VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

Object Oriented Java Programming

(23CS3PCOOJ)

Submitted by

LIKHITH D (1BM23CS170)

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 **Likhith D(1BM23CS170),** 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.

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

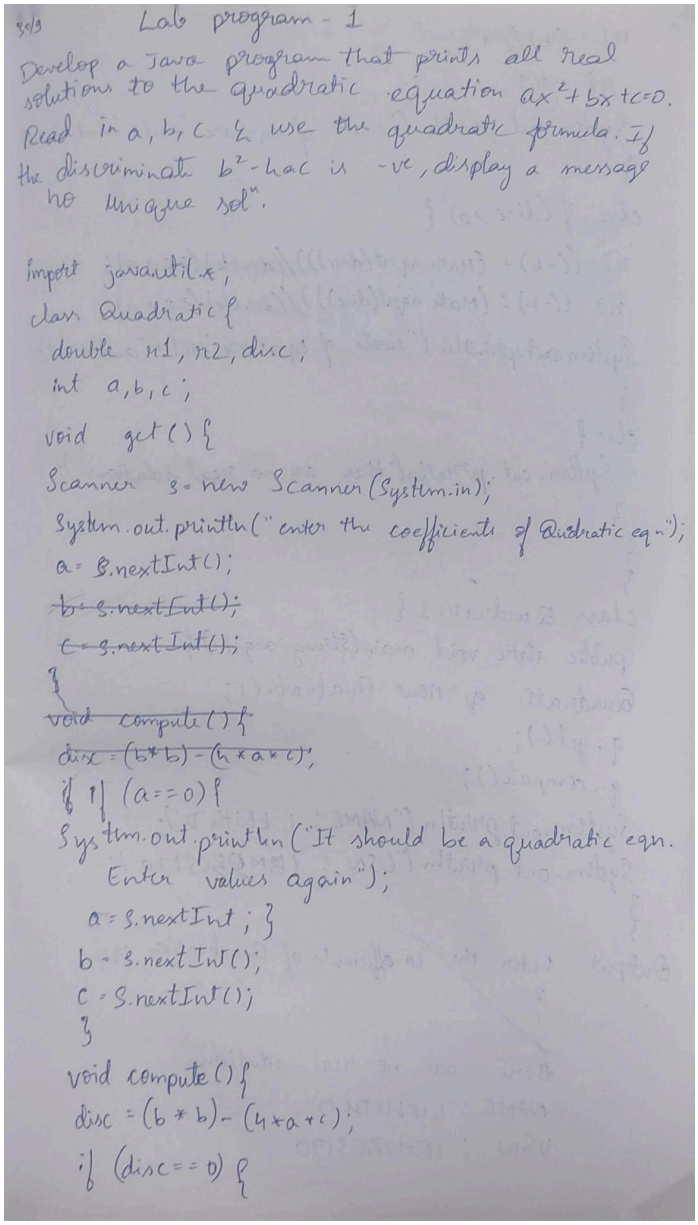
Index

Github Link:
https://github.com/LikhithD5206/JAVA-LAB/tree/main

Program 1

Implement Quadratic Equation

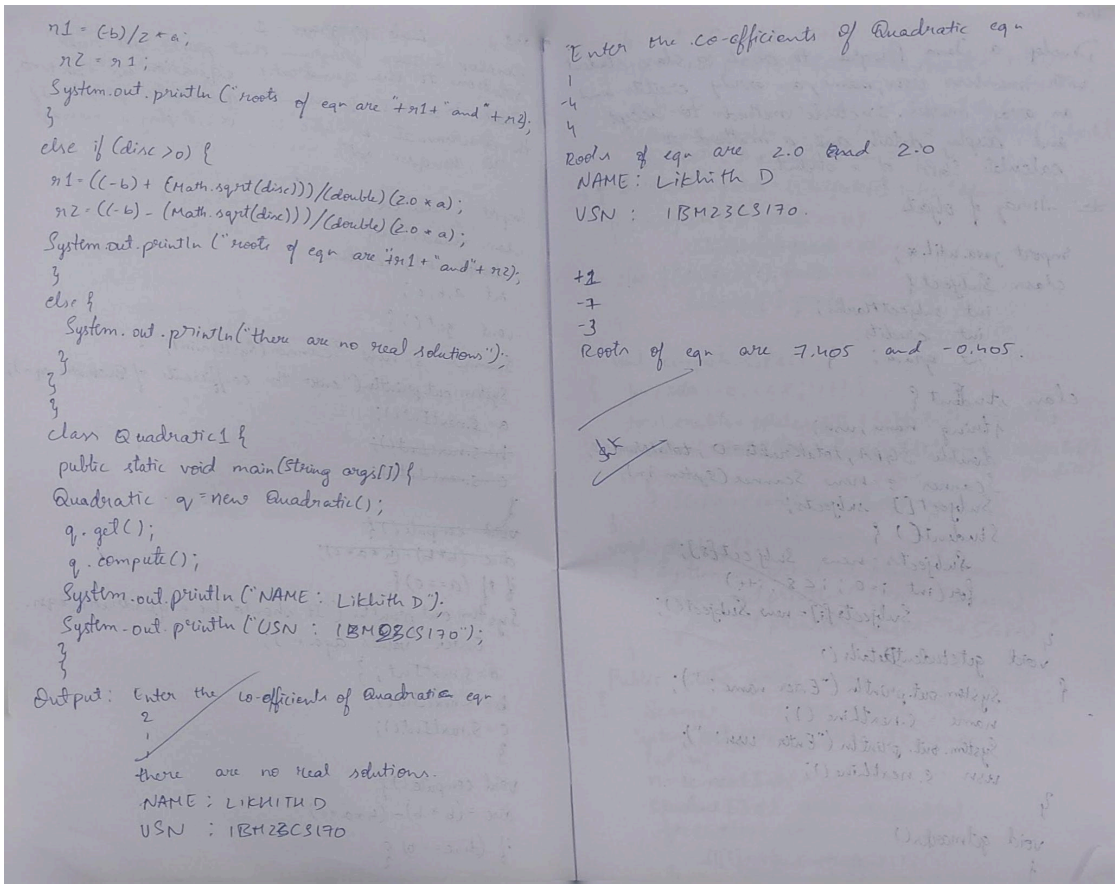
Algorithm:



Lab program - 1

Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c & use the quadratic formula. If the discriminate b^2-4ac is -ve, display a message no unique sol^n.

```
import java.util.*;
class Quadratic{
  double r1, r2, disc;
  int a,b,c;
  void get(){
  Scanner s= new Scanner(System.in);
  System.out.println("enter the coefficients of Quadratic eqn");
  a= s.nextInt();
  b= s.nextInt();
  c= s.nextInt();
  }
  void compute(){
  disc = (b*b)-(4*a*c);
  if ((a==0){
  System.out.println("It should be a quadratic eqn.
     Enter values again");
    a= s.nextInt; }
    b= s.nextInt();
    c= s.nextInt();
    }
    void compute(){
    disc = (b*b)-(4*a*c);
    if (disc==0){
```



Code:

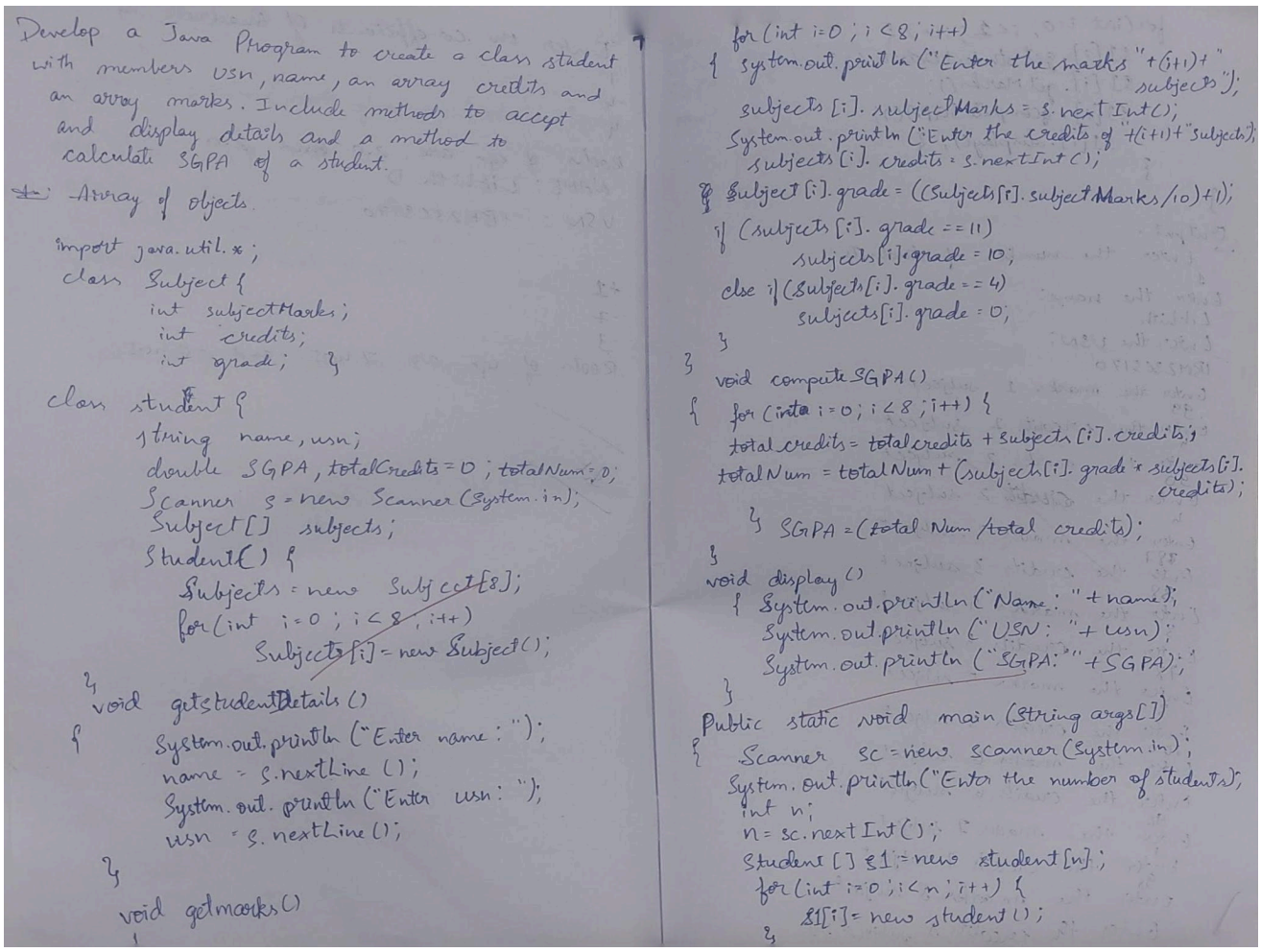
```
import java.util.*;
class Quadratic{
double r1,r2,disc;
int a,b,c;

void get(){
Scanner s=new Scanner(System.in);

System.out.println("enter the coefficients of Quadratic eqn");
a=s.nextInt();
if (a==0){
System.out.println("It should be a quadratic eqn. Enter values again");
a=s.nextInt();}

b=s.nextInt();
c=s.nextInt();
}
void compute()

{
disc=(b*b)-(4*a*c);
```

```
if (disc==0)
{
r1=(-b)/2*a;
r2=r1;
System.out.println("roots of eqn are"+r1+"and"+r2);
}
else if (disc>0)
{
r1=((-b)+ (Math.sqrt(disc)))/(double)(2.0*a);
r2=((-b)- (Math.sqrt(disc)))/(double)(2.0*a);
System.out.println("roots of eqn are " +r1+ "and" +r2);
}
else{
System.out.println("there are no real solutions");
}
}
}
class Quadratic1{
public static void main(String args[]){
Quadratic q=new Quadratic();
q.get();
q.compute();
System.out.println("NAME: Likhith D");
System.out.println("USN : 1BM23CS170");
}}
```

