B.M.S COLLEGE OF ENGINEERING BENGALURU
Autonomous Institute, Affiliated to VTU

Submitted in partial fulfillment of the requirements for record of

OBJECT ORIENTED JAVA PROGRAMMING

(23CS3PCOOJ)

Submitted by:

SUMITH UN

1BM22CS297

Faculty incharge:

Dr SHRAVYA

Department of Computer Science and Engineering B.M.S
College of Engineering
Bull Temple Road, Basavanagudi, Bangalore 560 019

B.M.S COLLEGE OF ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Lab program no 1:

Develop a Java program that prints all real solutions to the quadratic equation ax 2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b 2 -4ac is negative, display a message stating that there are no real solutions

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

      Quadratic q = new Quadratic();
      q.getd();
      q.compute();
   }
}

class Quadratic
{
   int a, b, c;
   double r1, r2, d;
   void getd()
```

```
{
   Scanner s = new Scanner(System.in);
   System.out.println("Enter the coefficients of a,b,c");
   a = s.nextInt();
   b = s.nextInt();
   c = s.nextInt();
}
void compute()
{
   while(a==0)
   {
      System.out.println("Not a quadratic equation");
      System.out.println("Enter a non zero value for a:");
      Scanner s = new Scanner(System.in);
      a = s.nextInt();
   }
   d = b*b-4*a*c;
   if(d==0)
   {
      r1 = (-b)/(2*a);
      System.out.println("Roots are real and equal");
      System.out.println("Roo1 = Root2 = " + r1);
   }
```

```
        else if(d>0)
        {
            r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
            r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Roo1 = " + r1 + " Root2 = " + r2);
        }
        else if(d<0)
        {
            System.out.println("Roots are imaginary");
            r1 = (-b)/(2*a);
            r2 = Math.sqrt(-d)/(2*a);
            System.out.println("Root1 = " + r1 + " + i"+r2);
            System.out.println("Root1 = " + r1 + " - i"+r2);
        }
    }
}
```

Lab program no 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.

```
import java.util.Scanner;
class Subject{
int subjectMarks;
int credits;
int grades;
}
class Student{
Subject subject[];
String name;
String usn;
double SGPA;
Scanner s;
Student(){
subject = new Subject[9];
for(int i = 0;i<9;i++){
subject[i] = new Subject();
}
s= new Scanner(System.in);
}
void getStudentDetails(){
System.out.println("Enter your name: ");
this.name = s.nextLine();
System.out.println("Enter your usn: ");
this.usn = s.next();
 }

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

  System.out.println("Enter the credits of the "+(i+1)+" subject");
  subject[i].credits = s.nextInt();
  subject[i].grades = (subject[i].subjectMarks/10)+1;

  if(subject[i].grades >10){
   subject[i].grades = 10;
  }
  if(subject[i].grades <4){
   subject[i].grades = 0;
  }
 }
}
```

```
 }

 void computeSGPA(){
  int sum=0;
  int totalCredits = 0;
  for(int i = 0;i<9;i++){
   sum+=(subject[i].grades * subject[i].credits);
   totalCredits += subject[i].credits;
  }
  this.SGPA = (double) sum/totalCredits;
 }
}

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

  System.out.println("Name: "+s1.name);
  System.out.println("Usn: "+s1.usn);
  System.out.println("SGPA: "+s1.SGPA);


 }
```

```
}
```

Lab Program no 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.

