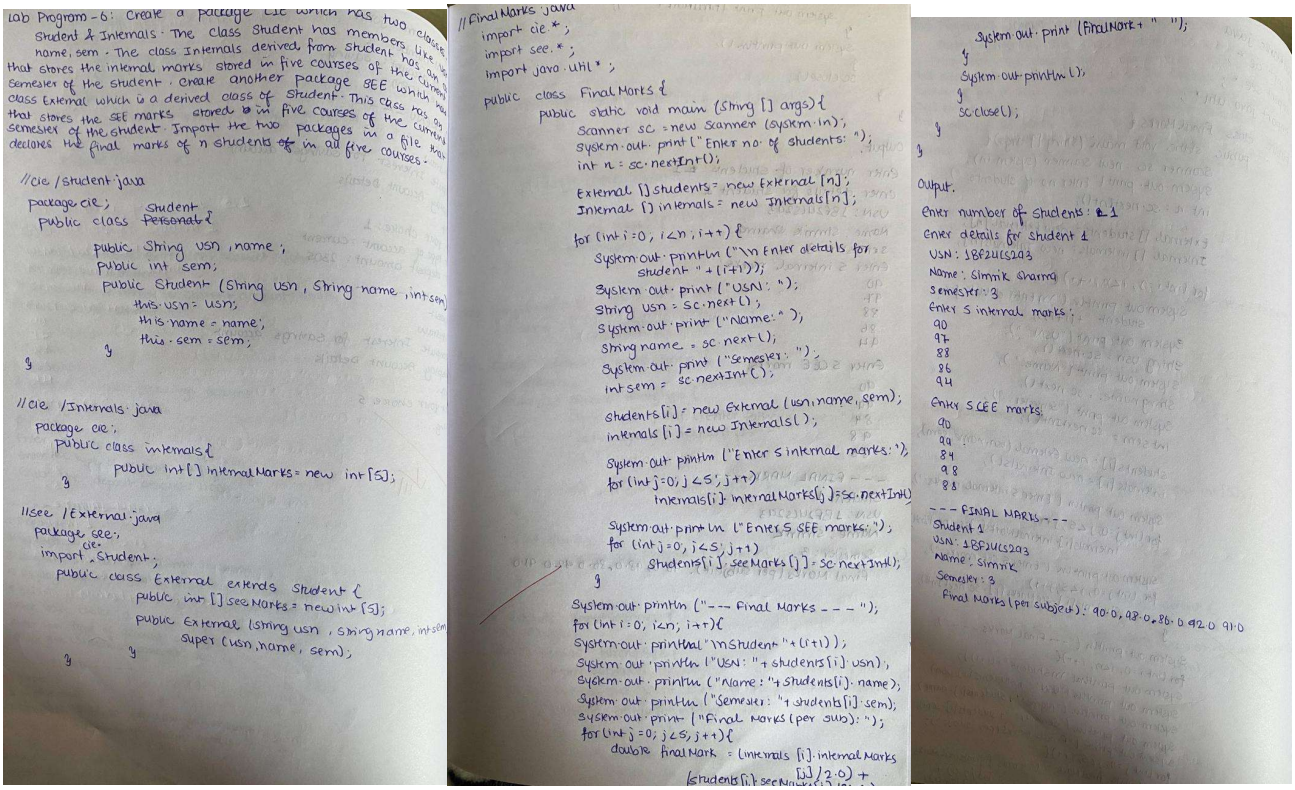
```
            balance -= amount;
        }
    }
    void computeInterest()
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the rate of interest: ");
        double rate = s.nextDouble();
        System.out.print("Enter the time period (years): ");
        int time = s.nextInt();
        double interest = balance * Math.pow((1 + rate / 100), time) - balance;
        balance += interest;
        System.out.println("Interest added = " + interest);
    }
}
class Cur_acct extends Account {
    final double minBalance = 500;  // minimum required balance
    final double serviceCharge = 100;
    void deposit()
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the deposit amount: ");
        double amount = s.nextDouble();
        balance += amount;
    }
    void withdraw()
    {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the withdrawal amount: ");
        double amount = s.nextDouble();
        if (amount > balance) {
            System.out.println("Insufficient balance!");
        } else {
            balance -= amount;
            checkMinBalance();
        }
    }
    void checkMinBalance()
    {
        if (balance < minBalance) {
            balance -= serviceCharge;
```

```
            System.out.println("Balance below minimum! Service charge of Rs." + serviceCharge + "
imposed.");
        }
    }
}
public class maibank {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        Sav_acct sav = new Sav_acct();
        Cur_acct cur = new Cur_acct();
        System.out.print("Enter customer name for savings account: ");
        sav.customerName = s.next();
        System.out.print("Enter account Number: ");
        sav.accountNumber = s.nextInt();
        sav.accountType = "saving";
        System.out.print("Enter customer name for current account: ");
        cur.customerName = s.next();
        System.out.print("Enter account Number: ");
        cur.accountNumber = s.nextInt();
        cur.accountType = "current";
        int choice;
        do {
            System.out.println("\n-----MENU-----");
            System.out.println("1. Deposit");
            System.out.println("2. Withdraw");
            System.out.println("3. Compute interest for SavingsAccount");
            System.out.println("4. Display account details");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = s.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter the type of account: ");
                    String type = s.next();
                    if (type.equalsIgnoreCase("saving"))
                        sav.deposit();
                    else
                        cur.deposit();
                    break;
                case 2:
                    System.out.print("Enter the type of account: ");
```

```
                type = s.next();
                if (type.equalsIgnoreCase("saving"))
                    sav.withdraw();
                else
                    cur.withdraw();
                break;
            case 3:
                sav.computeInterest();
                break;
            case 4:
                System.out.print("Enter the type of account: ");
                type = s.next();
                if (type.equalsIgnoreCase("saving"))
                    sav.display();
                else
                    cur.display();
                break;
            case 5:
                System.out.println("Exiting...");
                break;
            default:
                System.out.println("Invalid choice!");
        }
    } while (choice != 5);
  }
}
```