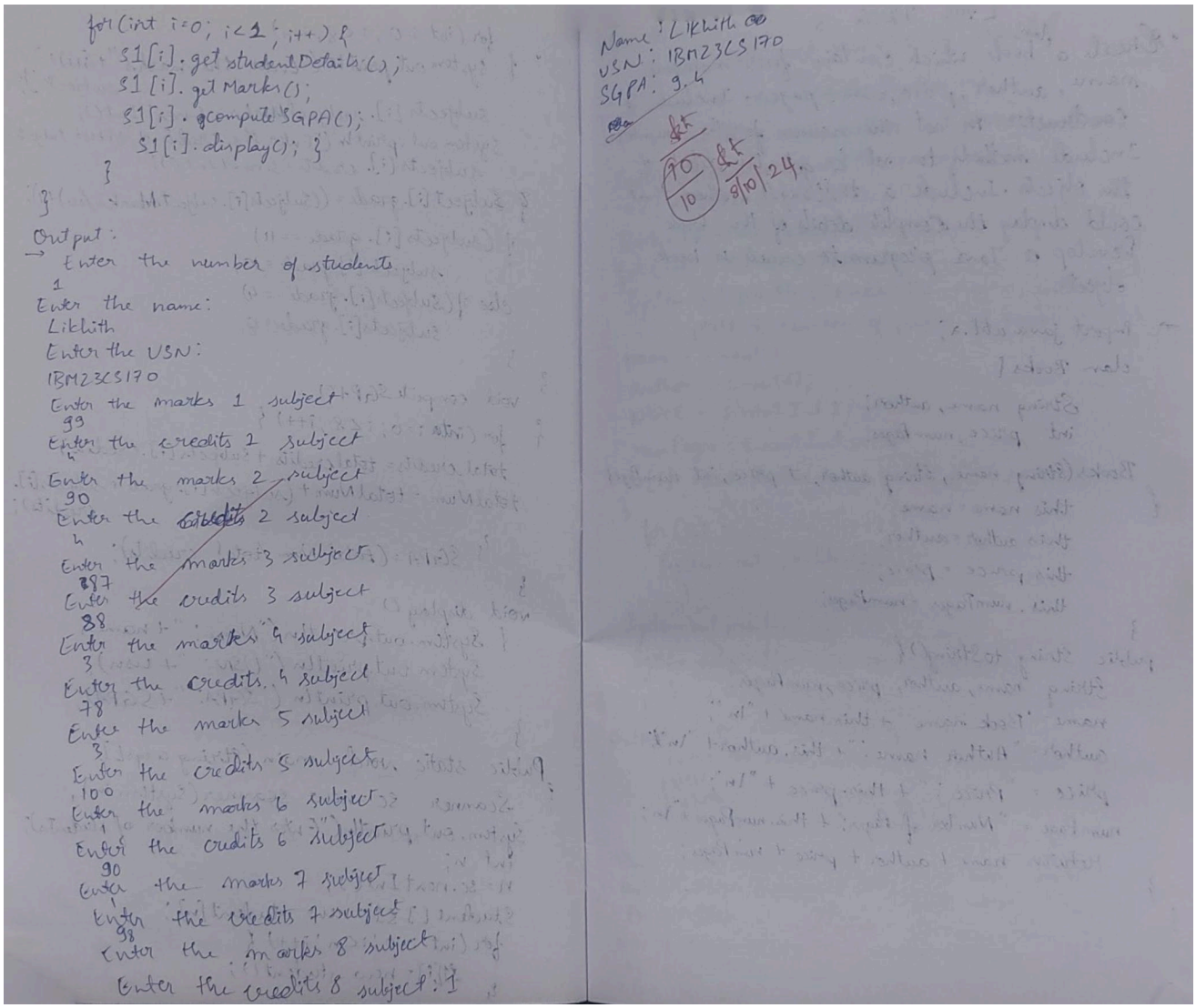
```
D:\1BM23CS170\OOJ>java Quadratic1
enter the coefficients of Quadratic eqn
0
It should be a quadratic eqn. Enter values again
2
3
2
there are no real solutions
NAME: Likhith D
USN : 1BM23CS170

D:\1BM23CS170\OOJ>java Quadratic1
enter the coefficients of Quadratic eqn
1
-7
-3
roots of eqn are 7.405124837953327and-0.405124837953327
NAME: Likhith D
USN : 1BM23CS170

D:\1BM23CS170\OOJ>java Quadratic1
enter the coefficients of Quadratic eqn
2
1
1
there are no real solutions
NAME: Likhith D
USN : 1BM23CS170

D:\1BM23CS170\OOJ>java Quadratic1
enter the coefficients of Quadratic eqn
1
-4
4
roots of eqn are2.0and2.0
NAME: Likhith D
USN : 1BM23CS170
```

Program 2
SGPA

Algorithm:



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

Array of objects.

import java.util.*;
class Subject {
        int subjectMarks;
        int credits;
        int grade; }

class student {
        string name, usn;
        double SGPA, totalCredits = 0 ; totalNum = 0;
        Scanner s = new Scanner (System.in);
        Subject[] subjects;
        Student() {
          Subjects = new Subject[8];
          for(int i=0 ; i<8; i++)
                Subjects[i] = new Subject();
    }
    void getstudentDetails()
    {   System.out.println ("Enter name : ");
        name = s.nextLine ();
        System.out.println ("Enter usn : ");
        usn = s.nextLine ();
    }
    void getmarks()
    {
      for (int i=0 ; i<8; i++)
    { System.out.println ("Enter the marks "+(i+1)+"
                                        subjects");
      subjects [i]. subjectMarks = s.nextInt();
      System.out.println ("Enter the credits of "+(i+1)+"subjects");
       subjects [i]. credits = s.nextInt();
    Subject [i]. grade = ((Subjects[i]. subjectMarks/10)+1);
    if (subjects [i]. grade == 11)
          subjects[i].grade = 10;
    else if(subjects[i]. grade == 4)
          subjects[i]. grade = 0;
    }
}
    void computeSGPA()
{   for (int i=0; i<8 ;i++) {
    total credits = total credits + subjects [i]. credits;
    totalNum = totalNum + (subject[i]. grade * subjects[i].
                                        credits);
      } SGPA = (total Num /total credits);
}
void display()
    { System.out.println("Name: "+name);
      System.out.println ("USN: "+usn);
      System.out.println ("SGPA:"'"+SGPA);
    }
Public static void main (String args[])
{   Scanner sc = new scanner(System.in);
    System.out.println("Enter the number of students);
    int n;
    n = sc.nextInt();
    Student [] s1 = new student[n];
     for(int i=0;i<n;i++) {
        s1[i] = new student();
     }
```



CODE:

```
import java.util.*;

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

        void calculateGrade() {
        if (subjectMarks >= 90) {
            grade = 10;
        } else if (subjectMarks >= 80) {
            grade = 9;
        } else if (subjectMarks >= 70) {
```

```
            grade = 8;
        } else if (subjectMarks >= 60) {
            grade = 7;
        } else if (subjectMarks >= 50) {
            grade = 6;
        } else if (subjectMarks >= 40) {
            grade = 5;
        } else {
            grade = 0; // Fail
        }
    }
}

class Student {
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject subject[];

      Student() {
        subject = new Subject[8]; // Array of 9 subjects
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject(); // Create an array of Subject objects
        }
        s = new Scanner(System.in);
    }

        void getStudentDetails() {
        System.out.print("Enter Name: ");
        name = s.nextLine();
        System.out.print("Enter USN: ");
        usn = s.nextLine();
    }


    void getMarks() {
        for (int i = 0; i < 8; i++) {
            System.out.print("Enter marks for Subject " + (i + 1) + ": ");
            subject[i].subjectMarks = s.nextInt();
            System.out.print("Enter credits for Subject " + (i + 1) + ": ");
            subject[i].credits = s.nextInt();


            subject[i].calculateGrade();
```

```
            if (subject[i].subjectMarks > 100) {
                System.out.println("Invalid marks. Marks should not exceed 100.");
                subject[i].subjectMarks = 100;
            } else if (subject[i].subjectMarks < 0) {
                System.out.println("Invalid marks. Marks should not be negative.");
                subject[i].subjectMarks = 0;            }
        }
    }

     void computeSGPA() {
        double totalCredits = 0;
        double totalGradePoints = 0;

        for (int i = 0; i < 8; i++) {
            totalCredits += subject[i].credits;
            totalGradePoints += subject[i].credits * subject[i].grade;
        }

        if (totalCredits > 0) {
            SGPA = totalGradePoints / totalCredits;
        } else {
            SGPA = 0; // Handle case with zero credits
        }
    }


    void displayResults() {
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
       System.out.println("SGPA:"+ SGPA);
    }



    public static void main(String[] args) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResults();
    }
}
```

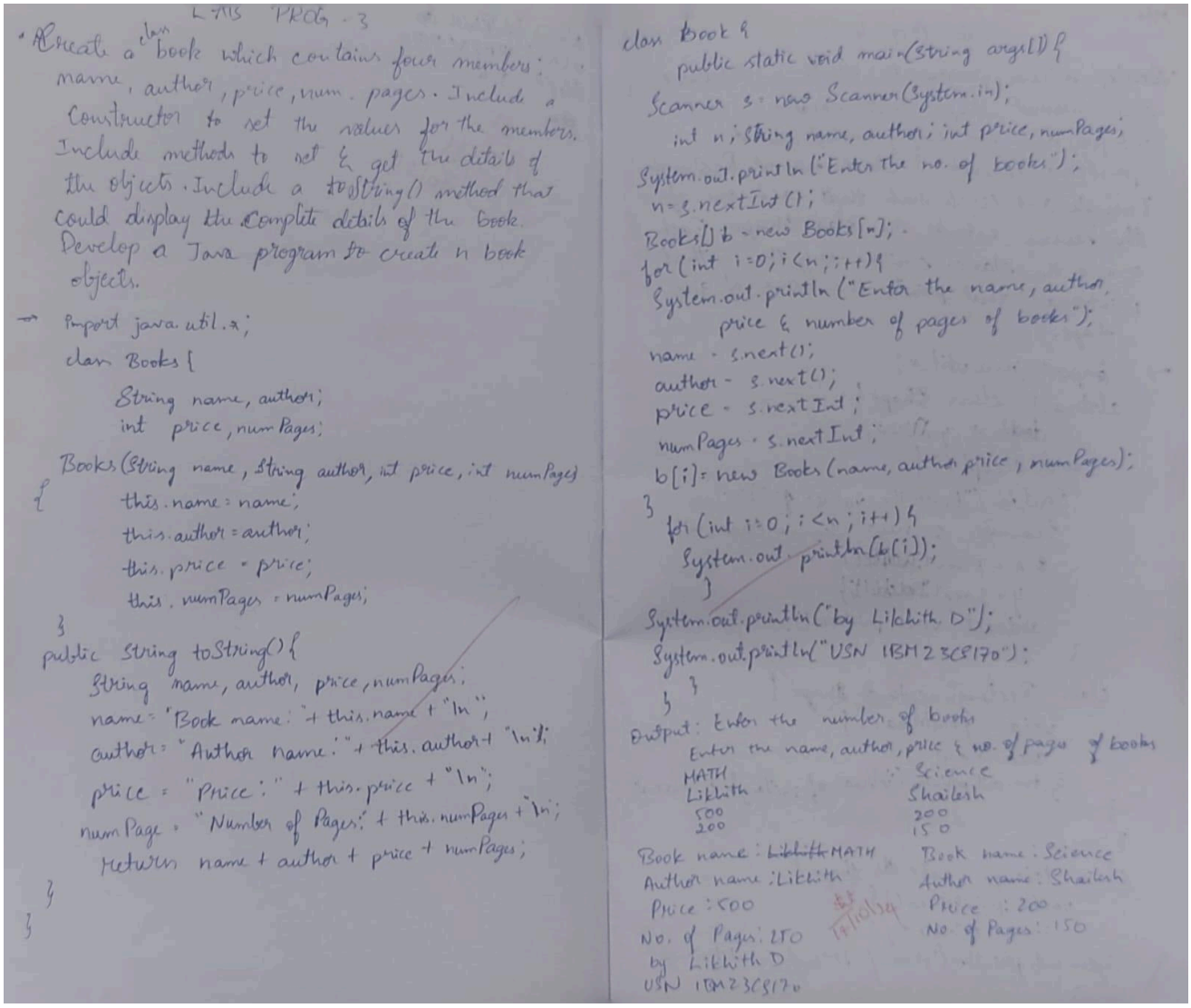
Command Prompt

