VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

Lakshya Khandelwal (1BM23CS166)

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 **Lakshya Khandelwal (1BM23CS166),** 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.

Srushti C S Assistant Professor Department of CSE, BMSCE	Dr. Jyothi S Nayak Professor & HOD Department of CSE, BMSCE

Index

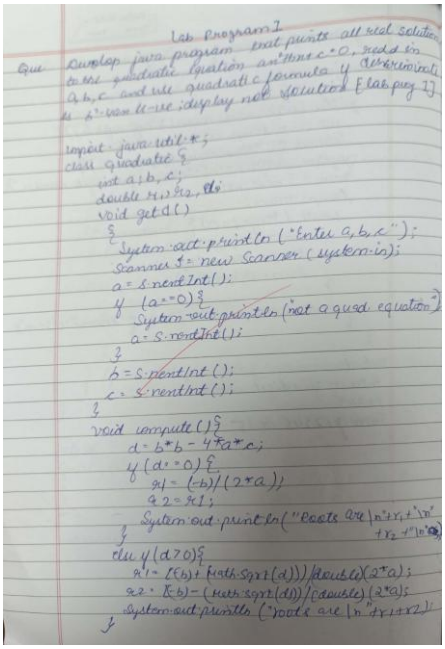
Sl. No.	Date	Experiment Title	Page No.
1	30/09/2024	Implement Quadratic Equation	3
2	08/10/2024	CGPA and SGPA Calculator	6
3	14/10/2024	Book Information Program	9
4	21/10/2024	Area Calculator	11
5	28/10/2024	Bank Interest Rate Calculator	14
6	11/11/2024	Final Marks Of Test Results	19
7	28/10/2024	Father Son Age Program	22
8	28/10/2024	Threads Program	24
9	28/10/2024	Open Ended Exercise	26
10	28/10/2024	InterProcess Communication and Deadlock	28

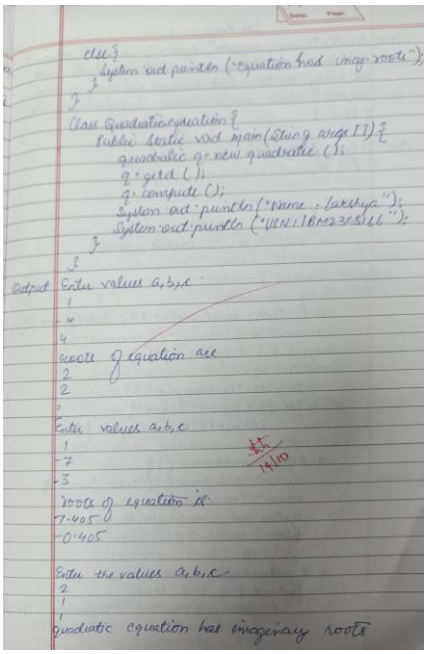
Github Link: https://github.com/lakshya1216/OOJ-Lab-programs/tree/main/1BM23CS166

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.

Algorithm:





Code:

```
import static java.lang.Math.sqrt;
import java.util.Scanner;

class quadratic {
    int a, b, c;
    double r1, r2, d;

    void input() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter value of a: ");
```

```
        a = sc.nextInt();

        while (a == 0) {
            System.out.println("Enter a non-zero number for a:");
            a = sc.nextInt();
        }

        System.out.print("Enter value of b: ");
        b = sc.nextInt();
        System.out.print("Enter value of c: ");
        c = sc.nextInt();

        d = b * b - 4 * a * c;
    }

    void display() {
        if (d == 0) {
            r1 = -b / (2.0 * a);
            System.out.println("Roots are real and equal");
            System.out.println("Root: " + r1);
        } else if (d > 0) {
            r1 = (-b + sqrt(d)) / (2.0 * a);
            r2 = (-b - sqrt(d)) / (2.0 * a);
            System.out.println("Roots are real and different");
            System.out.println("r1 = " + r1 + ", r2 = " + r2);
        } else {
            r1 = -b / (2.0 * a);
            r2 = sqrt(-d) / (2.0 * a);
            System.out.println("Roots are imaginary");
            System.out.println("r1 = " + r1 + " + " + r2 + "i");
            System.out.println("r2 = " + r1 + " - " + r2 + "i");
        }
    }

    public static void main(String[] args) {
        quadratic qe = new quadratic();
        qe.input();
        qe.display();
            System.out.println("Lakshya Khandelwal");
            System.out.println("1BM23cs166");
    }
}
```

```
C:\123>java quadratic
Enter value of a: 0
Enter a non-zero number for a:
1
Enter value of b: 2
Enter value of c: 1
Roots are real and equal
Root: -1.0
Lakshya Khandelwal
1BM23cs166
```

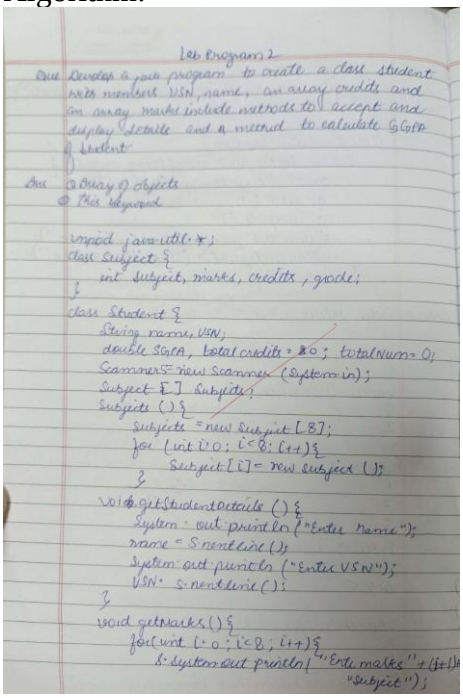
```
C:\123>java quadratic
Enter value of a: 5
Enter value of b: 3
Enter value of c: 8
Roots are imaginary
r1 = -0.3 + 1.22882057274444508i
r2 = -0.3 - 1.22882057274444508i
Lakshya Khandelwal
1BM23cs166
```

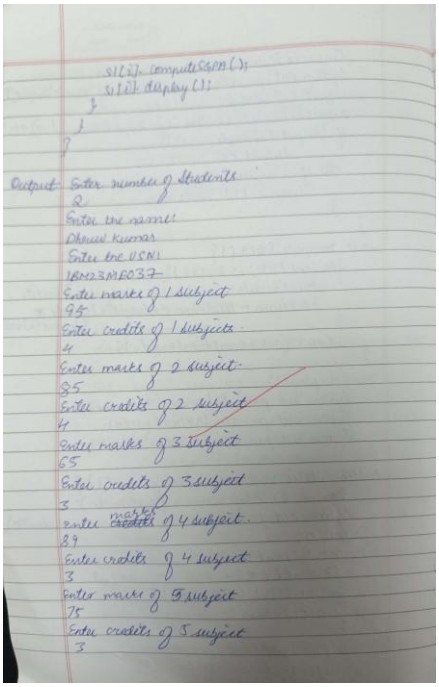
```
C:\123>java quadratic
Enter value of a: 2
Enter value of b: 6
Enter value of c: 3
Roots are real and different
r1 = -0.6339745962155614, r2 = -2.3660254037844384
Lakshya Khandelwal
1BM23cs166
```

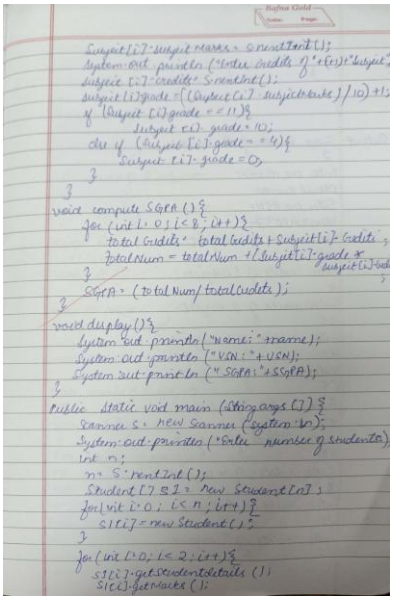
Program 2:

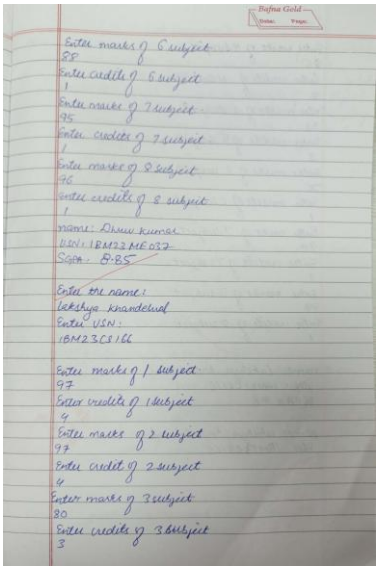
Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

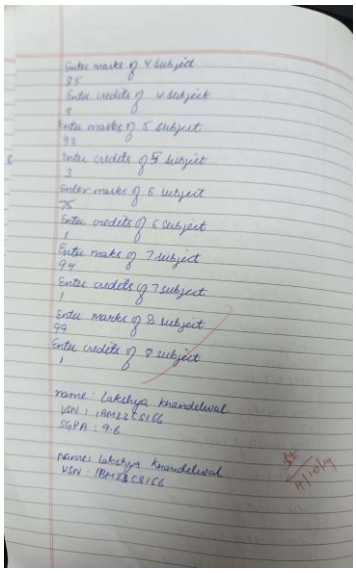
Algorithm:











Code:

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;

    void calGrade() {
        grade = subjectMarks / 10 + 1;
        if (grade == 11) {
            grade = 10;
        } else if (grade < 4) {
            grade = 0;
        }
    }
}

class Student1 {
    String name;
    String usn;
    double SGPA;
    double temp = 0.0;
    int totalCredits = 0;
    Subject[] subject;
    Scanner s;

    Student1() {
        subject = new Subject[8];
```

```
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
        s = new Scanner(System.in);
    }

    void getStudentDetails() {
        System.out.println("Enter student name and USN:");
        name = s.next();
        usn = s.next();
    }

    void getMarks() {
        System.out.println("Enter marks and credits for 8 subjects:");
        for (int i = 0; i < 8; i++) {
            System.out.println("Enter marks and credits of subject " + (i + 1) + ":");
            subject[i].subjectMarks = s.nextInt();
            subject[i].credits = s.nextInt();
            subject[i].calGrade();
        }
    }

    void calSgpa() {
        for (int i = 0; i < 8; i++) {
            totalCredits += subject[i].credits;
        }
        for (int i = 0; i < 8; i++) {
            temp += subject[i].grade * subject[i].credits;
        }
        SGPA = temp / totalCredits;
        System.out.println("SGPA of " + name + " is " + SGPA);
    }
}

public class student {
    public static void main(String[] args) {
        Student1 stud = new Student1();
        stud.getStudentDetails();
        stud.getMarks();
        stud.calSgpa();
        System.out.println("Lakshya Khandelwal");
        System.out.println("USN = 1BM23CS166");
    }
}
```

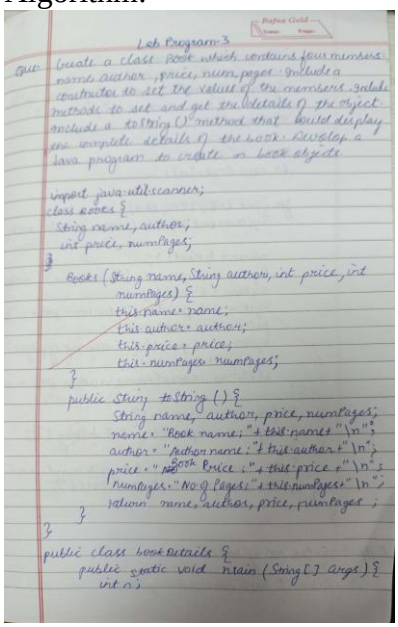
```
Enter student name and USN:
Achintya
1BM23CS001
Enter marks and credits for 8 subjects:
Enter marks and credits of subject 1:
94
4
Enter marks and credits of subject 2:
81
4
Enter marks and credits of subject 3:
89
3
Enter marks and credits of subject 4:
65
3
Enter marks and credits of subject 5:
95
3
Enter marks and credits of subject 6:
89
1
Enter marks and credits of subject 7:
85
1
Enter marks and credits of subject 8:
85
1
SGPA of Achintya is 9.05
Lakshya Khandelwal
USN = 1BM23CS166
```

```
Enter student name and USN:
Lakshya
1BM23CS166
Enter marks and credits for 8 subjects:
Enter marks and credits of subject 1:
97
4
Enter marks and credits of subject 2:
95
4
Enter marks and credits of subject 3:
93
3
Enter marks and credits of subject 4:
80
3
Enter marks and credits of subject 5:
84
3
Enter marks and credits of subject 6:
74
1
Enter marks and credits of subject 7:
91
1
Enter marks and credits of subject 8:
98
1
SGPA of Lakshya is 9.6
Lakshya Khandelwal
USN = 1BM23CS166
```

Program 3:

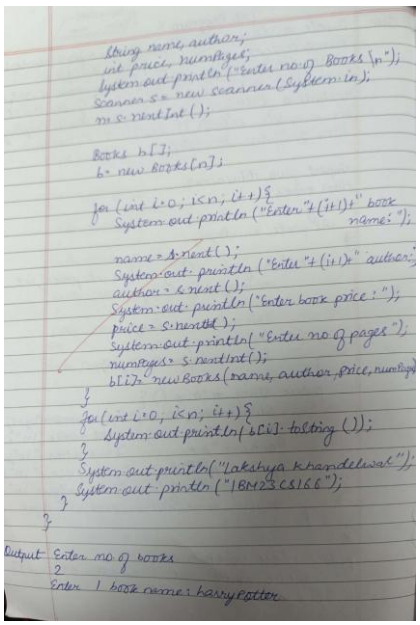
Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program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

Algorithm:

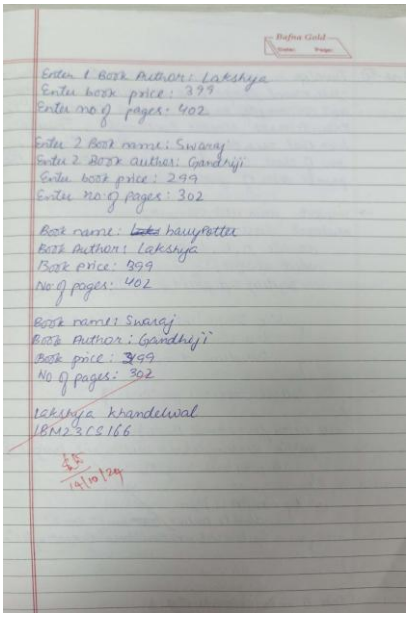

Lab Program 3

Que 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.