Program 6

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



```
Lab Program - 6: Create a package CIE which has two classes
Student & Internals: The class Student has members like
name, sem. The class Internals derived from Student has an
that stores the internal marks stored in five courses of the current
semester of the student. Create another package SEE which
class External which is a derived class of Student. This class has an
that stores the SEE marks stored 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.

//cie/Student.java
package cie;
public class Personal{
    public String usn, name;
    public int sem;
    public Student(String usn, String name, int sem){
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}

//cie/Internals.java
package cie;
public class Internals{
    public int[] internalMarks = new int[5];
}

//see/External.java
package see;
import cie.Student;
public class External extends Student{
    public int[] seeMarks = new int[5];
    public External(String usn, String name, int sem){
        super(usn, name, sem);
    }
}
```

```
//FinalMarks.java
import cie.*;
import see.*;
import java.util.*;
public class FinalMarks{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter no. of students: ");
        int n = sc.nextInt();
        External[] students = new External[n];
        Internals[] internals = new Internals[n];
        for(int i=0; i<n; i++){
            System.out.println("\n Enter details for
                student "+(i+1));
            System.out.print("USN: ");
            String usn = sc.next();
            System.out.print("Name: ");
            String name = sc.next();
            System.out.print("Semester: ");
            int sem = sc.nextInt();
            students[i] = new External(usn,name,sem);
            internals[i] = new Internals();
            System.out.println("Enter 5 internal marks:");
            for(int j=0; j<5; j++)
                internals[i].internalMarks[j]=sc.nextInt();
            System.out.println("Enter 5 SEE marks:");
            for(int j=0; j<5; j++)
                students[i].seeMarks[j] = sc.nextInt();
        }
        System.out.println("--- Final Marks ---");
        for(int i=0; i<n; i++){
            System.out.println("\nStudent "+(i+1));
            System.out.println("USN: "+students[i].usn);
            System.out.println("Name: "+students[i].name);
            System.out.println("Semester: "+students[i].sem);
            System.out.print("Final Marks (per sub): ");
            for(int j=0; j<5; j++){
                double finalMark = (internals[i].internalMarks
                    [j]/2.0) +
                    (students[i].seeMarks[j]/2.0);
                System.out.print(finalMark+" ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