```
import java.util.Scanner;

class Books{
 String name;
 String author;
 int price;
 int numPages;

 public 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;
}
void setName(String name){
 this.name = name;
}
void setAuthor(String author){
 this.author = author;
}
void setPrice(int price){
 this.price = price;
}
void setNumPages(int numPages){
 this.numPages = numPages;
}
String getName(){
  return name;
 }
 String getAuthor(){
  return author;
 }
 int getPrice(){
  return price;
 }
 int getPgNum(){
  return numPages;
 }
}
class Main{
 public static void main(String args[]){
  Scanner sc = new Scanner(System.in);
  int n, price, numPages;
  String name, author;
  System.out.println("Enter the number of books");
  n = sc.nextInt();
  sc.nextLine();
  Books b[]= new Books[n];
  for(int i =0;i<n;i++){
   System.out.println("Read name of the book");
   name = sc.nextLine();
   System.out.println("Read author of the book");
   author = sc.nextLine();
   System.out.println("Read the price of the book");
   price = sc.nextInt();
   System.out.println("Read pgNumbers of the book");
   numPages = sc.nextInt();
   sc.nextLine();
   System.out.println("----------------------------------------
");
   b[i] = new Books(name,author,price,numPages);
  }
  for(int i =0;i<n;i++){
   String bookDetails = b[i].toString();
   System.out.println(bookDetails);
  }
  for(int i =0;i<n;i++){
   System.out.println("Book name is "+b[i].getName());
   System.out.println("Book author is "+b[i].getAuthor());
   System.out.println("Book price is "+b[i].getPrice());
   System.out.println("Book has number of pages =
"+b[i].getPgNum()+"\n");
  }
```

```
}
}
```

Lab 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.

```
import java.util.Scanner;


class InputScanner{
 Scanner s;
 InputScanner(){
  s = new Scanner(System.in);
 }
}
abstract class Shape extends InputScanner{
 double a;
 double b;


 abstract void getInput();
 abstract void displayArea();
}

class Rectangle extends Shape{

 void getInput(){
  InputScanner is = new InputScanner();
  System.out.println("Enter the length and breadth of the rectangle
:");
  a = is.s.nextDouble();
  b = is.s.nextDouble();
 }
 void displayArea(){
  System.out.println("The area of the rectangle is :"+(a*b));
 }


}
class Triangle extends Shape{

 void getInput(){
  InputScanner is = new InputScanner();
  System.out.println("Enter the base and height of the triangle :");
  a = is.s.nextDouble();
  b = is.s.nextDouble();
 }
 void displayArea(){
  System.out.println("The area of the triangle is :"+(a*b*0.5));
 }


}
class Circle extends Shape{

 void getInput(){
  InputScanner is = new InputScanner();
  System.out.println("Enter radius of the Cirlce :");
  a = is.s.nextDouble();
 }
 void displayArea(){
  System.out.println("The area of the Circle is :"+(3.14*a*a));
 }


}
public class AbstractMain{
 public static void main(String args[]){
  System.out.println("HI");
  Rectangle rect = new Rectangle();
  rect.getInput();
  rect.displayArea();

  Triangle triangle = new Triangle();
  triangle.getInput();
  triangle.displayArea();

  Circle circle = new Circle();
  circle.getInput();
  circle.displayArea();
}
}
```

Lab 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.

```
import java.util.*;
class Account{
String name;
int accno;
String type;
double balance;
private int mins=2000;
Account(String name,int accno,String type,double balance){
this.name = name;
this.accno = accno;
this.type = type;
this.balance = balance;
if(balance<mins){
System.out.println("Insufficient balance");
}
}
void deposit(double amount){
balance +=amount;
}
 void withdraw(double amount){
  if((balance-amount)>=0 ){
   balance -=amount;
  }
  else{
   System.out.println("Insufficient balance");
   return;
  }
 }
 void display(){
  System.out.println("Name : "+name+"\n"+
       "AccountNo : "+accno+"\n"+
       "Type : "+type+"\n"+
       "balance: "+balance+"\n");
 }
}

class SavingAccount extends Account{
 private static int rate = 5;
 SavingAccount(String name,int accno,String type,double balance){
  super(name,accno,type,balance);
 }
 void balanceWithInterest(){
  balance +=balance*rate/100;
  System.out.println("balance: "+balance);
 }

}

class CurrAccount extends Account{
 private static int minBalance = 2000;
 private static int charge  = 100;