```
import java.util.Scanner;
class Books {
String name, author;
int price, numPages;
}
Books (String name, String author, int price, int
    numPages) {
    this.name= name;
    this.author= author;
    this.price= price;
    this.numPages= numPages;
}
public String toString () {
    String name, author, price, numPages;
    name= "Book name: "+ this.name+ "\n";
    author= "Author name: "+ this.author+ "\n";
    price= "Book Price: "+ this.price+ "\n";
    numPages= "No of Pages: "+ this.numPages+ "\n";
    return name, author, price, numPages;
}
public class bookDetails {
    public static void main (String[] args) {
    int n;
```


```
String name, author;
int price, numPages;
System.out.println("Enter no of Books \n");
Scanner s= new Scanner(System.in);
n= s.nextInt();

Books b[];
b= new Books[n];

for (int i=0; i<n; i++) {
    System.out.println("Enter "+(i+1)+" book
                        name: ");

    name= s.next();
    System.out.println("Enter "+(i+1)+" author");
    author= s.next();
    System.out.println("Enter book price: ");
    price= s.nextInt();
    System.out.println("Enter no of pages");
    numPages= s.nextInt();
    b[i]= new Books(name, author, price, numPages);
}
for (int i=0; i<n; i++) {
    System.out.println(b[i].toString());
}
System.out.println("Lakshya Khandelwal");
System.out.println("1BM23CS166");
}
```

Output: Enter no of books
2
Enter 1 book name: harryPotter


```
Enter 1 Book Author: Lakshya
Enter book price: 399
Enter no of pages: 402

Enter 2 Book name: Swaraj
Enter 2 Book Author: Gandhiji
Enter book price: 299
Enter no of pages: 302

Book name: harryPotter
Book Author: Lakshya
Book price: 399
No of pages: 402

Book name: Swaraj
Book Author: Gandhiji
Book price: 299
No of pages: 302

Lakshya Khandelwal
1BM23CS166
```

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="Book price:"+this.price+"\n";
numPages="No. of pages:"+this.numPages+"\n";
return name+author+price+numPages;
}
}
public class bookDetails
{
public static void main(String[] args)
{
int n;
String name;
String author;
int price;
int numPages;
System.out.println("Enter number of books\n");
Scanner s=new Scanner(System.in);
n=s.nextInt();

Books b[];
b=new Books[n];

for(int i=0;i<n;i++){
System.out.print("Enter Books"+(i+1)+" name: ");
name=s.next();
System.out.print("\nEnter Books"+(i+1)+" author: ");
author=s.next();
System.out.print("\nEnter Books"+(i+1)+" price: ");
price=s.nextInt();
System.out.print("\nEnter Books"+(i+1)+" no. of pages: ");
numPages=s.nextInt();
b[i]=new Books(name,author,price,numPages);
}
```

```
for(int i=0;i<n;i++)
{
System.out.println(b[i].toString());
}
System.out.println("Lakshya Khandelwal");
System.out.println("1BM23CS166");
}
}
```

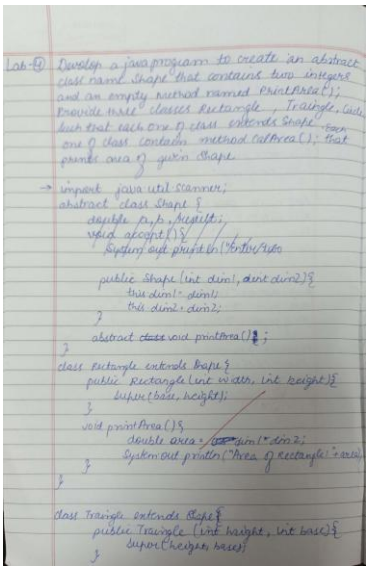
```
Enter number of books

2
Enter Books1 name: harryPotter

Enter Books1 author: Lakshya

Enter Books1 price: 399

Enter Books1 no. of pages: 402
Enter Books2 name: Swaraj

Enter Books2 author: Gandhi

Enter Books2 price: 299

Enter Books2 no. of pages: 302
Book name:harryPotter
Author name:Lakshya
Book price:399
No. of pages:402

Book name:Swaraj
Author name:Gandhi
Book price:299
No. of pages:302

Lakshya Khandelwal
1BM23CS166
```

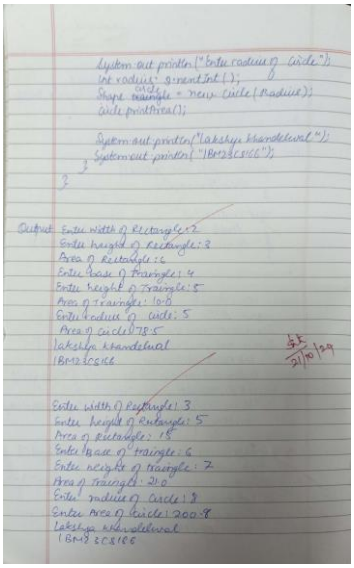
Program 4:

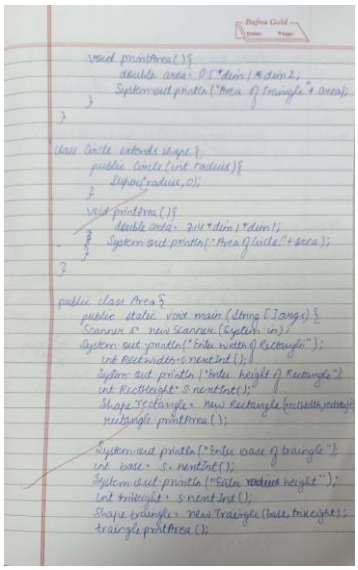
Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea(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

P.T.O.

Algorithm:







Code:

```
import java.util.Scanner;
abstract class Shape {
    int dimension1;
    int dimension2;

    public Shape(int dimension1, int dimension2) {
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }

    abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle(int width, int height) {
        super(width, height);
    }
    void printArea() {
        int area = dimension1 * dimension2;
        System.out.println("Area of Rectangle: " + area);
    }
}

class Triangle extends Shape {
    public Triangle(int base, int height) {
        super(base, height);
    }

    void printArea() {
        double area = 0.5 * dimension1 * dimension2;
        System.out.println("Area of Triangle: " + area);
    }
}
```

```
class Circle extends Shape {
    public Circle(int radius) {
        super(radius, 0);
    }

    void printArea() {
        double area = 3.14 * dimension1 * dimension1;
        System.out.println("Area of Circle: " + area);
    }
}

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

        System.out.print("Enter width of Rectangle: ");
        int rectWidth = scanner.nextInt();
        System.out.print("Enter height of Rectangle: ");
        int rectHeight = scanner.nextInt();
        Shape rectangle = new Rectangle(rectWidth, rectHeight);
        rectangle.printArea();

        System.out.print("Enter base of Triangle: ");
        int triBase = scanner.nextInt();
        System.out.print("Enter height of Triangle: ");
        int triHeight = scanner.nextInt();
        Shape triangle = new Triangle(triBase, triHeight);
        triangle.printArea();

        System.out.print("Enter radius of Circle: ");
        int circleRadius = scanner.nextInt();
        Shape circle = new Circle(circleRadius);
        circle.printArea();

            System.out.println("Lakshya Khandelwal");
            System.out.println("1BM23CS166");

        scanner.close();
    }
}
```

```
Enter width of Rectangle: 2
Enter height of Rectangle: 3
Area of Rectangle: 6
Enter base of Triangle: 4
Enter height of Triangle: 5
Area of Triangle: 10.0
Enter radius of Circle: 5
Area of Circle: 78.5
Lakshya Khandelwal
1BM23CS166
```

```
Enter width of Rectangle: 3
Enter height of Rectangle: 5
Area of Rectangle: 15
Enter base of Triangle: 6
Enter height of Triangle: 7
Area of Triangle: 21.0
Enter radius of Circle: 8
Area of Circle: 200.96
Lakshya Khandelwal
1BM23CS166
```

Program 5:

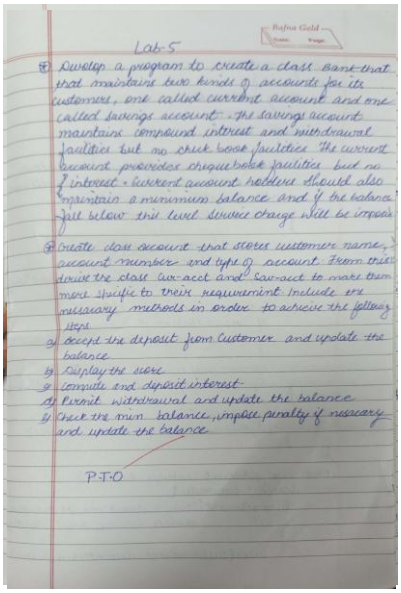
Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book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

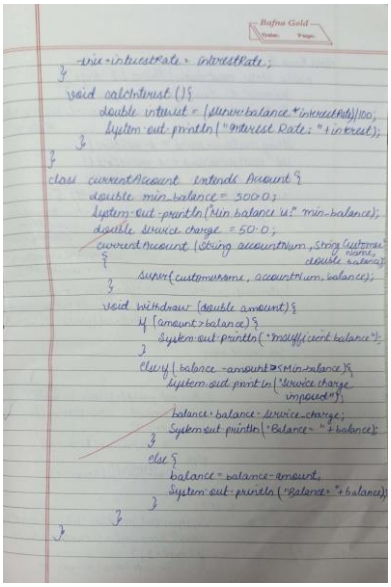
a) Accept deposit from customer and update the balance.

b) Display the balance.

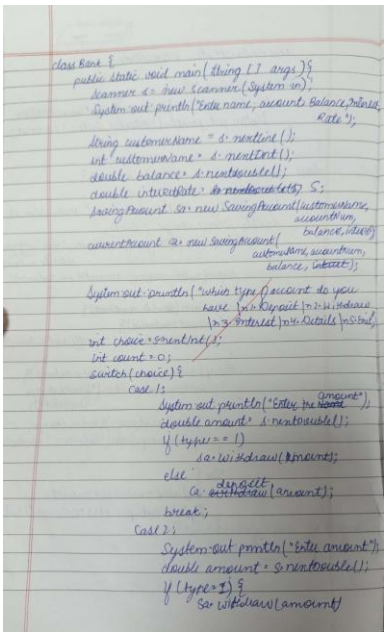
c) Compute and deposit interest

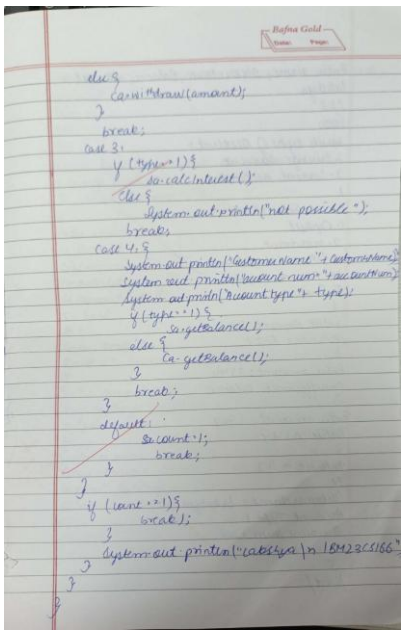
d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

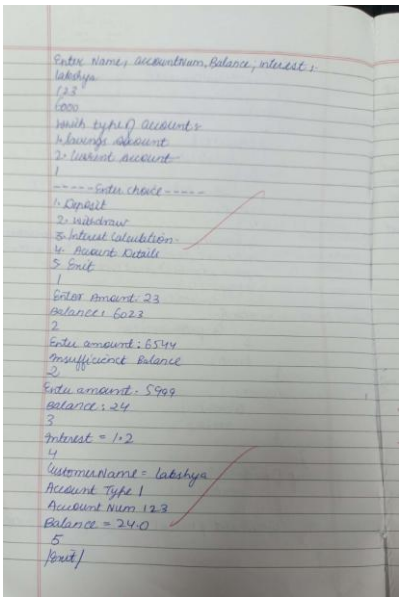


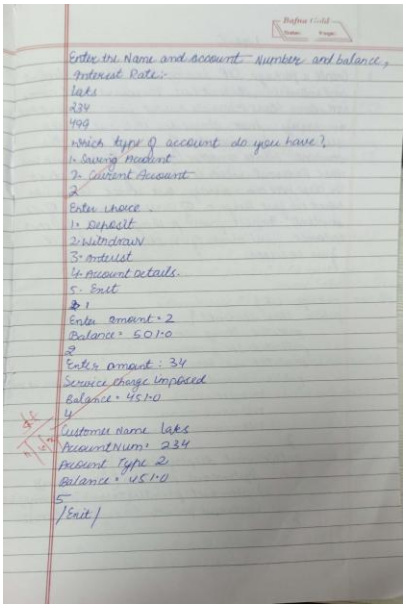


P.T.O.









Code:

```
import java.util.*;
class Account
{
	String customerName;
	int accountNum;
	double balance;

	Account(String customerName,int accountNum,double balance)
	{
		this.customerName=customerName;
```

```
		this.accountNum=accountNum;
		this.balance=balance;
	}

	void deposit(double amount)
	{
		balance=balance+amount;
		System.out.println("Balance after Deposit = "+balance);
	}

	void withdraw(double amount)
	{
		if(amount > balance)
			System.out.println("Insufficient Balance in Account");
		else
		{
			balance=balance-amount;
			System.out.println("Balance after Withdraw = "+balance);
		}
	}

	void getBalance()
	{
		System.out.println("Balance = "+balance);
	}
}

class savingAccount extends Account
{
	double interestRate;

	savingAccount(String customerName, int accountNumber, double balance, double interestRate)
	{
	super(customerName, accountNumber, balance);
	this.interestRate = interestRate;
	}

	void calcInterest()
	{
		double interest = (super.balance*interestRate)/100;
		System.out.println("Interest is "+interest);
	}
}

class currentAccount extends Account
{
	double MIN_BALANCE = 500.0;
	double SERVICE_CHARGE = 50.0;

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

	void withdraw(double amount)
	{
		if(amount > balance)
```

```
                        System.out.println("Insufficient Balance in Account");
                else if(balance-amount<MIN_BALANCE)
                {
                        System.out.println("Service charge will be imposed");
                        balance=balance-SERVICE_CHARGE;
                        System.out.println("Balance after Service charge imposed = "+balance);
                }
                else
                {
                        balance=balance-amount;
                        System.out.println("Balance after Withdraw = "+balance);
                }
        }
}

class Bank
{
        public static void main(String args[])
        {
                Scanner s=new Scanner(System.in);
                System.out.println("Enter the Name and Account Number And Balance, Interest Rate");
                String customerName=s.nextLine();
                int accountNum=s.nextInt();
                double balance=s.nextDouble();
                double interestRate=5;
                savingAccount sa=new savingAccount(customerName,accountNum,balance,interestRate);
                currentAccount ca=new currentAccount(customerName,accountNum,balance);
                System.out.println("Which type of account do you have?\n 1.Saving Account\n 2.Current Account");
                int type=s.nextInt();
                while(true)
                {
                        System.out.println("-----------Enter your choice----------\n1.Deposit\n2.withdraw\n3.Interest
Calculation\n4.Account Details\n5.Exit");
                        int choice=s.nextInt();
                        int count=0;
                        switch(choice)
                        {
                                case 1:
                                        {
                                                System.out.println("Enter the amount : ");
                                                double amount=s.nextDouble();
                                                if(type==1)
                                                        sa.deposit(amount);
                                                else
                                                        ca.deposit(amount);
                                                break;
                                        }
                                case 2:
                                        {
                                                System.out.println("Enter the amount : ");
                                                double amount=s.nextDouble();
                                                if(type==1)
                                                        sa.withdraw(amount);
                                                else
                                                        ca.withdraw(amount);
                                                break;
                                        }
```

```
                        case 3:
                                {
                                        if(type==1)
                                                sa.calcInterest();
                                        else
                                                System.out.println("not possible for current account");
                                        break;
                                }
                        case 4:
                                {
                                        System.out.println("Customer Name "+customerName);
                                        System.out.println("Account Number "+accountNum);
                                        System.out.println("Account Type "+type);
                                        if(type==1)
                                                sa.getBalance();
                                        else
                                                ca.getBalance();
                                        break;
                                }
                        default:System.out.println("Invalid choice");
                                count=1;
                                break;

                }
                if(count==1)
                        break;
                System.out.println("Lakshya Khandelwal\n1BM23CS166");
        }
    }
}
```

```
Enter the Name and Account Number And Balance, Interest Rate
Lakshya
123
6000
Which type of account do you have?
 1.Saving Account
 2.Current Account
1
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
1
Enter the amount :
23
Balance after Deposit = 6023.0
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
2
Enter the amount :
6544

Insufficient Balance in Account
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
2
Enter the amount :
6234
Insufficient Balance in Account
```

```
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
2
Enter the amount :
5999
Balance after Withdraw = 24.0
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
3
Interest is 1.2
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
4
Customer Name Lakshya
Account Number 123
Account Type 1
Balance = 24.0
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
5
Invalid choice
```