```
D:\1BM23CS170\OOJ>javac student.java

D:\1BM23CS170\OOJ>java student
Error: Could not find or load main class student
Caused by: java.lang.NoClassDefFoundError: Student (wrong name: student)

D:\1BM23CS170\OOJ>java Student
Enter Name: Likhith D
Enter USN: 1BM23CS170
Enter marks for Subject 1: 99
Enter credits for Subject 1: 4
Enter marks for Subject 2: 99
Enter credits for Subject 2: 4
Enter marks for Subject 3: 99
Enter credits for Subject 3: 3
Enter marks for Subject 4: 99
Enter credits for Subject 4: 3
Enter marks for Subject 5: 99
Enter credits for Subject 5: 3
Enter marks for Subject 6: 99
Enter credits for Subject 6: 1
Enter marks for Subject 7: 99
Enter credits for Subject 7: 1
Enter marks for Subject 8: 99
Enter credits for Subject 8: 1
Name: Likhith D
USN: 1BM23CS170
SGPA:10.0

D:\1BM23CS170\OOJ>
```

Program 3

Book Details

Algorithm:



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 & 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.*;
class Books{
    String name, author;
    int price, numPages;
    Books(String name, String author, int price, int numPages){
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString(){
        String name, author, price, numPages;
        name = "Book name: " + this.name + "\n";
        author = "Author name: " + this.author + "\n";
        price = "Price: " + this.price + "\n";
        numPage = "Number of Pages: " + this.numPages + "\n";
        return name + author + price + numPages;
    }
}
class Book {
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        int n; String name, author; int price, numPages;
        System.out.println("Enter the no. of books");
        n = s.nextInt();
        Books[] b = new Books[n];
        for(int i=0; i<n; i++){
            System.out.println("Enter the name, author, price & number of pages of books");
            name = s.next();
            author = s.next();
            price = s.nextInt;
            numPages = s.nextInt;
            b[i] = new Books(name, author, price, numPages);
        }
        for(int i=0; i<n; i++){
            System.out.println(b[i]);
        }
        System.out.println("by Likhith D");
        System.out.println("USN 1BM23CS170");
    }
}
```

Output:

```
Enter the number of books
Enter the name, author, price & no. of pages of books
MATH
Likhith
500
200
Science
Shailesh
200
150
Book name: MATH
Author name: Likhith
Price: 500
No. of Pages: 200
Book name: Science
Author name: Shailesh
Price: 200
No. of Pages: 150
by Likhith D
USN 1BM23CS170
```

Code:

```
import java.util.*;
class Books{
        String name,author;
        int numPages,price;


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

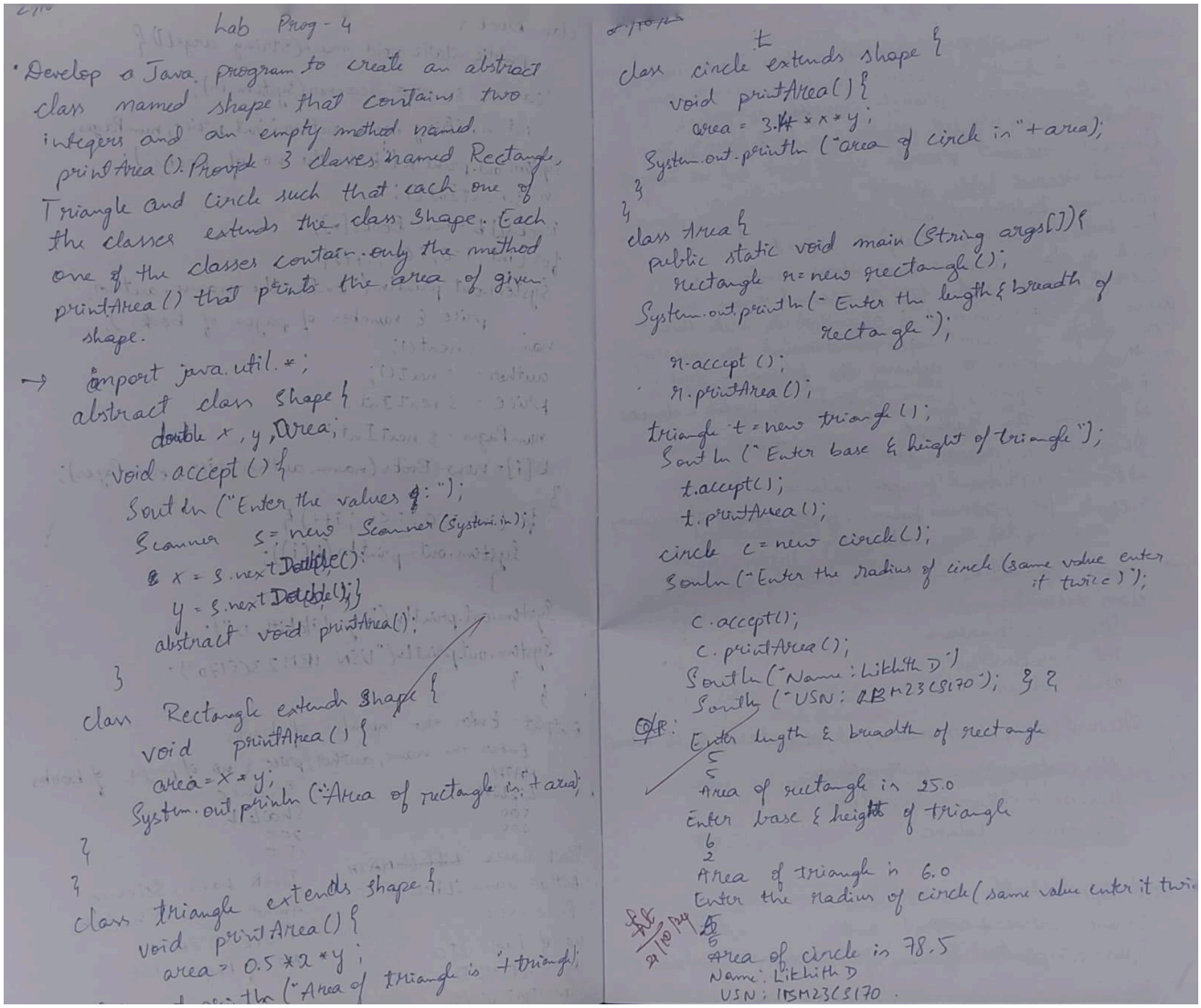
```
this.author = author;
this.price = price;
this.numPages = numPages;
}
public String toString()
{
        String name, author, price, numPages;
        name = "Book name: " + this.name + "\n";
        author = "Author name: " + this.author + "\n";
        price = "Price: " + this.price + "\n";
  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

class book
{

        public static void main(String args[])

{

        Scanner s = new Scanner(System.in);

        int n; String name; String author; int price; int numPages;

System.out.println(" enter the number of books");
n = s.nextInt();


Books[] b = new Books[n];
for ( int i=0;i<n;i++)
{
System.out.println(" enter the name,author, price and number of pages of books");

name=s.next();
author=s.next();
price=s.nextInt();
numPages=s.nextInt();
b[i]=new Books(name,author,price,numPages);

}
for(int i=0;i<n;i++){
System.out.println(b[i]);

   }
System.out.println("by Likhith D");
```

```
System.out.println("USN 1BM23CS170");

    }
  }
```

```
D:\1BM23CS170\OOJ>javac book.java

D:\1BM23CS170\OOJ>java book
 enter the number of books
2
 enter the name,author, price and number of pages of books
MATH
Likhith
500
250
 enter the name,author, price and number of pages of books
chemistry
Shailesh
200
150
Book name: MATH
Author name: Likhith
Price: 500
Number of pages: 250

Book name: chemistry
Author name: Shailesh
Price: 200
Number of pages: 150

by Likhith D
USN 1BM23CS170
```

Program 4
Shape Area

Algorithm:


Lab Prog - 4

• Develop a Java program to create an abstract class named shape that contains two integers and an empty method named printArea(). Provide 3 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 given shape.