 CurrAccount(String name,int accno,String type,double balance){
  super(name,accno,type,balance);
 }
 void checkMin(){
  if(balance<minBalance){
   System.out.println("Balance is less than min balance service
charge exposed " + charge);
   balance -=charge;
   return;
  }

  System.out.println("balanc is "+balance);
 }
```

```
}



public class Main{
 public static void main(String args[]){
System.out.println("Shashidhar B M");
        System.out.println("1BM22CS257");

  Scanner s = new Scanner(System.in);
  System.out.println("Enter your name: ");
  String name = s.nextLine();

  System.out.println("Enter the account type (current or deposit)");
  String type = s.next();

  System.out.println("Enter the account number: ");
  int accno = s.nextInt();

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

  Account acc = new Account(name,accno,type,balance);
  SavingAccount sa =  new SavingAccount(name,accno,type,balance);
  CurrAccount ca = new CurrAccount(name,accno,type,balance);
  double amount;
  while(true){
   if(acc.type.equals("savings")){
    System.out.println("\n-------MENU------\n");
    System.out.println("1. Deposit \t2.Withdraw \t
3.compute interest for SavingsAccount \t 4.Display Account Details\n
5.Exit\t");

    System.out.println("Enter your choice");
    int choice = s.nextInt();
    switch(choice){
     case 1:System.out.println("Enter the deposit
amount");
      amount = s.nextDouble();
      sa.deposit(amount);
      break;
     case 2: System.out.println("Enter the withdrawl
amount ");
      amount = s.nextDouble();
      sa.withdraw(amount);
      break;
     case 3:sa.balanceWithInterest();
      break;
     case 4:System.out.println("Details: ");
      sa.display();
      break;
     case 5: return;
     default: System.out.println("Invalid choice ");
    }
   }
   else{
    System.out.println("1. Deposit \t2.Withdraw \t
3.Display Account Details\n 4.Exit\t");
    System.out.println("Enter the choice");
    int choice = s.nextInt();
    switch(choice){
     case 1:System.out.println("Enter the amount : ");
            amount = s.nextInt();
             ca.deposit(amount);
      break;
     case 2:System.out.println("Enter the amount: ");
      amount = s.nextInt();
      ca.withdraw(amount);
      ca.checkMin();
      break;
     case 3 : ca.display();
      break;
     case 4: System.exit(0);
    }

   }
  }
 }
}
```

Lab program 6-a

Write a Java program to create a generic class Stack which hold 5 integers and 5 double values

and

String method demonstrations

```
import java.util.ArrayList;
import java.util.List;

class Stack<T> {
    private List<T> elements = new ArrayList<>();
    private int maxSize;
```

```
public Stack(int maxSize) {
    this.maxSize = maxSize;
}

public void push(T element) {
    if (elements.size() < maxSize) {
        elements.add(element);
        System.out.println("Pushed: " + element);
    } else {
        System.out.println("Stack is full. Cannot push more elements.");
    }
}

public T pop() {
    if (!elements.isEmpty()) {
        T poppedElement = elements.remove(elements.size() - 1);
        System.out.println("Popped: " + poppedElement);
        return poppedElement;
    } else {
        System.out.println("Stack is empty. Cannot pop elements.");
        return null;
    }
}
```

```
}

public class Main {
    public static void main(String[] args) {
        // Creating a stack for integers
        Stack<Integer> intStack = new Stack<>(5);
        intStack.push(1);
        intStack.push(2);
        intStack.push(3);
        intStack.pop();
        intStack.push(4);
        intStack.push(5);
        intStack.push(6);

        Stack<Double> doubleStack = new Stack<>(5);
        doubleStack.push(1.1);
        doubleStack.push(2.2);
        doubleStack.push(3.3);
        doubleStack.pop();
        doubleStack.push(4.4);
        doubleStack.push(5.5);
        doubleStack.push(6.6);
```

```
   }
}
```

Strings:

```
import java.util.*;
class StringConstructor{
 public static void main(String args[]){