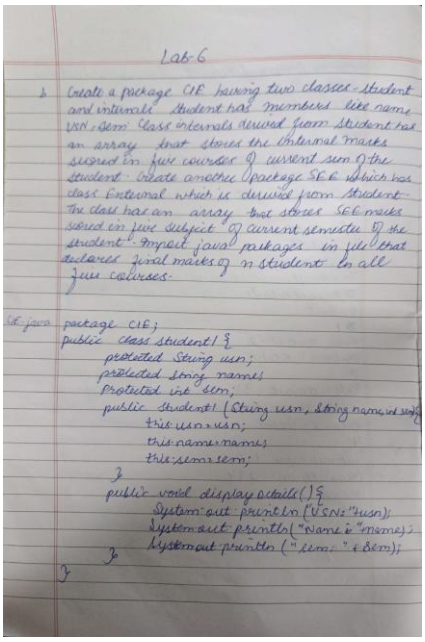
```
Enter the Name and Account Number And Balance, Interest Rate
Laks
234
499
Which type of account do you have?
 1.Saving Account
 2.Current Account
2
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
1
Enter the amount :
2
Balance after Deposit = 501.0
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
2
Enter the amount :
34
Service charge will be imposed
Balance after Service charge imposed = 451.0
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
3
not possible for current account
```

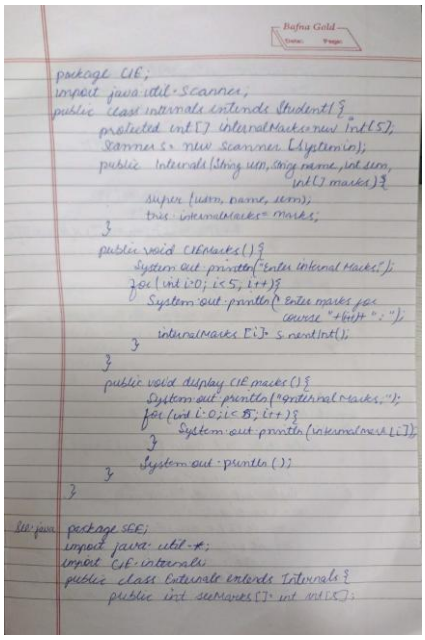
```
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
4
Customer Name Laks
Account Number 234
Account Type 2
Balance = 451.0
----------Enter your choice----------
1.Deposit
2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
5
Invalid choice
```

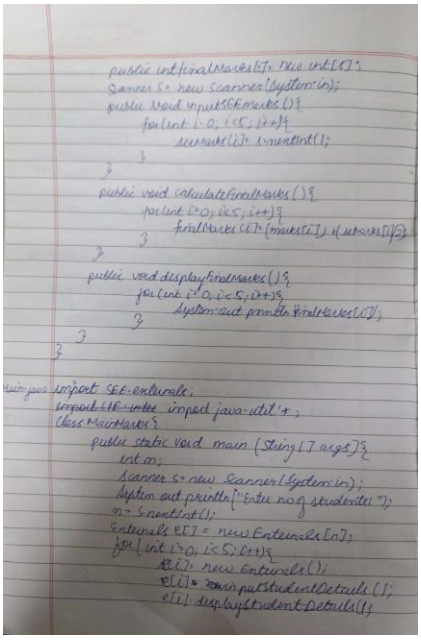
Program 6:

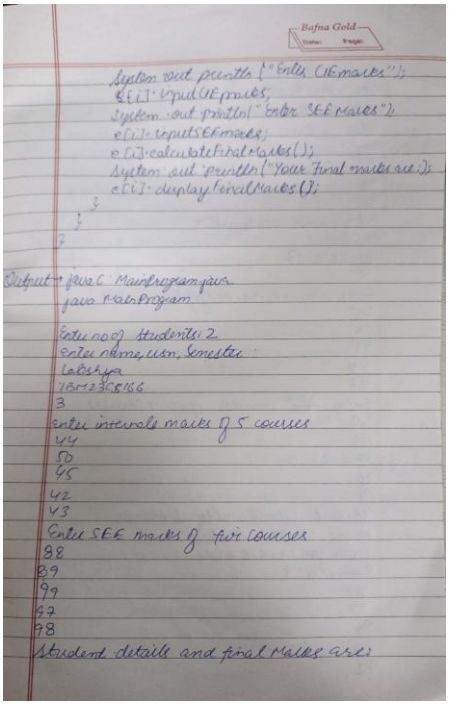
Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External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

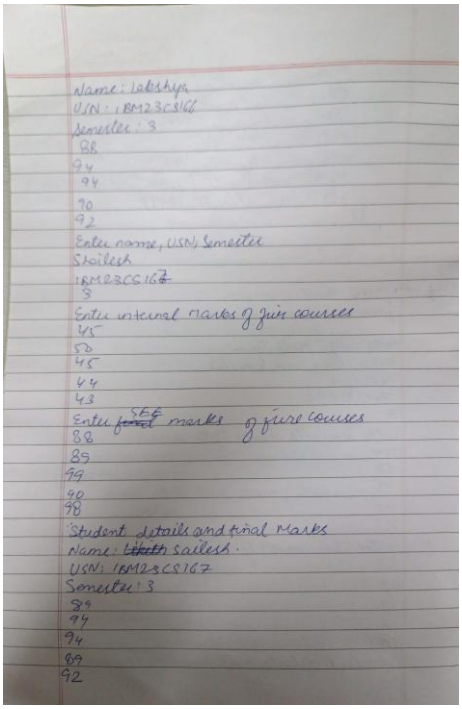
Algorithm:











Code:

```
package CIE;
import java.util.*;
class Student
{
	public String usn;
	public String name;
	public int sem;
	public void inputStudentDetails()
	{
		Scanner s=new Scanner(System.in);
		System.out.println("Enter Name,USN,Semester");
		usn=s.next();
		name=s.next();
		sem=s.nextInt();
	}
	public void displayStudentDetails()
	{
		System.out.println("USN:"+usn);
		System.out.println("Name"+name);
		System.out.println("Semester:"+sem);
	}
}
public class Internals extends Student
{
	public int marks[]=new int[5];
	public void inputCIEMarks()
	{
		Scanner s=new Scanner(System.in);
```