```
import java.util.*;
abstract class Shape{
    double x, y, Area;
    void accept(){
        Scanner s = new Scanner(System.in);
        x = s.nextDouble();
        y = s.nextDouble();}
    abstract void printArea();
}
class Rectangle extends Shape{
    void printArea(){
        area = x * y;
        System.out.println("Area of rectangle is:"+area);
    }
}
class Triangle extends Shape{
    void printArea(){
        area = 0.5 * x * y;
        th("Area of triangle is :"+triangl
class Circle extends Shape{
    void printArea(){
        area = 3.14 * x * y;
        System.out.println("area of circle is:"+area);
    }
}
class Area{
    public static void main(String args[]){
        Rectangle r = new Rectangle();
        System.out.println("Enter the length & breadth of
                            rectangle");
        r.accept();
        r.printArea();
        Triangle t = new Triangle();
        Sout ln("Enter base & height of triangle");
        t.accept();
        t.printArea();
        Circle c = new Circle();
        Sout ln("Enter the radius of circle (same value enter
                                    it twice)");
        c.accept();
        c.printArea();
        Sout ln("Name: Likhith D")
        Sout ln("USN: 1BM23CS170");  } }
```

O/P:

```
Enter length & breadth of rectangle
5
5
Area of rectangle is 25.0
Enter base & height of triangle
6
2
Area of Triangle is 6.0
Enter the radius of circle (same value enter it twice)
5
Area of circle is 78.5
Name: Likhith D
USN: 1BM23CS170
```

Code:

```
import java.util.*;
abstract class shape{
double x,y,area;

void accept(){
Scanner s=new Scanner(System.in);
x=s.nextDouble();
y=s.nextDouble();
```

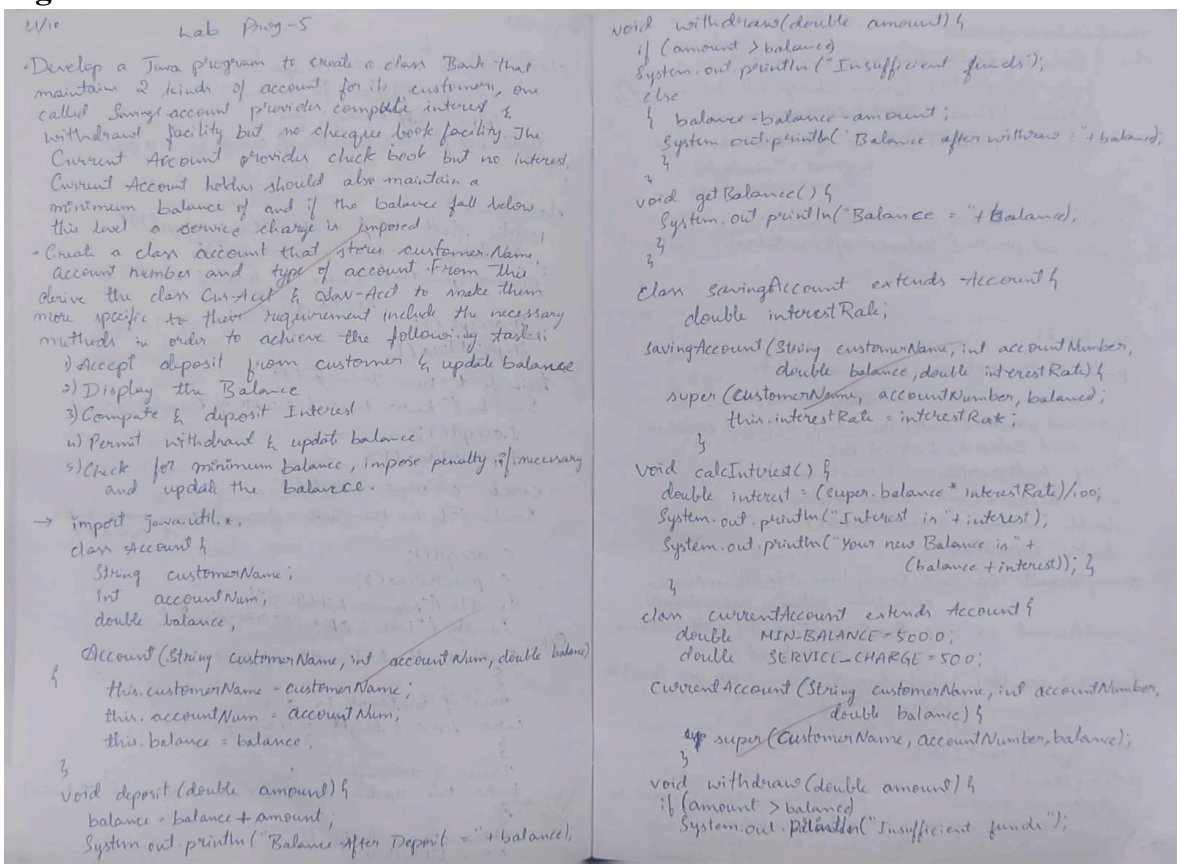
```
}
abstract void printArea();
}

class rectangle extends shape{
void printArea(){
area = x*y;
System.out.println(" area of rectangle is"+ area);

}}
class triangle extends shape{
void printArea(){
area=0.5*x*y;
System.out.println(" area of triangle is"+ area);

}}
class circle extends shape{
void printArea(){
area = 3.14*x*y;
System.out.println(" area of circle is"+ area);

}}
class Area{
public static void main(String args[]){
 rectangle r= new rectangle();
System.out.println(" \n enter the length and breadth of rectangle\n ");
r.accept();
r.printArea();
 triangle t= new triangle();
System.out.println("\n enter the base and height of triangle \n");
t.accept();
t.printArea();
 circle c= new circle();
System.out.println("\n enter the radius of circle(enter the same value twice)");
c.accept();
c.printArea();
System.out.println("Name: Likhith D");
System.out.println("USN : 1BM23CS170");
 }
}
```

```
D:\1BM23CS170\OOJ>javac Area.java

D:\1BM23CS170\OOJ>java Area

 enter the length and breadth of rectangle

5
5
 area of rectangle is25.0

 enter the base and height of triangle

6
2
 area of triangle is6.0

 enter the radius of circle(enter the same value twice)
5
5
 area of circle is78.5
Name: Likhith D
USN : 1BM23CS170

D:\1BM23CS170\OOJ>java Area

 enter the length and breadth of rectangle

2
2
 area of rectangle is4.0

 enter the base and height of triangle

5
4
 area of triangle is10.0

 enter the radius of circle(enter the same value twice)

6
9
 area of circle is169.56
Name: Likhith D
USN : 1BM23CS170

D:\1BM23CS170\OOJ>
```

Program 5

Bank Class

Algorithm:



21/10 Lab Prog-5

- Develop a Java program to create a class Bank that maintains 2 kinds of account for its customers, one called Savings account provides compute interest & withdrawal facility but no cheques book facility. The Current Account provides check book but no interest. Current Account holder should also maintain a minimum balance of and if the balance fall 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 class Cur-Acct & Sav-Acct to make them more specific to their requirement include the necessary methods in order to achieve the following tasks:
  1) Accept deposit from customer & update balance
  2) Display the Balance
  3) Compute & deposit Interest
  4) Permit withdrawal & update balance
  5) Check for minimum balance, impose penalty if necessary and update the balance.

```
→ 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 withdraws(double amount) {
    if (amount > balance)
    System.out.println("Insufficient funds");
    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);
    System.out.println("your new Balance is " +
                       (balance + interest)); }
}
class currentAccount extends Account {
    double MIN_BALANCE = 500.0;
    double SERVICE_CHARGE = 500;
currentAccount(String customerName, int accountNumber,
               double balance) {
    super(customerName, accountNumber, balance);
}
void withdraws(double amount) {
    if (amount > balance)
    System.out.println("Insufficient funds");
```

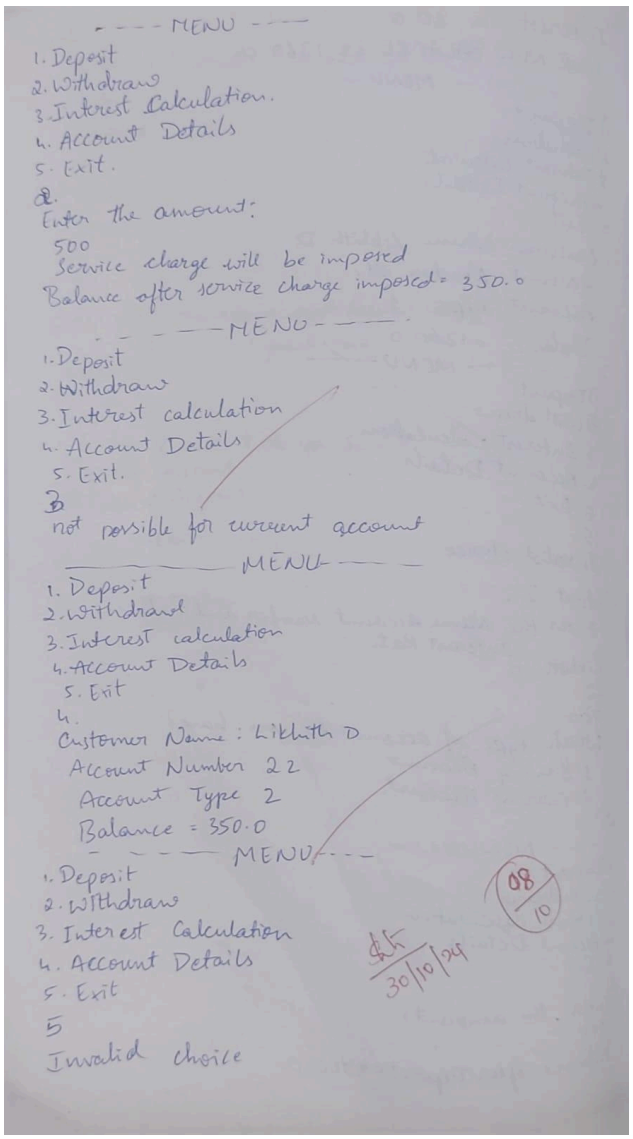
```
else if (balance < amount)
else if ((balance - amount) < MIN_BALANCE){
    System.out.println("Service charge will be imposed");
    balance = balance - SERVICE_CHARGE - amount;
    System.out.println("Balance after service charge
                        imposed = "+balance);
  }
  else {
    balance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

class Bank1
{
  public static void main(String args[]) {
    Scanner s=new Scanner(System.in);
  System.out.println("Enter the Name, Account Number
      and Balance, Interest Rate");
  String customerName = s.nextLine();
    int   accountNum = s.nextDouble();
    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("------MENU------ \n 1.Deposit \n
        2.Withdraw \n 3.Interest Calculation \n
          4.Account Details \n 5.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;
    }
}
```

Output 1: Enter the Name and Account Number and Balance, Interest Rate

```
Likhith D.
21
1000
Which type of account do you have?
  1) Saving Account
  2) Current Account
1
------MENU------
1. Deposit
2. Withdraw
3. Interest Calculation
4. Account Details
5. Exit
1
Enter the amount:
500
Balance after Deposit = 1500.0
------MENU------
1. Deposit
2. Withdraw
3. Interest Calculation
4. Account Details
5. Exit
2
Enter the amount:
300
Balance after withdraw = 1200.0
------MENU------
1. Deposit
2. Withdraw
3. Interest Calculation
4. Account Details
5. Exit
3
Interest is 60.0
YOUR NEW BALANCE IS 1260.0
------MENU------
1) Deposit
2) Withdraw
3) Interest Calculate
4) Account Details
5. Exit
4
Customer Name Likhith D
Account Number 21
Account Type 1
Balance = 1260.0
------MENU------
1) Deposit
2) Withdraw
3. Interest Calculation
4. Account Details
5. Exit
5
Invalid choice
```

Output 2:

```
Enter the Name, Account Number and Balance,
    Interest Rate
Likhith D
22
700
Which type of account do you have?
  1. Saving Account
  2. Current Account.
2
------MENU------
1. Deposit
2. Withdraw
3. Interest Calculation
4. Account Details
5. Exit
1
Enter the amount:
200
Balance after Deposit = 900.0
```



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);
System.out.println("YOUR NEW BALANCE IS " + (balance+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-amount;
System.out.println("Balance after Service charge imposed = "+balance);}
else
{
balance=balance-amount;
System.out.println("Balance after Withdraw = "+balance);}
}}

class Bank1{
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("            MENU          \n1.Deposit\n 2.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("Name: Likhith D");
System.out.println("USN: 1BM23CS170");
}}
```

Current Account :

```
C:\BMSCE\OOJ>javac Bank1.java

C:\BMSCE\OOJ>java Bank1
Enter the Name and Account Number And Balance, Interest Rate
Likhith D
22
700
Which type of account do you have?
 1.Saving Account
 2.Current Account
2
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
1
Enter the amount :
200
Balance after Deposit = 900.0
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
2
Enter the amount :
500
Service charge will be imposed
Balance after Service charge imposed = 350.0
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
3
not possible for current account
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
4
Customer Name Likhith D
Account Number 22
Account Type 2
Balance = 350.0
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
5
Invalid choice
```