  System.out.println("Question : 1");

  String s1  = new String();
  s1 = "newString";
  char ch[]={'a','b','c','d'};
  String s2 = new String(ch);
  String demo = "Hello";
  String s = new String(demo);

  String charString = new String(ch,1,2);

  System.out.println(s1);
  System.out.println(s2);
  System.out.println(s);
  System.out.println(charString);


  System.out.println("Question : 2");


  String name ="shashidhar";
  String lname  = " B M ";
  System.out.println("The first string length is "+name.length());
  name = name.concat(lname);
  System.out.println(name)
  System.out.println("Question : 3");


  Integer num = 9807;
  String snum = num.toString();
  System.out.println(snum);


  System.out.println("Question : 4");


  String extract = "Welcome to bmsce college";
  char chs[] = new char[20];
  extract.getChars(10,16,chs,0);
  String ans = new String(chs);
  System.out.println(ans);
```

```
 System.out.println("Question : 5");

 String myName = "shashihdar";
 char charArray[] = myName.toCharArray();
 for(char val: charArray){
  System.out.print(val+"\t");
 }
 System.out.println();
 byte arr[] = myName.getBytes();
 for(byte val: arr){
  System.out.print(val+"\t");
 }
 System.out.println();

 System.out.println("Question : 6");

 System.out.println("Bmsce".equals("Bmsce"));
 System.out.println("Bmsce".equals("College"));
 System.out.println("Bmsce".equals("BMSCE"));
 System.out.println("Bmsce".equalsIgnoreCase("BMSCE"));


}
```

```
}
class Sorts{
 public static void main(String args[]){
  System.out.println("b".compareTo("a"));
  String st[] = {"van", "watch","ball", "cat","xmas","yatch"," zee",

 "apple","ice","jug","kite","lift","man","net","orange","dog","ent","free"
 ,"gun","hen","parrot","queen","ring","star","tree","umbrella"};
  for(int i =0;i<st.length();i++){
   for(int j =0;j<st.length();j++){
    if(st[i].compareTo(st[j])==1){
     String temp = st[i];
     st[i] = st[j];
     st[j] = temp;
    }
   }
  }
  for(String c:st){
   System.out.println(c);
  }
 }
}
import java.util.*;
```

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

  String subString = "Bmsce collge";
  String  val = "Welcome to Bmsce College of Engineering";
if(val.regionMatches(11,subString,0,5)){
System.out.println("String matches");
}
else{
}
System.out.println("String not matches");
System.out.println(subString.startsWith("B"));
System.out.println(subString.startsWith("r"));
String a = "Hello";
String b = "Hello";
String c = new String("Hello");
System.out.println(a==b);
System.out.println(b==c);
System.out.println(b.equals(c));
}
}
```

Lab program 6 -b packages

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 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.

CIE/Student.java

```
package CIE;

import java.util.Scanner;

public class Student {


    protected String usn = new String();

    protected String name = new String();

    protected int sem;


    public void inputStudentDetails() {

        System.out.println("Enter the usn , name , sem :\n");

        Scanner sc = new Scanner(System.in);
        this.usn = sc.nextLine();
        name = sc.nextLine();
        sem = sc.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("Student details are \n Name : name \n Usn : usn
\n
Sem : sem \n");
    }
}
```

CIE/Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends Student {
    protected int marks[] = new int[5];
    public void inputCIEmarks(){