```
Output:

Enter number of students: 1
Enter details for student 1
USN: 1BI21CS203
Name: Simrik Sharma
Semester: 3
Enter 5 internal marks:
90
97
88
86
94
Enter 5 SEE marks:
90
99
84
98
88
--- FINAL MARKS ---
Student 1
USN: 1BI21CS203
Name: Simrik
Semester: 3
Final Marks (per subject): 90.0, 98.0, 86.0, 92.0, 91.0
```

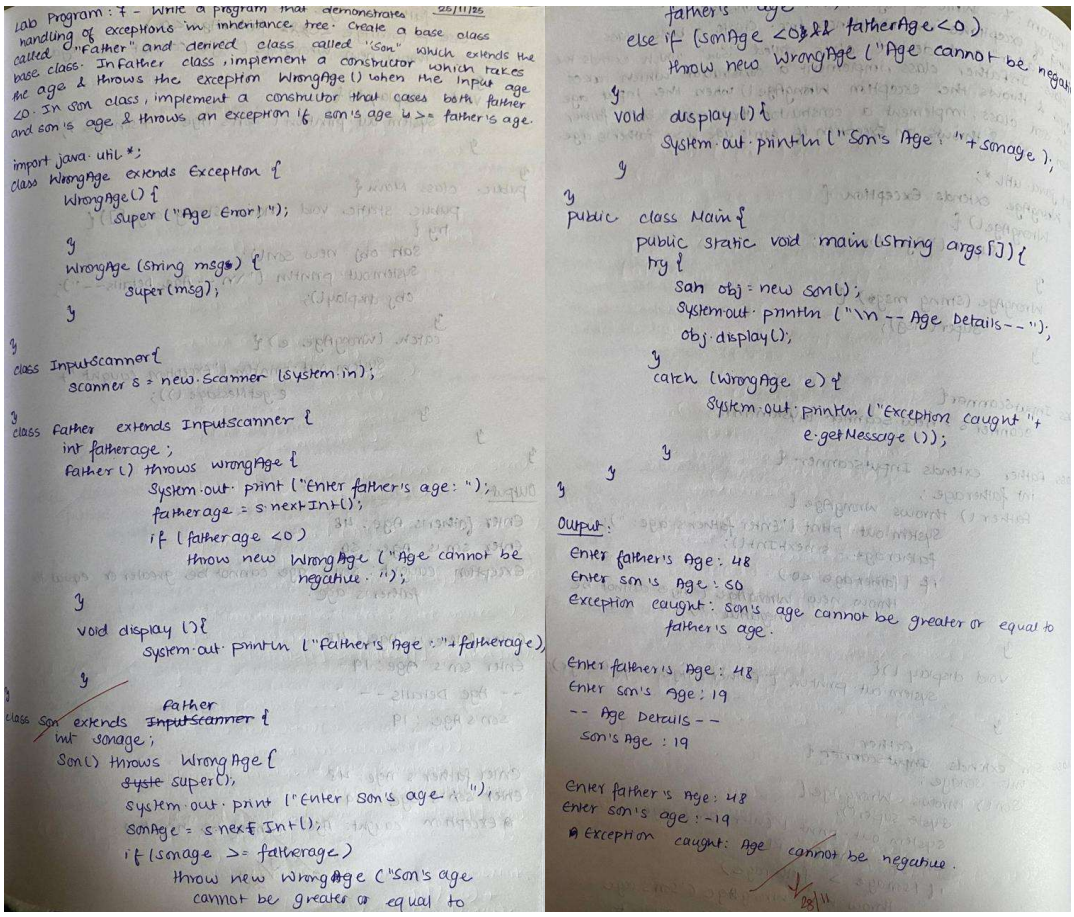
Code:

```
// cie/Personal.java
package cie;
public class Personal {
    public String usn, name;
    public int sem;
    public Personal(String usn, String name, int sem) {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
// cie/Internals.java
package cie;
public class Internals {
    public int[] internalMarks = new int[5];
}
// see/External.java
package see;
import cie.Personal;
public class External extends Personal {
    public int[] seeMarks = new int[5];
    public External(String usn, String name, int sem) {
        super(usn, name, sem);
    }
}
// FinalMarks.java
import cie.*;
import see.*;
import java.util.*;
public class FinalMarks {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int n = sc.nextInt();
        External[] students = new External[n];
        Internals[] internals = new Internals[n];
        for (int i = 0; i < n; i++) {
            System.out.println("\nEnter details for student " + (i + 1));
            System.out.print("USN: ");
            String usn = sc.next();
            System.out.print("Name: ");
            String name = sc.next();
            System.out.print("Semester: ");
            int sem = sc.nextInt();
            students[i] = new External(usn, name, sem);
```

```
            internals[i] = new Internals();
            System.out.println("Enter 5 internal marks:");
            for (int j = 0; j < 5; j++)
                internals[i].internalMarks[j] = sc.nextInt();
            System.out.println("Enter 5 SEE marks:");
            for (int j = 0; j < 5; j++)
                students[i].seeMarks[j] = sc.nextInt();
        }
        System.out.println("\n----- FINAL MARKS -----");
        for (int i = 0; i < n; i++) {
            System.out.println("\nStudent " + (i + 1));
            System.out.println("USN: " + students[i].usn);
            System.out.println("Name: " + students[i].name);
            System.out.println("Semester: " + students[i].sem);
            System.out.print("Final Marks (per subject): ");
            for (int j = 0; j < 5; j++) {
                double finalMark = (internals[i].internalMarks[j] / 2.0)
                        + (students[i].seeMarks[j] / 2.0);
                System.out.print(finalMark + " ");
            }
            System.out.println();
        }
        sc.close();
    }
}
```

Program 7

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



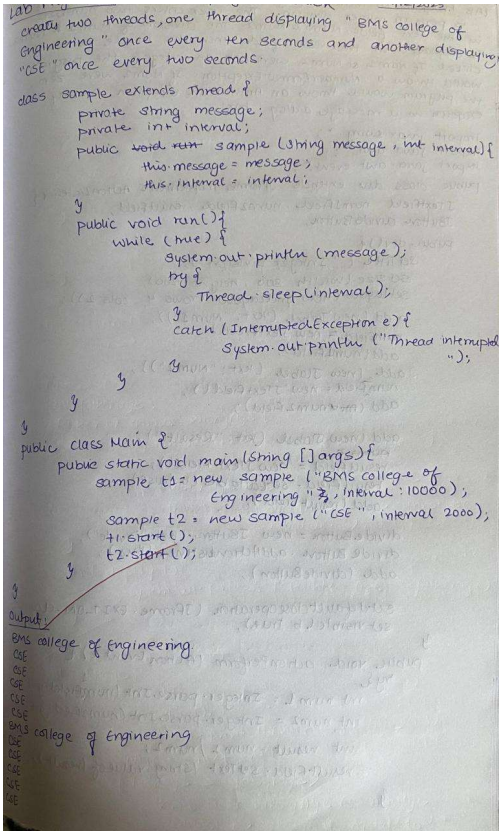
Code:

```
import java.util.*;
class WrongAge extends Exception {
   WrongAge() {
      super("Age Error!");
   }
   WrongAge(String msg) {
      super(msg);
   }
}
class InputScanner {
   Scanner s = new Scanner(System.in);
}
class Father extends InputScanner {
```