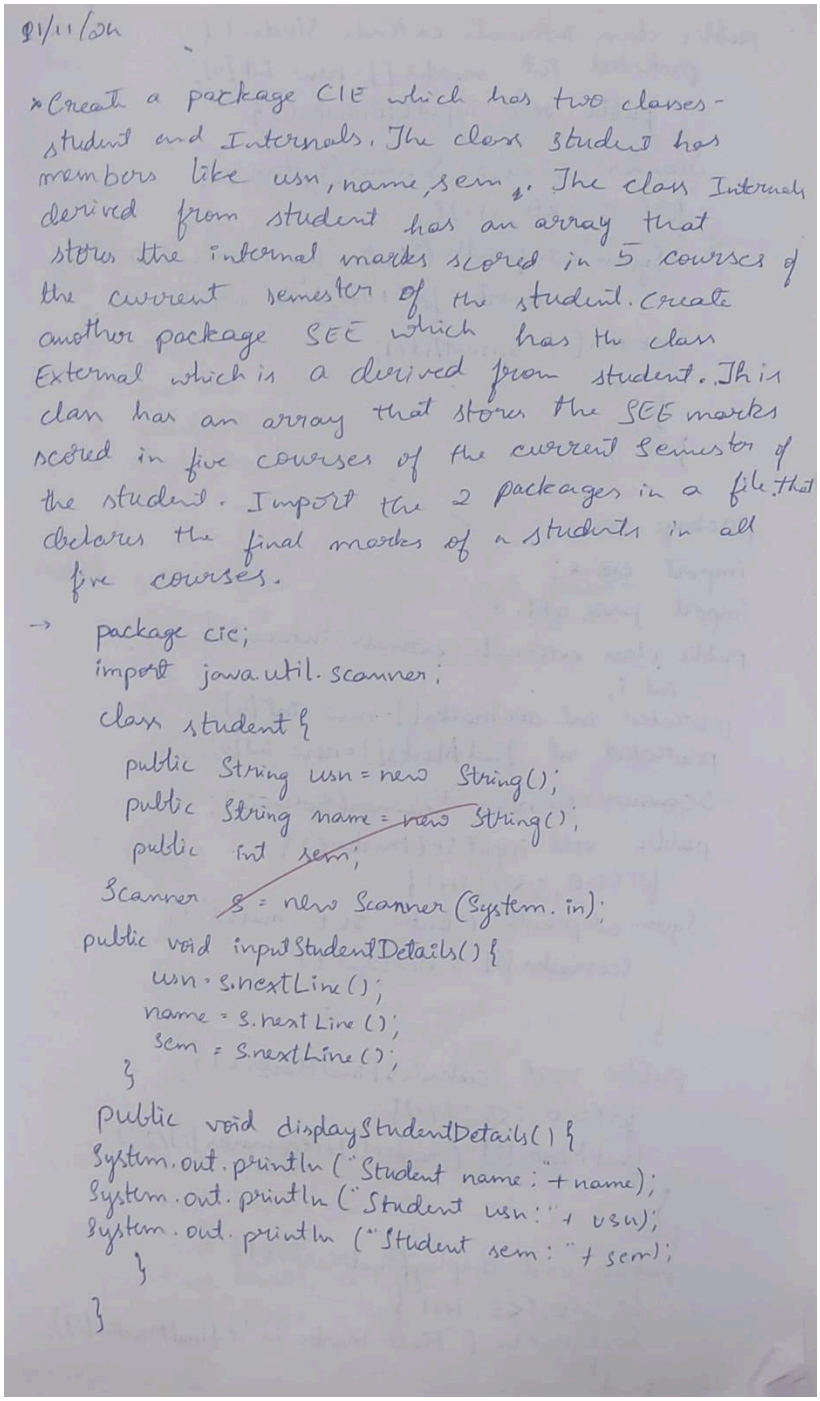
Savings account:

```
C:\BMSCE\OOJ>javac Bank1.java

C:\BMSCE\OOJ>java Bank1
Enter the Name and Account Number And Balance, Interest Rate
Likhith D
21
1000
Which type of account do you have?
 1.Saving Account
 2.Current Account
1
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
1
Enter the amount :
500
Balance after Deposit = 1500.0
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
2
Enter the amount :
300
Balance after Withdraw = 1200.0
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
3
Interest is 60.0
YOUR NEW BALANCE IS 1260.0
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
4
Customer Name Likhith D
Account Number 21
Account Type 1
Balance = 1200.0
            MENU
1.Deposit
 2.withdraw
3.Interest Calculation
4.Account Details
5.Exit
5
Invalid choice
```

Program 6

Packages

Algorithm:

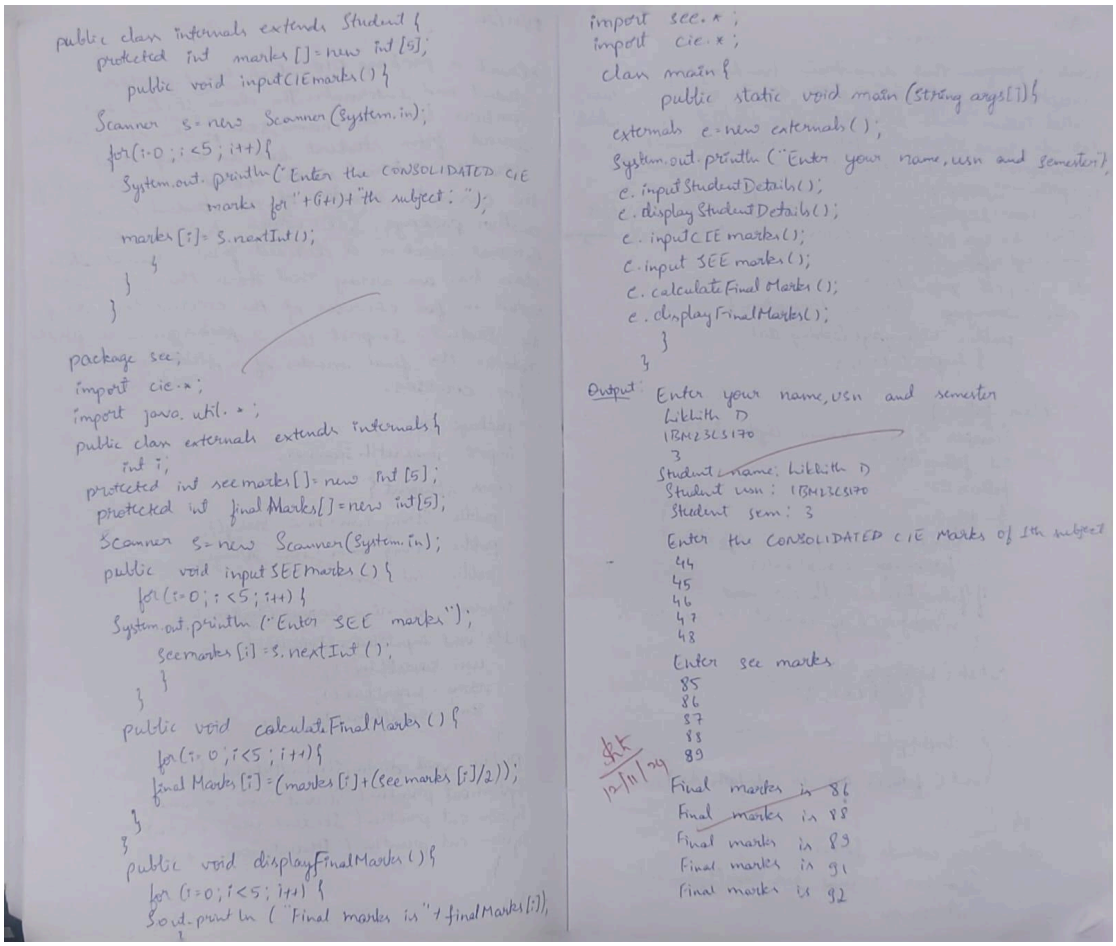


21/11/24

*Create a package CIE which has two classes - student and Internals. The class student has members like usn, name, sem. The class Internals derived from student has an array that stores the internal marks scored in 5 courses of the current semester of the student. Create another package SEE which has the class External which is a derived from student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the 2 packages in a file that declares the final marks of a students in all five courses.

→

```
package cie;
import java.util.scanner;
class student{
  public String usn = new String();
  public String name = new String();
  public int sem;
Scanner s = new Scanner(System.in);
public void inputStudentDetails(){
    usn = s.nextLine();
    name = s.nextLine();
    sem = s.nextLine();
  }
  public void displayStudentDetails(){
System.out.println("Student name:"+name);
System.out.println("Student usn:"+usn);
System.out.println("Student sem:"+sem);
  }
}
```



Code:

Externals code:

```
package see;

import cie.*;

import java.util.Scanner;

public class externals extends internals {
int i;
protected int seemarks[]=new int[5];

protected int finalMarks[]=new int[5];

Scanner s=new Scanner(System.in);
public void inputSEEmarks()
  {
for(i=0;i<5;i++)
{
System.out.println("Enter see marks");
seemarks[i]=s.nextInt();
  }
```

```
}
public void calculateFinalMarks() {
for(i=0;i<5;i++){
finalMarks[i]=(marks[i]+(seemarks[i]/2));


}

 }

public void displayFinalMarks() {

for (i=0;i<5;i++){
System.out.println("Final marks is"+finalMarks[i]);
}}}
```

Internals :

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

class Student {

public String usn = new String();
 public String name = new String();
 public int sem;
Scanner s=new Scanner(System.in);

public void inputStudentDetails() {
 usn=s.nextLine();
 name=s.nextLine();
 sem=s.nextInt();
}

public void displayStudentDetails() {
System.out.println("student name: " + name);
System.out.println("student usn: " + usn);
System.out.println("student semester: " + sem);
  }
}


public class internals extends Student {

protected int marks[] = new int[5];

public void inputCIEmarks()