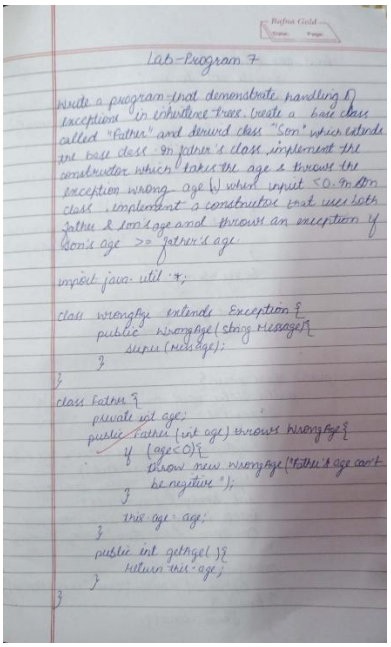
```
                for(int i=0;i<5;i++)
                        marks[i]=s.nextInt();
        }
}


package SEE;
import java.util.*;
import CIE.Internals;
public class Externals extends Internals
{
        public int seemarks[]=new int[5];
        public int finalmarks[]=new int[5];
        Scanner s=new Scanner(System.in);
        public void inputSEEmarks()
        {
                for(int i=0;i<5;i++)
                        seemarks[i]=s.nextInt();
        }
        public void calculateFinalMarks()
        {
                for(int i=0;i<5;i++)
                        finalmarks[i]=(marks[i])+(seemarks[i]/2);
        }
        public void displayFinalMarks()
        {
                for(int i=0;i<5;i++)
                        System.out.println(finalmarks[i]);
        }
}

package SEE;
import java.util.*;
import CIE.Internals;
public class Externals extends Internals
{
        public int seemarks[]=new int[5];
        public int finalmarks[]=new int[5];
        Scanner s=new Scanner(System.in);
        public void inputSEEmarks()
        {
                for(int i=0;i<5;i++)
                        seemarks[i]=s.nextInt();
        }
        public void calculateFinalMarks()
        {
                for(int i=0;i<5;i++)
                        finalmarks[i]=(marks[i])+(seemarks[i]/2);
        }
        public void displayFinalMarks()
        {
                for(int i=0;i<5;i++)
                        System.out.println(finalmarks[i]);
        }
}
```

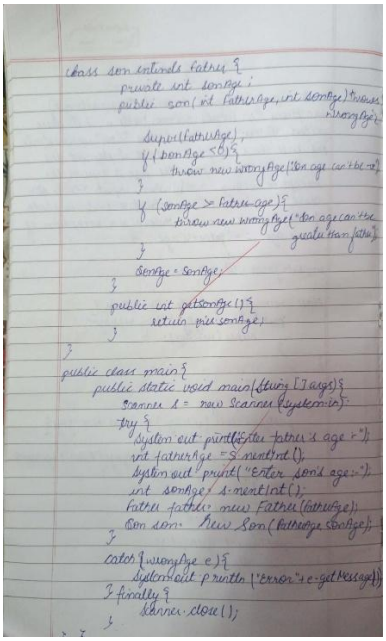
```
D:\1BM23CS172(java)>javac MainProgram.java

D:\1BM23CS172(java)>java MainProgram
Enter the Number of students
2
Enter Name,USN,Semester
shailesh
1BM23CS172
3
Enter the Internals marks of five Courses
44
50
45
42
43
Enter the SEE marks of five Courses
88
89
99
97
99
Student Details and Final Marks
Name:1BM23CS172
USN:shailesh
Semester:3
88
94
94
90
92
Enter Name,USN,Semester
Likith
1BM23CS170
3
Enter the Internals marks of five Courses
45
50
45
44
43
Enter the SEE marks of five Courses
88
89
99
90
98
Student Details and Final Marks
Name:1BM23CS170
USN:Likith
Semester:3
89
94
94
89
92
```

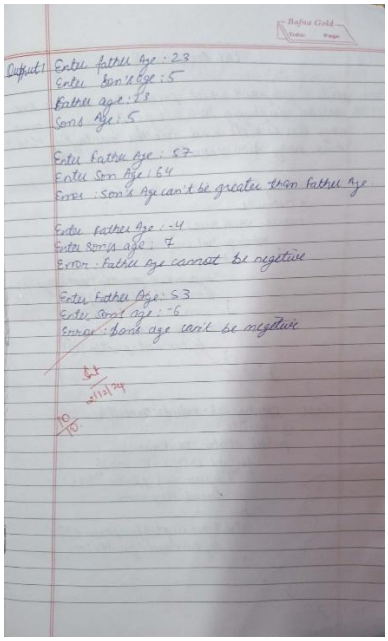
Program 7:

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

Algorithm:



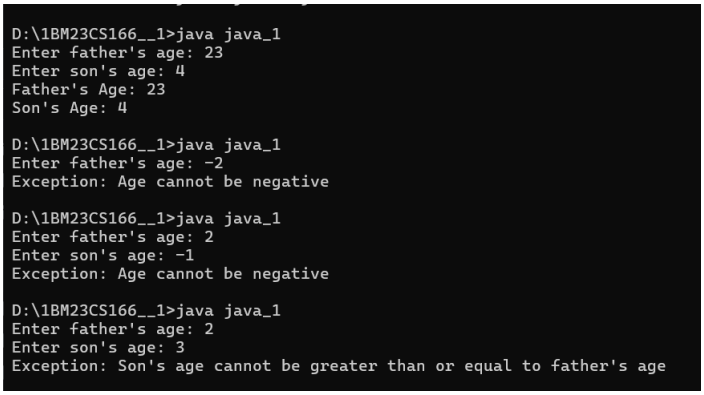




Code:

```
import java.util.Scanner;


class WrongAge extends Exception {

    public WrongAge() {
        super("Age Error");
    }
    public WrongAge(String message) {
        super(message);
    }
}
class Father {
    protected int fatherAge;
    public Father(Scanner scanner) throws WrongAge {
        System.out.print("Enter father's age: ");
        fatherAge = scanner.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
    public void display() {
        System.out.println("Father's Age: " + fatherAge);
    }
}
class Son extends Father {
    private int sonAge;
    public Son(Scanner scanner) throws WrongAge {
        super(scanner);
```

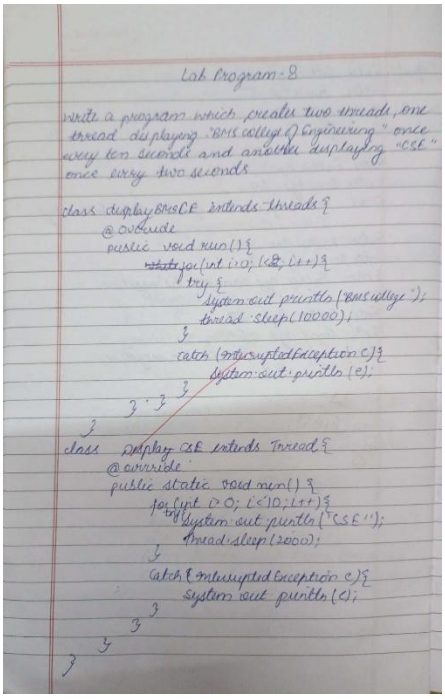
```
        System.out.print("Enter son's age: ");
        sonAge = scanner.nextInt();
        if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
        if (sonAge >= fatherAge) {
            throw new WrongAge("Son's age cannot be greater than or equal to father's age");
        }
    }
    public void display() {
        super.display();
        System.out.println("Son's Age: " + sonAge);
    }
}
public class java_1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
            Son son = new Son(scanner);
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        } finally {
            scanner.close();
        }
    }
}
```

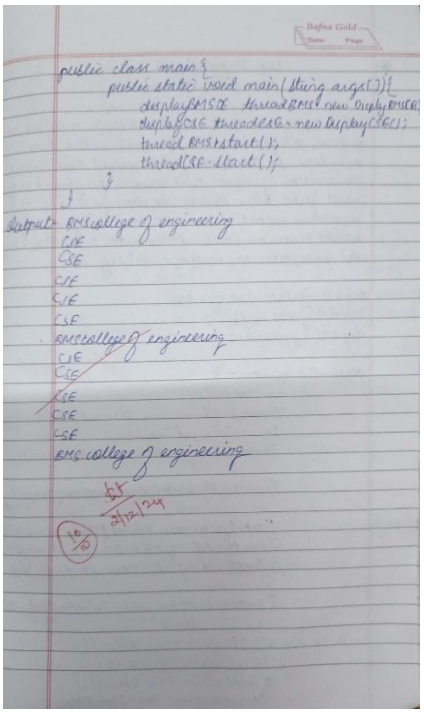


Program 8:

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.