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the marks of 5 subject");
        for(int i =0;i<5;i++){
            marks[i] = sc.nextInt();
        }
    }
}
```

SEE/Externals.java

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

public class Externals extends Internals {
    protected int marks[];
    protected int finalMarks[];
    public Externals() {
        marks = new int[5]; finalMarks = new int[5];
    }
        public void inputSEEmarks() {
            Scanner s = new Scanner(System.in);
            for(int i=0;i<5;i++) {
                System.out.print("Subject "+(i+1)+" marks: ");
                marks[i] = s.nextInt();
            }
        }
    public void calculateFinalMarks() {

        for(int i=0;i<5;i++){
            finalMarks[i] = marks[i]/2 + super.marks[i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        for(int i=0;i<5;i++){
            System.out.println("Subject " + (i+1) + ": " + finalMarks[i]);
        }
    }

}
Main.java import SEE.Externals;
import CIE.*;

class Main {

    public static void main(String args[]){
        int numOfStudents = 2;
        Externals finalMarks[] = new Externals[numOfStudents];
        for(int i=0;i<numOfStudents;i++){
            finalMarks[i] = new Externals();
            finalMarks[i].inputStudentDetails();

            System.out.println("Enter CIE marks");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE marks");
            finalMarks[i].inputSEEmarks();
        }
        System.out.println("Displaying data:\n");

        for(int i=0;i<numOfStudents;i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

Lab program 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class

called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age is >=father’s age.

```
import java.util.*;
class WrongAge extends Exception{
    public WrongAge(String s){
        super(s);
    }
}
class InputScanner{
    Scanner sc;
    public InputScanner(){
        sc = new Scanner(System.in);
    }
}

class Father extends InputScanner{
    int fatherAge;
    public Father() throws WrongAge{
        InputScanner ss = new InputScanner();
        System.out.println("Enter the father age: ");
        fatherAge = ss.sc.nextInt();
        if(fatherAge<0){
            throw new WrongAge("Age cannot be negative");
        }
    }
    void fdisplay(){
        System.out.println("Father age is : "+fatherAge);
    }
}

class Son extends Father{
    int sonAge;
    public Son() throws WrongAge{
        InputScanner ss = new InputScanner();
        System.out.println("Enter the Son age: ");
        sonAge = ss.sc.nextInt();
        if(sonAge==fatherAge){
            throw new WrongAge("Son's age cannot be equal to father age");
        }
        else if(sonAge>fatherAge){
            throw new WrongAge("Son's age cannot be greater than father's age");
        }
        else if(sonAge<0){
            throw new WrongAge("Age cannot be negative");
        }
    }
    void sdisplay(){
        System.out.println("Son's age is :"+sonAge );
    }
}

public class PMain{
    public static void main(String args[]){
        Son p;
        try{
            p = new Son();
            p.fdisplay();
            p.sdisplay();
        }
        catch(WrongAge e){
            System.out.println(e);
        }
System.out.println("Shashidhar B M");
         System.out.println("1BM22CS257");
    }
}
```

Lab 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.

```
class Bms extends Thread {
    public void run() {
        for (int i = 1; i <= 50; i++) {
            try {
                Thread.sleep(10000);
                System.out.println("BMS College of Engineering" + i);
            } catch (InterruptedException e) {
                System.out.println("thread error");
            }
        }
    }
}

class Cse extends Thread {
    public void run() {
        for (int i = 1; i <= 50; i++) {
            try {
                Thread.sleep(2000);
                System.out.println("Computer Science " + i);
            } catch (InterruptedException e) {
                System.out.println("thread error");
            }
        }
    }
}

public class TreadsMain {
    public static void main(String args[]) {
        Bms c1 = new Bms();
        c1.start();

        Cse i1 = new Cse();
        i1.start();
    }
}
```

Lab 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 ArithmeticException Display the exception in a message dialog box.