{
Scanner s = new Scanner(System.in);
for(int i=0;i<5;i++){
System.out.println("enter the CONSOLIDATED CIE marks  of " +(i+1)+"th  subject");
marks[i]=s.nextInt();    } }}
```

Main Code :

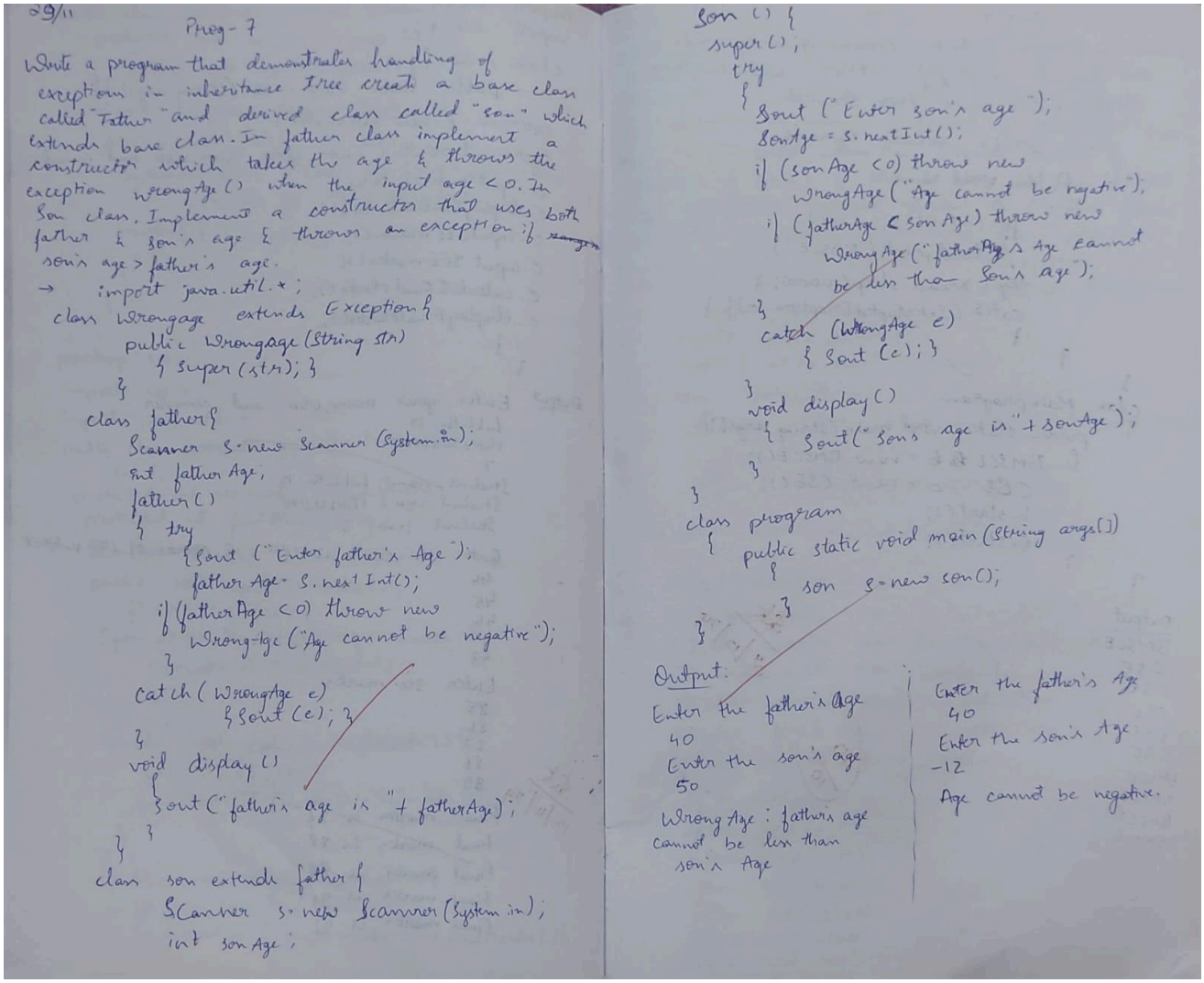
```
import see.*;
import cie.*;
class main{
  public static void main(String args[]){
externals e=new externals();
System.out.println(" enter your name , usn and semester");
e.inputStudentDetails();
e.displayStudentDetails();
e.inputCIEmarks();
e.inputSEEmarks();
e.calculateFinalMarks();
e.displayFinalMarks();
}
}
```

```
D:\1BM23CS170\OOJ>javac cie/internals.java

D:\1BM23CS170\OOJ>javac see/externals.java

D:\1BM23CS170\OOJ>javac main.java

D:\1BM23CS170\OOJ>java main
 enter your name , usn and semester
Likhith D
1BM23CS170
3
student name: 1BM23CS170
student usn: Likhith D
student semester: 3
enter the CONSOLIDATED CIE marks  of 1th  subject
44
enter the CONSOLIDATED CIE marks  of 2th  subject
45
enter the CONSOLIDATED CIE marks  of 3th  subject
46
enter the CONSOLIDATED CIE marks  of 4th  subject
47
enter the CONSOLIDATED CIE marks  of 5th  subject
48
Enter see marks
85
Enter see marks
86
Enter see marks
87
Enter see marks
88
Enter see marks
89
Final marks is86
Final marks is88
Final marks is89
Final marks is91
Final marks is92
```

PROGRAM 7

User Defined EXCEPTIONS

Algorithm :



Prog - 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 base class. In father class implement a constructor which takes the age & throws the exception wrongAge() when the input age < 0. In Son class, Implement a constructor that uses both father & son's age & throws an exception if son's age > father's age.

```
→ import java.util.*;
class Wrongage extends Exception{
    public Wrongage(String str)
    { super(str); }
}
class father{
    Scanner s=new Scanner(System.in);
    int fatherAge;
    father()
    { try
      {System.out("Enter father's Age");
       fatherAge=s.nextInt();
      if(fatherAge<0) throw new
        Wrong-Ige("Age cannot be negative");
      }
    catch(WrongAge e)
        {System.out(e);}
    }
    void display()
      {
       System.out("father's age is "+fatherAge);
      }
}
class son extends father{
    Scanner s=new Scanner(System.in);
    int sonAge;
son(){
 super();
  try
   {
    System.out("Enter son's age");
    sonAge=s.nextInt();
    if(sonAge<0) throw new
      WrongAge("Age cannot be negative");
    if(fatherAge<sonAge) throw new
        WrongAge("father's Age cannot
        be less than Son's Age");
   }
   catch(WrongAge e)
     {System.out(e);}
  }
  void display()
   {System.out("son's age is "+sonAge);
   }
}
class program
 {public static void main(String args[])
   {
    son s=new son();
   }
 }
```

Output:

```
Enter the father's Age
40
Enter the son's age
50
WrongAge : father's age
cannot be less than
son's Age
```

```
Enter the father's Age
40
Enter the son's Age
-12
Age cannot be negative.
```

Code :

```
import java.util.*;
class wrongAge extends Exception
{
        public wrongAge(String str){
super(str);
}

}
class father
{
```

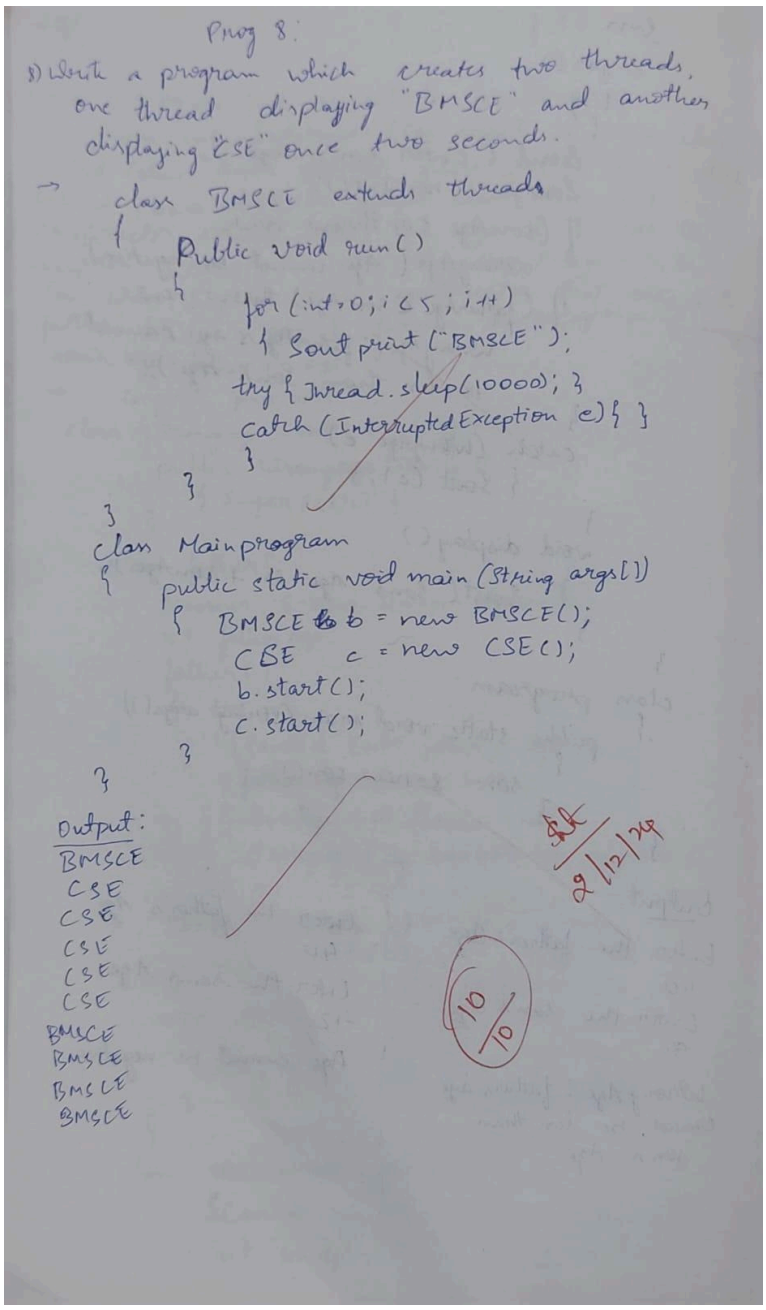
```
	Scanner s=new Scanner(System.in);
	int fatherAge;
	father()
	{
		try
		{
			System.out.println("enter the father's age");
			fatherAge=s.nextInt();
			if(fatherAge<0) throw new wrongAge("Age cannot be negative");
		}

	catch(wrongAge e){System.out.println(e);}
	}
	void display()
	{
		System.out.println("father's age is "+fatherAge);
	}
}
class son extends father
{
	Scanner s=new Scanner(System.in);
	int sonAge;
	son()
	{
		super();
		try
		{
			System.out.println("enter the son's age");
			sonAge=s.nextInt();
			if(sonAge<0) throw new wrongAge("Age cannot be negative");
			if(fatherAge<sonAge)
			throw new wrongAge("father's age cannot be less than son's age");
		}
		catch(wrongAge e){System.out.println(e);}
	}
	void display()
	{
		System.out.println("son's age is "+sonAge);
	}
}
class main
{
	public static void main(String args[])
	{
		son so=new son();
	}
}
```

```
PS C:\BMSCE> cd ooj
PS C:\BMSCE\ooj> javac main.java
PS C:\BMSCE\ooj> java main
enter the father's age
55
enter the son's age
21
PS C:\BMSCE\ooj> javac main.java
PS C:\BMSCE\ooj> java main
enter the father's age
22
enter the son's age
55
wrongAge: father's age cannot be less than son's age
PS C:\BMSCE\ooj> java main
enter the father's age
22
enter the son's age
-1
wrongAge: Age cannot be negative
```

Program 8

Multi threading

Algorithm:



Prog 8:

8) Write a program which creates two threads, one thread displaying "BMSCE" and another displaying "CSE" once two seconds.

→

```
class BMSCE extends Threads
{
    Public void run()
    {
        for (int=0; i<5; i++)
        { Sout print ("BMSCE");
        try { Thread.sleep(10000); }
        catch (InterruptedException e){ }
        }
    }
}
class Mainprogram
{
    public static void main (String args[])
    { BMSCE b = new BMSCE();
      CBE   c = new CSE();
      b.start();
      c.start();
    }
}
```

Output:

```
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
BMSCE
BMSCE
BMSCE
```

29/12/24

10/10

Code:

```
class BMSCE extends Thread
{
        public void run()
```

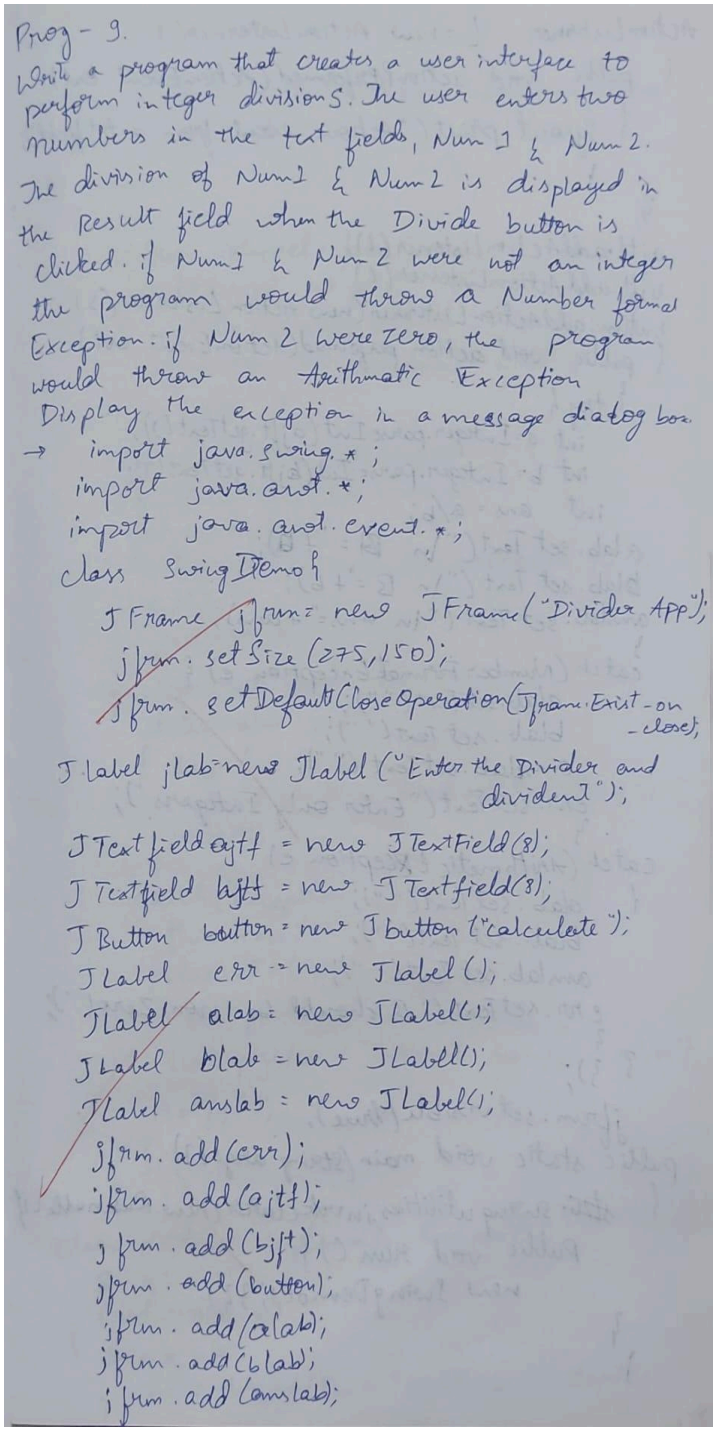
```
        {
                for(int i=0;i<5;i++)
                {
                        System.out.println("BMSCE");
                        try{Thread.sleep(10000);}
                        catch(InterruptedException e){}
                }
        }
}
class CSE extends Thread
{
        public void run()
        {
                for(int i=0;i<5;i++)
                {
                        System.out.println("CSE");
                        try{Thread.sleep(2000);}
                        catch(InterruptedException e){}
                }
        }
}
class mainProgram
{
        public static void main(String args[])
        {
                BMSCE b=new BMSCE();
                CSE c=new CSE();
                b.start();
                c.start();
                try{Thread.sleep(50000);}
                        catch(InterruptedException e){}
                System.out.println("Name: M S Shailesh");
                System.out.println("USN: 1BM23CS172");
        }
}
```

```
PS D:\1BM23CS170\OOJ> javac mainProgram.java
PS D:\1BM23CS170\OOJ> java mainProgram
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
BMSCE
BMSCE
BMSCE
Name: LIKHITH D
USN: 1BM23CS170
```

Program 9

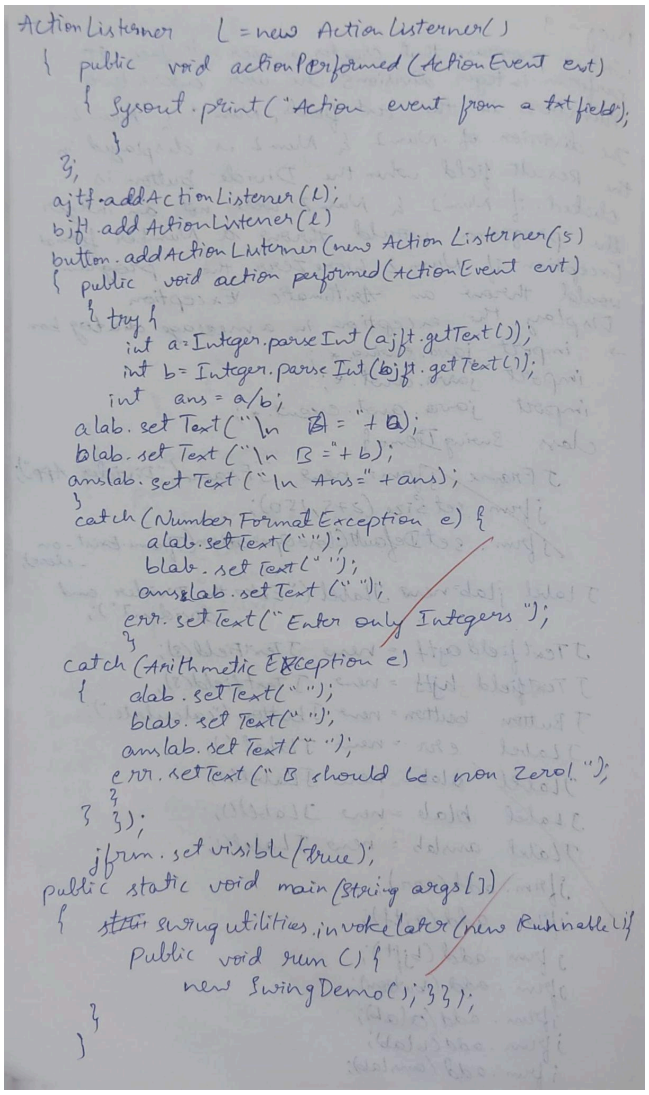
User Interface of Division

Algorithm:

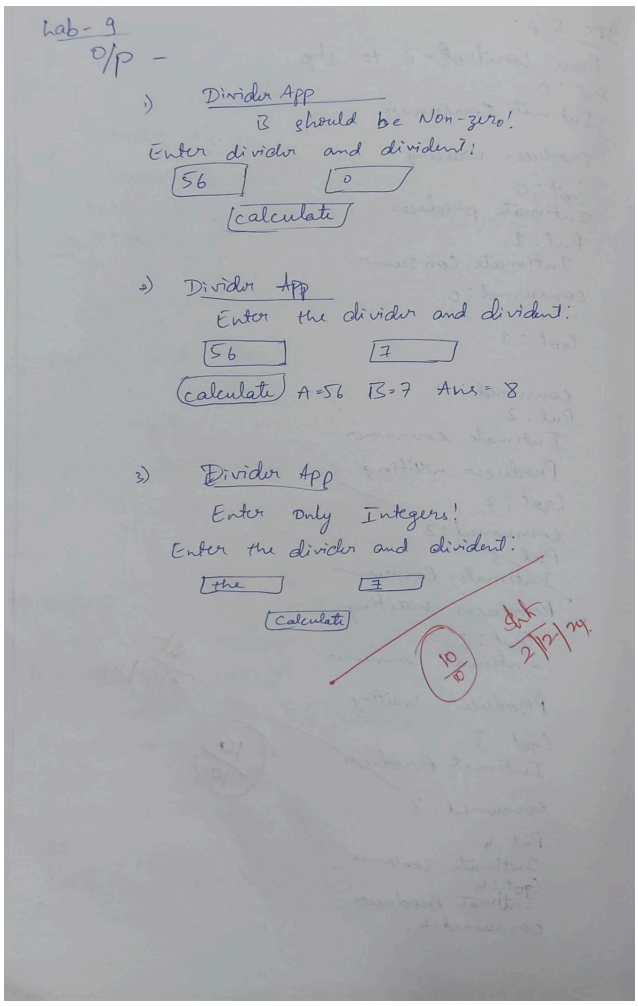


Prog - 9.
Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 & Num2. The division of Num1 & Num2 is displayed in the Result field when the Divide button is clicked. If Num1 & Num2 were not an integer the program would throw a Number format Exception. If Num2 were zero, the program would throw an Arithmatic Exception Display the exception in a message dialog box.