```
    int fatherAge;
    Father() throws WrongAge {
        System.out.print("Enter father's age: ");
        fatherAge = s.nextInt();
        if (fatherAge < 0)
            throw new WrongAge("Age cannot be negative");
    }
    void display() {
        System.out.println("Father's age: " + fatherAge);
    }
}
class Son extends Father {
    int sonAge;
    Son() throws WrongAge {
        super();
        System.out.print("Enter son's age: ");
        sonAge = s.nextInt();
        if (sonAge >= fatherAge)
            throw new WrongAge("Son's age cannot be greater than or equal to father's age");
        else if (sonAge < 0 && fatherAge < 0)
            throw new WrongAge("Age cannot be negative");
    }
    void display() {
        System.out.println("Son's age: " + sonAge);
    }
}
public class Main {
    public static void main(String[] args) {
        try {
            Son obj = new Son();
            System.out.println("\n--- Age Details ---");
            obj.display();
        } catch (WrongAge e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
    }
}
```

Program 8

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



```
create two threads, one thread displaying "BMS college of
Engineering" once every ten seconds and another displaying
"CSE" once every two seconds

class sample extends Thread {
    private String message;
    private int interval;
    public void ~~ret~~ sample (String message, int interval){
        this.message = message;
        this.interval = interval;
    }
    public void run(){
        while (true) {
            System.out.println (message);
            try{
                Thread.sleep(interval);
            }
            catch (InterruptedException e){
                System.out.println ("Thread interrupted
                                    ");
            }
        }
    }
}
public class Main {
    public static void main(String []args){
        sample t1= new sample ("BMS college of
                    Engineering", interval:10000);
        sample t2 = new sample ("CSE", interval 2000);
        t1.start();
        t2.start();
    }
}
Output:
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
```

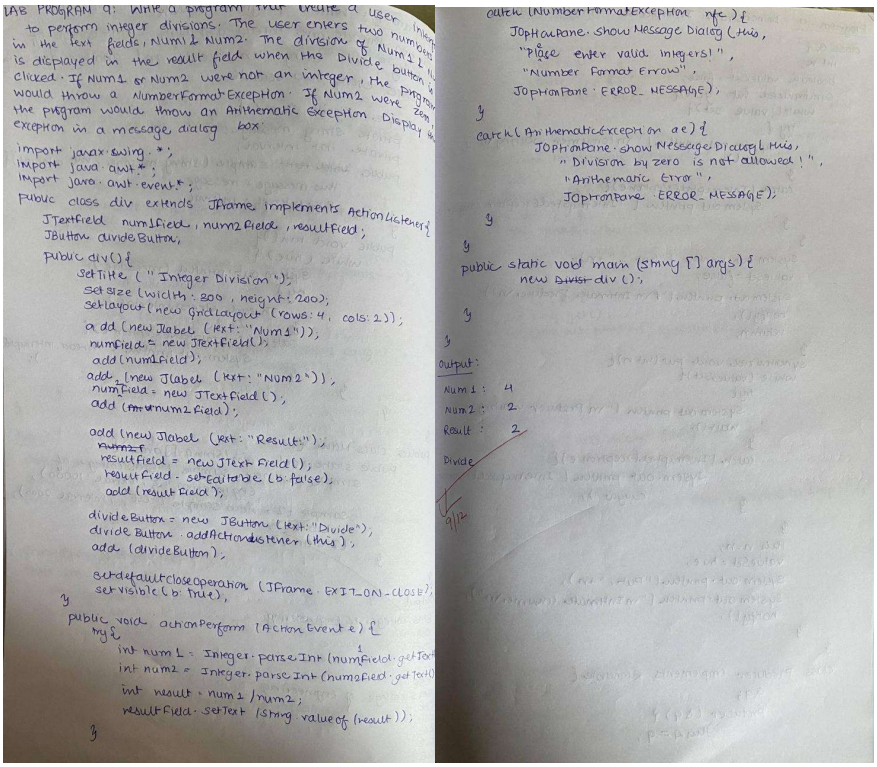
Code:

```
class sample extends Thread {
    private String message;
    private int interval;
    public sample(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }
    public void run() {
        while (true) {
            System.out.println(message);
            try {
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                System.out.println("Thread interrupted");
            }
        }
    }
}
public class Main {
```

```
    public static void main(String[] args) {
        sample t1 = new sample("BMS College of Engineering", 10000);
        sample t2 = new sample("CSE", 2000);
        t1.start();
        t2.start();
    }
}
```

Program 9

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



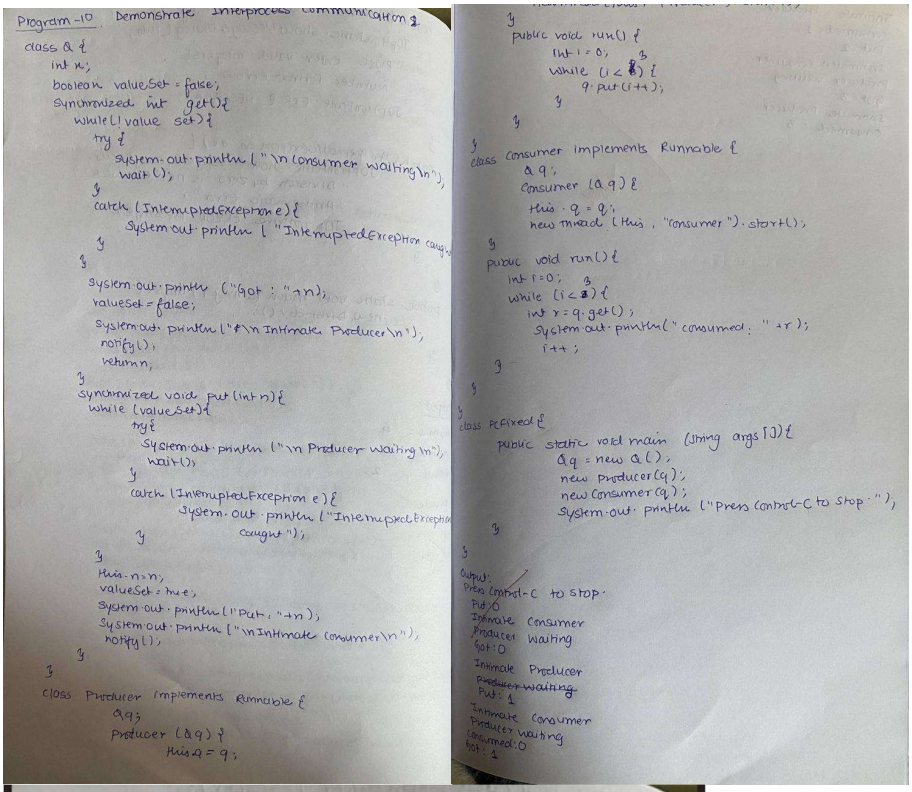
Code:

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

public class DivisionUI extends JFrame implements ActionListener {

    JTextField num1Field, num2Field, resultField;
    JButton divideButton;

    public DivisionUI() {
        setTitle("Integer Division");
        setSize(300, 200);
        setLayout(new GridLayout(4, 2));

        add(new JLabel("Num1:"));
        num1Field = new JTextField();
```