Algorithm:





Code:

```
class DisplayBMS extends Thread {
   @Override
   public void run() {
      for(int i = 0;i<2;i++) {
         try {
            System.out.println("BMS College of Engineering");
            Thread.sleep(10000);
         } catch (InterruptedException e) {
            System.out.println(e);
         }
      }
   }
}

class DisplayCSE extends Thread {
   @Override
   public void run() {
      for(int i = 0;i<10;i++) {
         try {
            // Display "CSE" every 2 seconds
            System.out.println("CSE");
            Thread.sleep(2000);
         } catch (InterruptedException e) {
            System.out.println(e);
         }
      }
```

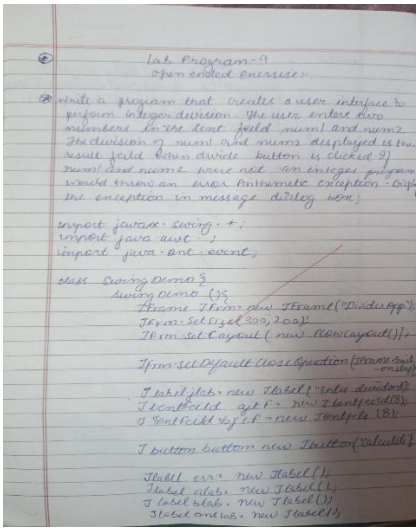
```
    }
}

public class threads {
    public static void main(String[] args) {
        DisplayBMS threadBMS = new DisplayBMS();
        DisplayCSE threadCSE = new DisplayCSE();
        threadBMS.start();
        threadCSE.start();
    }
}
```

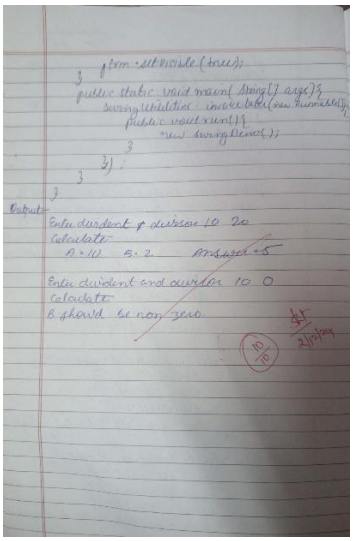
```
ads }
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
PS C:\Users\LAKSHYA KHANDELWAL\OneDrive\Desktop\java coding>
```

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.

Alogrithm:

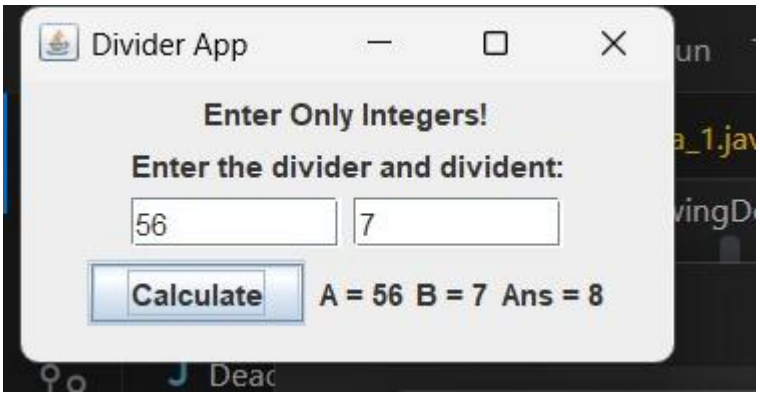




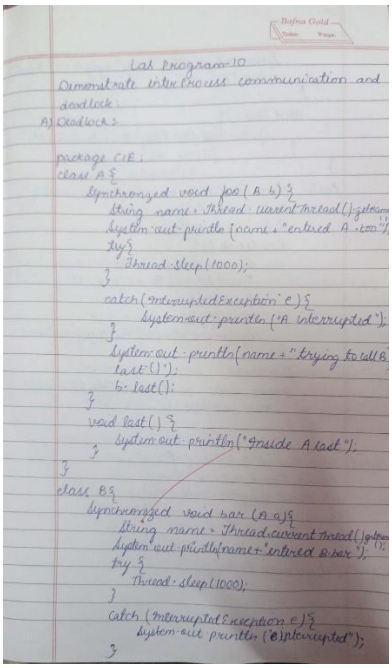
Code:

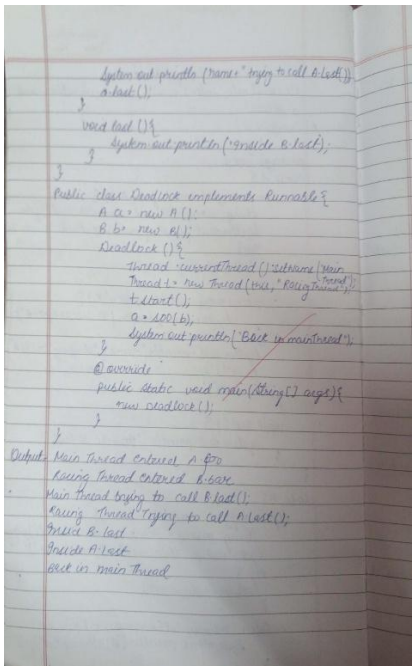
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
```

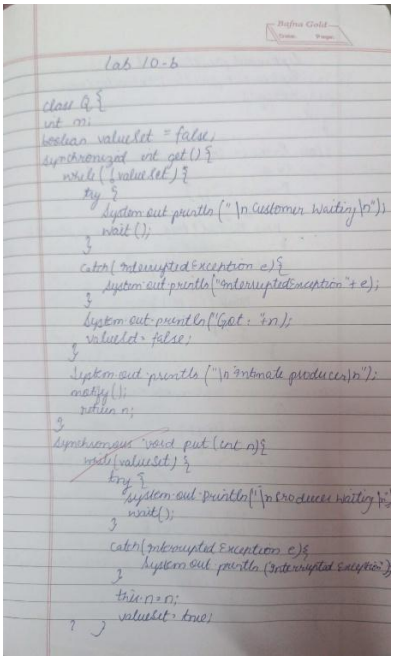
```
class SwingDemo{
SwingDemo(){
JFrame jfrm = new JFrame("Divider App");
jfrm.setSize(275, 150);
jfrm.setLayout(new FlowLayout());
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JLabel jlab = new JLabel("Enter the divider and divident:");
JTextField ajtf = new JTextField(8);
JTextField bjtf = new JTextField(8);
JButton button = new JButton("Calculate");
JLabel err = new JLabel();
JLabel alab = new JLabel();
JLabel blab = new JLabel();
JLabel anslab = new JLabel();
jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
ActionListener l = new ActionListener() {
public void actionPerformed(ActionEvent evt) { System.out.println("Action event from a text field"); } };
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent evt) { try{ int a =
Integer.parseInt(ajtf.getText()); int b = Integer.parseInt(bjtf.getText()); int ans = a/b;
alab.setText("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = "+ ans);
}
catch(NumberFormatException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("Enter Only Integers!"); }
catch(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("B should be NON zero!"); } }
});
jfrm.setVisible(true);
}
public static void main(String args[]){
SwingUtilities.invokeLater(new Runnable(){ public void run(){
new SwingDemo();
}
});
}
}
```

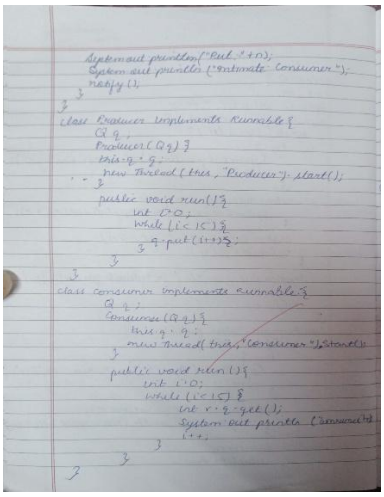


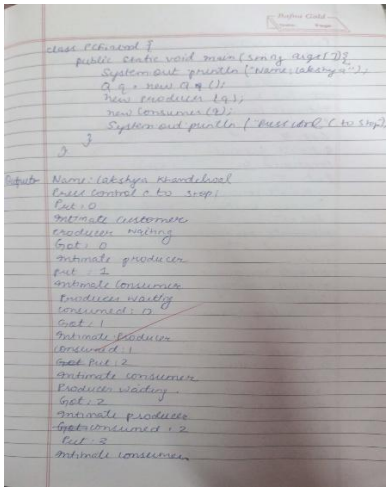
Program 10: Demonstrate interprocess Communication and Deadlock.

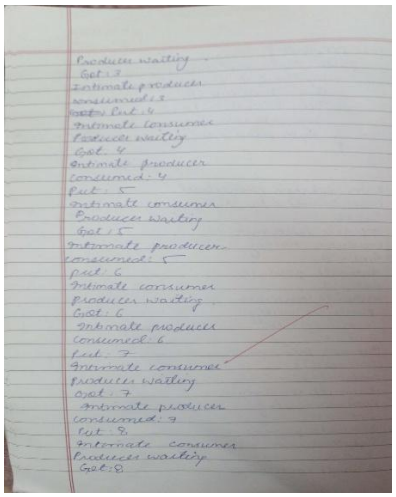


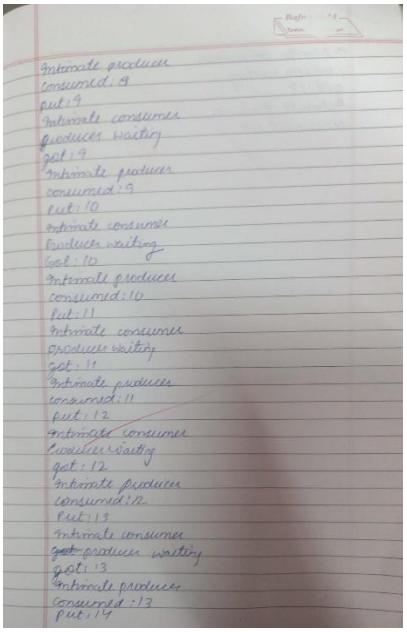


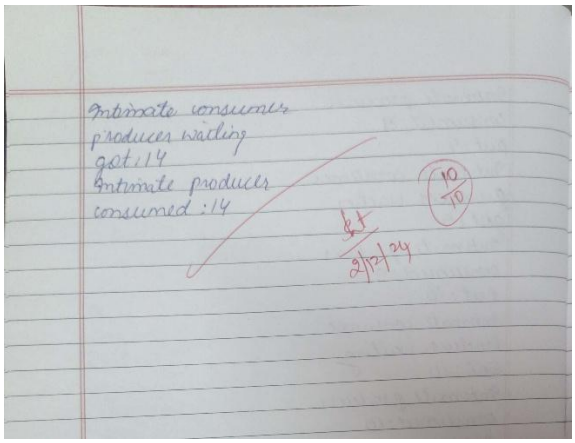








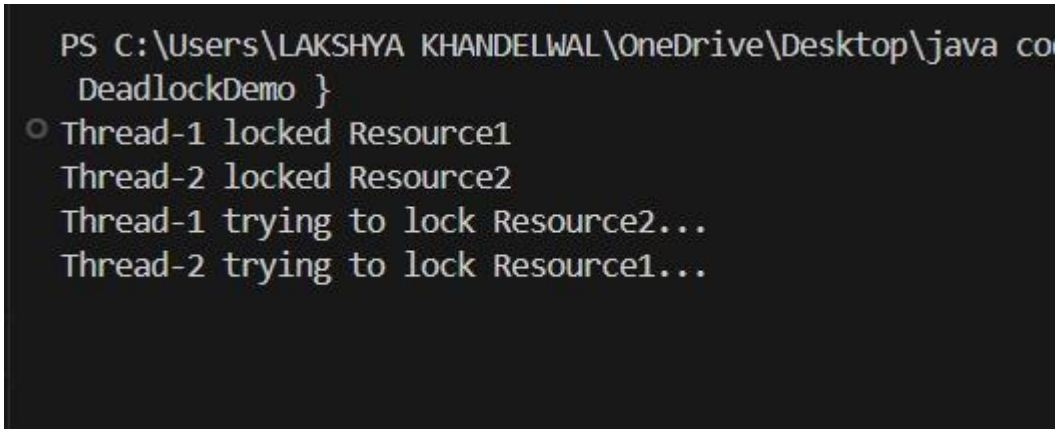




Code:

```
class Resource1 {
    synchronized void lock(Resource2 res2) {
        System.out.println(Thread.currentThread().getName() + " locked Resource1");

        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            System.out.println(e);
        }

        System.out.println(Thread.currentThread().getName() + " trying to lock Resource2...");
        res2.method2();
    }

    synchronized void method1() {
        System.out.println(Thread.currentThread().getName() + " is working with Resource1");
    }
}

class Resource2 {
    synchronized void lock(Resource1 res1) {
        System.out.println(Thread.currentThread().getName() + " locked Resource2");

        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            System.out.println(e);
        }
```

```
        System.out.println(Thread.currentThread().getName() + " trying to lock Resource1...");
        res1.method1();
    }

    synchronized void method2() {
        System.out.println(Thread.currentThread().getName() + " is working with Resource2");
    }
}

public class DeadlockDemo {
    public static void main(String[] args) {
        Resource1 res1 = new Resource1();
        Resource2 res2 = new Resource2();
        Thread t1 = new Thread(() -> res1.lock(res2), "Thread-1");
        Thread t2 = new Thread(() -> res2.lock(res1), "Thread-2");

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




```
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;

class SafeResource1 {
    private final Lock lock = new ReentrantLock();

    public boolean tryLockBoth(SafeResource1 otherResource) {
        while (true) {
            boolean gotFirstLock = lock.tryLock();
            boolean gotSecondLock = otherResource.tryLock();

            if (gotFirstLock && gotSecondLock) {
                System.out.println(Thread.currentThread().getName() + " acquired both locks");
                return true;
            }

            if (gotFirstLock) lock.unlock();
            if (gotSecondLock) otherResource.unlock();
        }
```

```
    }

    public boolean tryLock() {
        return lock.tryLock();
    }

    public void unlock() {
        lock.unlock();
    }
}

public class DeadlockResolved {
    public static void main(String[] args) {
        SafeResource1 res1 = new SafeResource1();
        SafeResource1 res2 = new SafeResource1();

        Thread t1 = new Thread(() -> {
            if (res1.tryLockBoth(res2)) {
                System.out.println("Thread-1 completed safely");
                res1.unlock();
                res2.unlock();
            }
        }, "Thread-1");

        Thread t2 = new Thread(() -> {
            if (res2.tryLockBoth(res1)) {
                System.out.println("Thread-2 completed safely");
                res2.unlock();
                res1.unlock();
            }
        }, "Thread-2");

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