```
→ import java.swing.*;
  import java.awt.*;
  import java.awt.event.*;
  class SwingDemo{
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275,150);
      jfrm.setDefaultCloseOperation(JFrame.Exist-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 JLabell();
  JLabel anslab = new JLabel();
  jfrm.add(err);
  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.print("Action event from a txtfield");
  }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l)
button.addActionListener(new ActionListener(s)
{ public void action performed(ActionEvent evt)
  { try {
      int a=Integer.parseInt(ajtf.getText());
      int b=Integer.parseInt(bjtf.getText());
      int ans = a/b;
    alab.setText("\n A = "+a);
    blab.setText("\n B ="+b);
    anslab.setText("\n Ans="+ans);
    }
    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[])
{ static SwingUtilities.invokeLater(new Runnable(){
    Public void run(){
        new SwingDemo();}});
  }
}
```



lab - 9

O/P -

1) Divider App

B should be Non-zero!

Enter divider and divident:

56 0

calculate

2) Divider App

Enter the divider and divident:

56 7

calculate A=56 B=7 Ans = 8

3) Divider App

Enter only Integers!

Enter the divider and divident:

the 7

Calculate

10/10

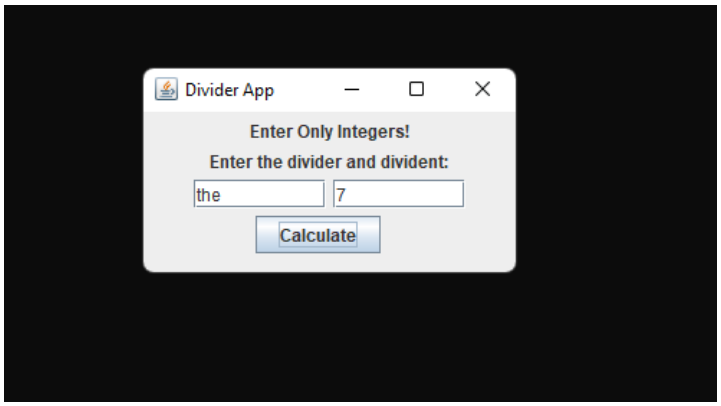
Sjt 2/12/24

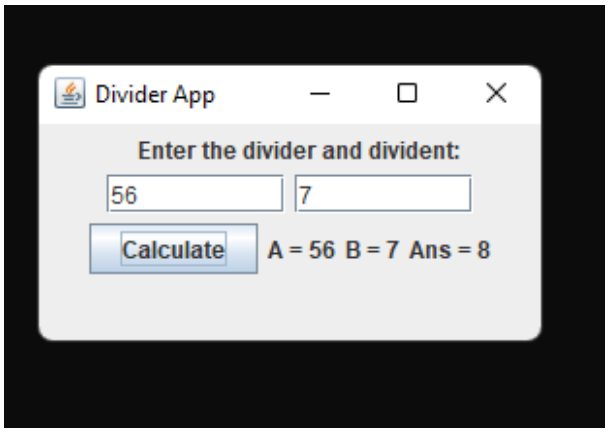
Code:

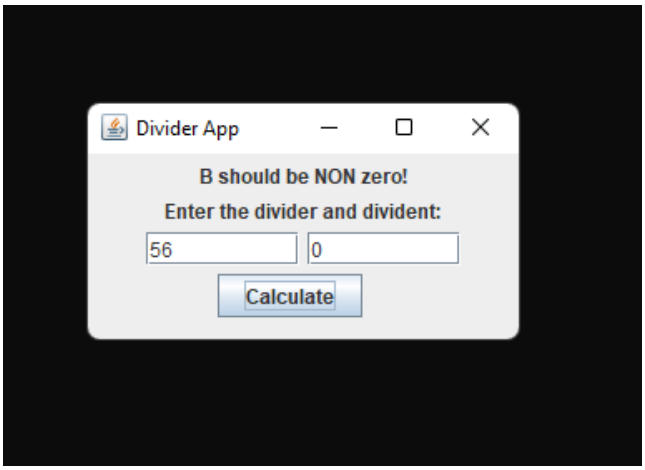
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo{
SwingDemo(){
// create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
// to terminate on close
```

```
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label
        JLabel jlab = new JLabel("Enter the divider and divident:");
// add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
// calc button
        JButton button = new JButton("Calculate");
// labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
// add in order :)
        jfrm.add(err); // to display error bois
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);
ActionListener l = new ActionListener() {
public void actionPerformed(ActionEvent evt) {
System.out.println("Action event from a text field"); }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent evt) { try{
int a = Integer.parseInt(ajtf.getText());
int b = Integer.parseInt(bjtf.getText());
int ans = a/b;
alab.setText("\nA = " + a);
blab.setText("\nB = " + b);
anslab.setText("\nAns = "+ ans);
}
catch(NumberFormatException e){
alab.setText("");
blab.setText("");
anslab.setText("");
err.setText("Enter Only Integers!"); }

catch(ArithmeticException e){
alab.setText("");
blab.setText("");
anslab.setText("");
```

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



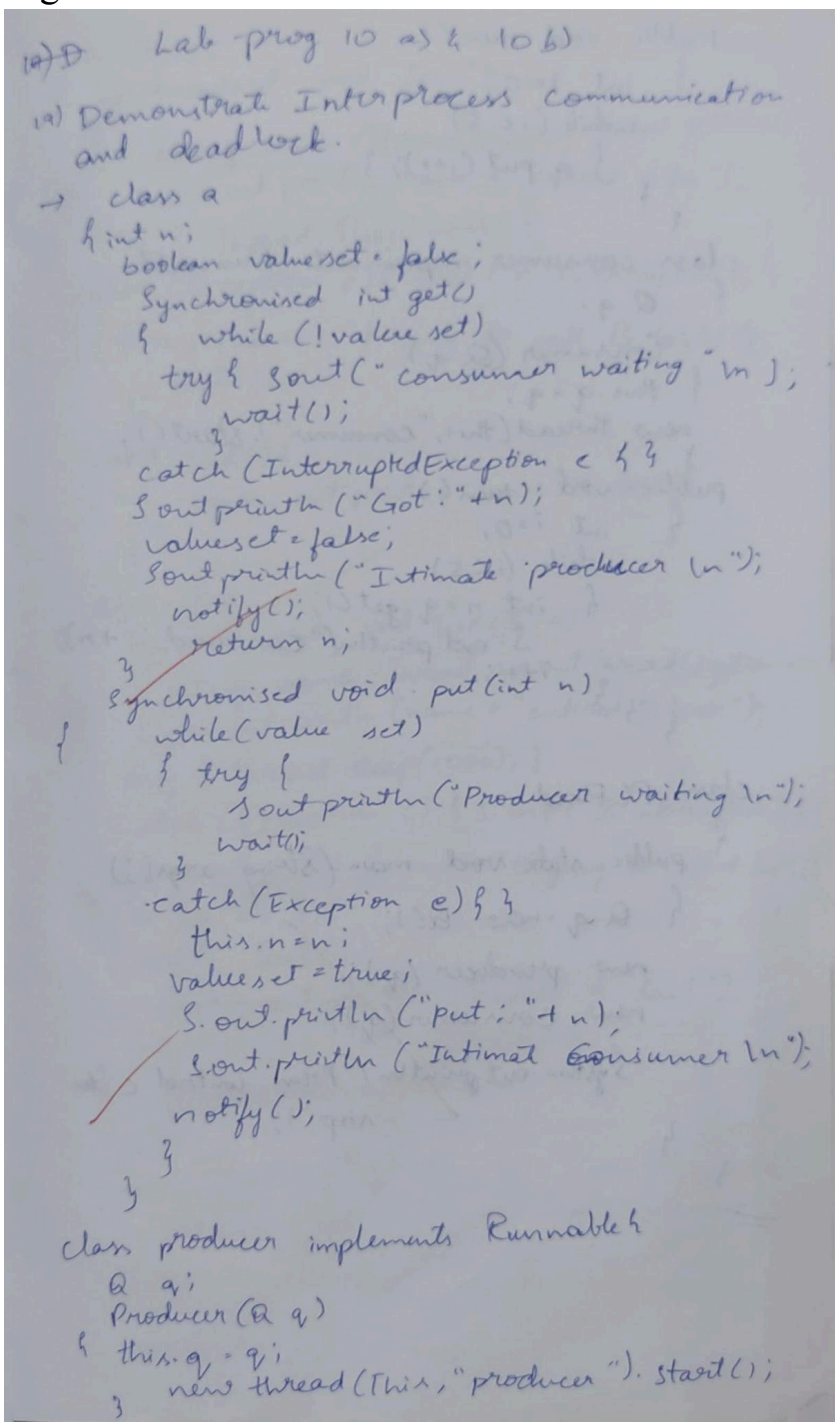




Program 10

Demonstrating Deadlock and IPC

Algorithm:



```
(a)b  Lab prog 10 a) & 10 b)

10a) Demonstrate Interprocess communication
     and deadlock.

→ class Q
  { int n;
    boolean valueset = false;
    Synchronised int get()
    { while (!valueset)
      try { System.out.println("consumer waiting \n");
            wait();
          }
      catch (InterruptedException e) { }
      System.out.println("Got: "+n);
      valueset = false;
      System.out.println("Intimate producer \n");
      notify();
      return n;
    }
  Synchronised void put(int n)
{   while(valueset)
    { try {
        System.out.println("Producer waiting \n");
        wait();
      }
     catch(Exception e) { }
      this.n = n;
      valueset = true;
      System.out.println("put : "+n);
      System.out.println("Intimat Consumer \n");
      notify();
    }
  }
class producer implements Runnable {
  Q q;
  Producer(Q q)
{ this.q = q;
    new Thread(this, "producer").start();
}
```

```
Public void run()
  { int i=0;
    while (i<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 n=q.get();
           S.out.println("consumed : "+n);
          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");
   }
}
```

```
class A
{ synchronised void foo(B b)
  { String name=Thread.currentThread.getName();
    System.out.println(name + "entered A.foo");
   try { Thread.Sleep(1000); }
catch(Exception e){ S.o.p("A Interrupted"); }
   S.o.P(name+"trying to call B.last()");
    b.last();
   }
  void last(){
      System.out.println("Inside A last");
  }}
class B{
  Synchronised void bar(A a)
  { String name=Thread.currentThread().getName();
   System.out.println(name+"entered B.bar");
  try { Thread.sleep(1000); }
  catch(Exception e) { S.o.P("B interrupted"); }
 System.out.println(name+"trying to call A.last()");
   a.last(); }
   void last(){
    System.out.println("Inside A.last");
   }
}
```

```
class Deadlock implements Runnable
{   A a=new A();
    B b=new B();
   Deadlock()
{  Thread.current().setName("Main Thread");
   Thread t=new Thread(this, "Racing Thread");
   t.start();
   a.foo(b);
   System.out.println("Back in main thread");
  }
public void run()
  {  b.bar(a);
     System.out.println("Back in other thread");
  }
public static void main(String args[])
   { new Deadlock();
   }
}
```

→ Output:

```
Main Thread entered A.foo
RacingThread entered B.bar
Main Thread trying to call B.last()
Inside A.last
Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread.
```

IPC O/p:

```
Press control-C to stop
Put : 0
Intimate Consumer
Producer waiting
Got : 0
Intimate producer
Put : 1
Intimate Consumer
consumed : 0
Got : 1
consumed : 1
Put : 2
Intimate consumer
Producer waiting
Got : 2
consumed : 2
Put : 3
Intimate Consumer
Producer waiting
Got : 3
Intimate Consumer
Producer waiting
Got : 3
Intimate Producer
consumed : 3
Put : 4
Intimate consumer
Got : 4
Intimate Producer
consumed : 4
```

dt 2/12/24.

10/10

Code:

DEADLOCK

```
class A {
synchronized void foo(B b) {
String name =Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("A Interrupted");
}
System.out.println(name + " trying to call B.last()");
b.last();
}
void last() {
System.out.println("Inside A.last");
}
}

class B {
synchronized void bar(A a) {
String name =Thread.currentThread().getName();
System.out.println(name + " entered B.bar");
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("B Interrupted");
}
System.out.println(name + " trying to call A.last()");
a.last();
}
void last() {
System.out.println("Inside A.last");
}
}

class Deadlock implements Runnable
{
A a = new A();
B b = new B();
Deadlock() {
Thread.currentThread().setName("MainThread");
Thread t = new Thread(this,"RacingThread");
t.start();
a.foo(b); // get lock on a in this thread.
System.out.println("Back in main thread");
}
public void run() {
b.bar(a); // get lock on b in other thread.
System.out.println("Back in other thread");
```

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

O/P:

```
PS D:\1BM23CS170\OOJ> javac Deadlock.java
PS D:\1BM23CS170\OOJ> java Deadlock
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
Inside A.last
Back in other thread
MainThread trying to call B.last()
Inside A.last
Back in main thread
```

IPC

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

class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<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.");
}
}
```

```
PS D:\1BM23CS170\OOJ> javac PCFixed.java
PS D:\1BM23CS170\OOJ> java PCFixed
Press Control-C to stop.
Put: 0
Intimate Consumer

Producer waiting

Got: 0
Intimate Producer

Put: 1
Intimate Consumer

Producer waiting

consumed:0
Got: 1
Intimate Producer

consumed:1
Put: 2
Intimate Consumer

Producer waiting

Got: 2
Intimate Producer

consumed:2
Put: 3
Intimate Consumer

Producer waiting

Got: 3
Intimate Producer

consumed:3
Put: 4
Intimate Consumer

Got: 4
Intimate Producer

consumed:4
```

----------THE END----------