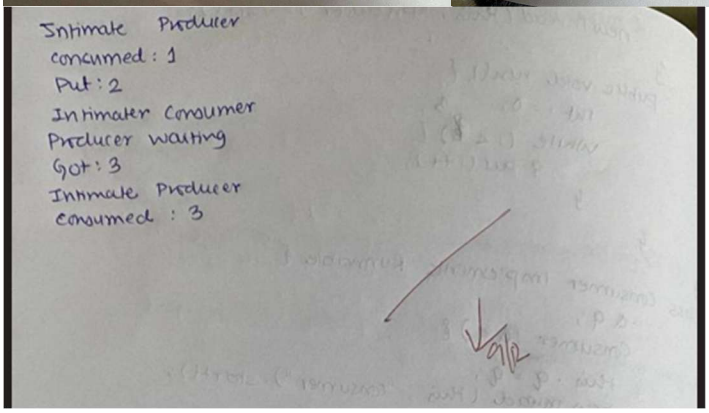
```
        add(num1Field);

        add(new JLabel("Num2:"));
        num2Field = new JTextField();
        add(num2Field);

        add(new JLabel("Result:"));
        resultField = new JTextField();
        resultField.setEditable(false);
        add(resultField);

        divideButton = new JButton("Divide");
        divideButton.addActionListener(this);
        add(divideButton);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

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

            int result = num1 / num2;
            resultField.setText(String.valueOf(result));

        } catch (NumberFormatException nfe) {
            JOptionPane.showMessageDialog(this,
                    "Please enter valid integers!",
                    "Number Format Error",
                    JOptionPane.ERROR_MESSAGE);
        } catch (ArithmeticException ae) {
            JOptionPane.showMessageDialog(this,
                    "Division by zero is not allowed!",
                    "Arithmetic Error",
                    JOptionPane.ERROR_MESSAGE);
        }
    }

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

Program 10

Demonstrate Inter process Communication and deadlock



```
Program -10  Demonstrate Interprocess communication.

class Q {
    int n;
    boolean valueSet = false;
    synchronized int get(){
      while(!value set){
        try {
          System.out.println("\n Consumer waiting\n");
          wait();
        }
        catch (InterruptedException e){
          System.out.println("InterruptedException caught");
        }
      }
      System.out.println("Got : "+n);
      valueSet = false;
      System.out.println("\n Intimate Producer\n");
      notify();
      return n;
    }
    synchronized void put(int n){
      while (valueSet){
          try{
            System.out.println("\n Producer waiting\n");
            wait();
          }
          catch (InterruptedException e){
              System.out.println("InterruptedException
                  caught");
          }
      }
      this.n=n;
      valueSet = true;
      System.out.println("Put: "+n);
      System.out.println("\nIntimate Consumer\n");
      notify();
    }
}

class Producer implements Runnable {
      Q q;
      Producer(Q q){
          this.q = q;

    public void run(){
        int i=0;
        while (i<5){
            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<5){
        int r = q.get();
        System.out.println("consumed: "+r);
        i++;
      }
    }
}
class PCFixed{
    public static void main(String args[]){
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}

Output:
Press Control-C to stop.
Put:0
Intimate Consumer
Producer waiting
Got:0
Intimate Producer
Producer waiting
Put: 1
Intimate Consumer
Producer waiting
Consumed:0
Got: 1
```



```
Intimate Producer
consumed : 1
Put: 2
Intimater Consumer
Producer waiting
Got: 3
Intimate Producer
consumed : 3
```

Code:

```
class Q {
   int n;
   boolean valueSet = false;
   synchronized int get() {
      while (!valueSet) {
         try {
            System.out.println("\nConsumer waiting");
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
         }
      }
      System.out.println("Got: " + n);
      valueSet = false;
      System.out.println("\nIntimate Producer");
      notify();
      return n;
   }
   synchronized void put(int n) {
      while (valueSet) {
         try {
            System.out.println("\nProducer waiting");
            wait();
         } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
         }
      }
      this.n = n;
      valueSet = true;
      System.out.println("Put: " + n);
      System.out.println("\nIntimate Consumer");
      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++;
      }
   }
}
public class PCFixed {
   public static void main(String args[]) {
      Q q = new Q();
      new Producer(q);
      new Consumer(q);
      System.out.println("Press Control-C to stop.");
   }
}
```