```
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 :)
          // to display error bois
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(err);
        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();
            }
        });
    }
}
```

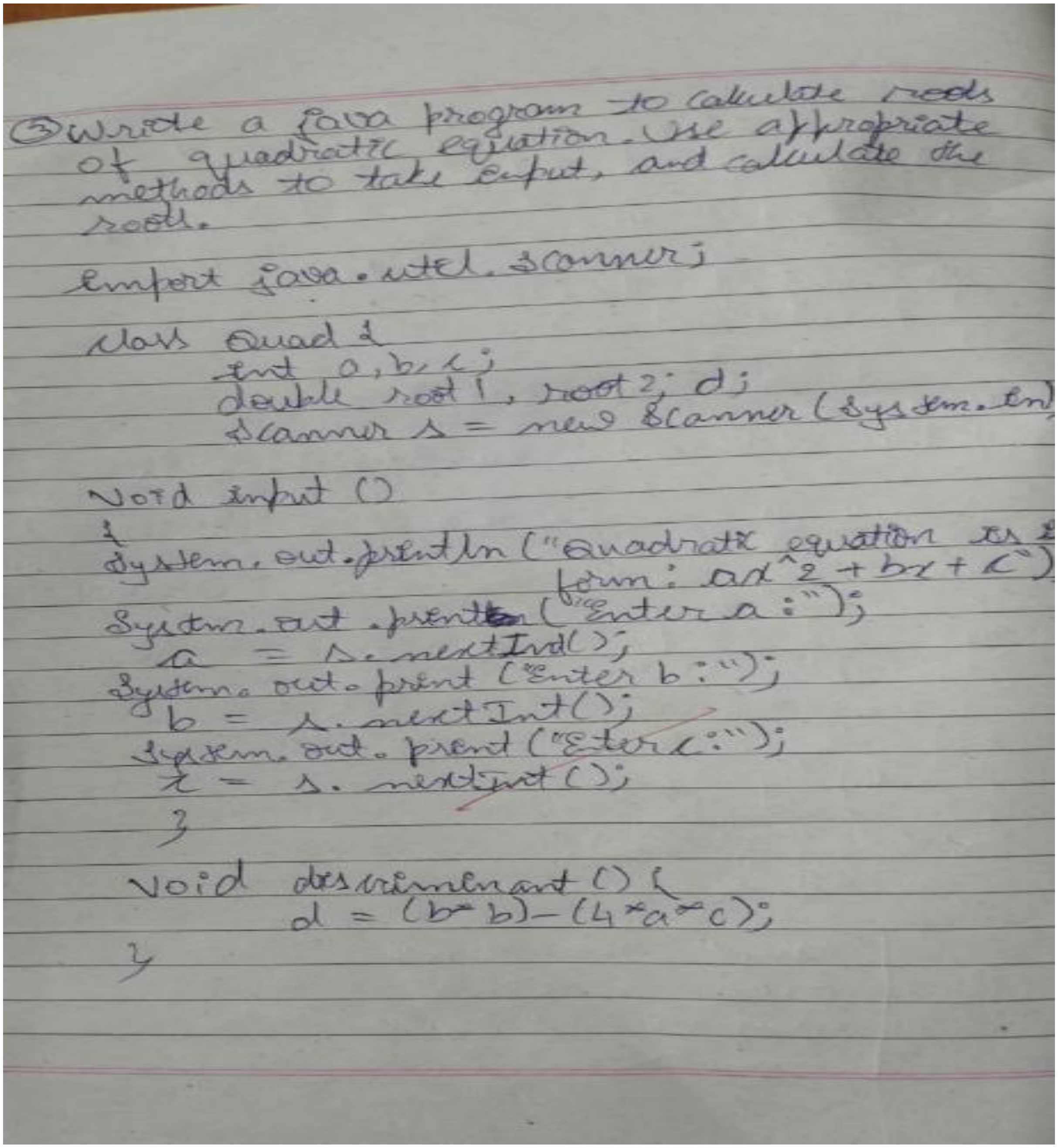
Lab program 10

Demonstrate Inter process Communication and deadlock

```
class Q {

    int n;
    boolean valueSet = false;

    synchronized int get() {
        while (!valueSet)
            try {
                System.out.println("\nConsumer waiting\n");
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedExceptioncaught");
            }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("\nIntimate Producer\n");
    notify();
    return n;

}

synchronized void put(int n) {

    while (valueSet)

        try {
            System.out.println("\nProducer waiting\n");
            wait();

        } catch (InterruptedException e) {

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

        }

    this.n = n;

    valueSet = true;
```

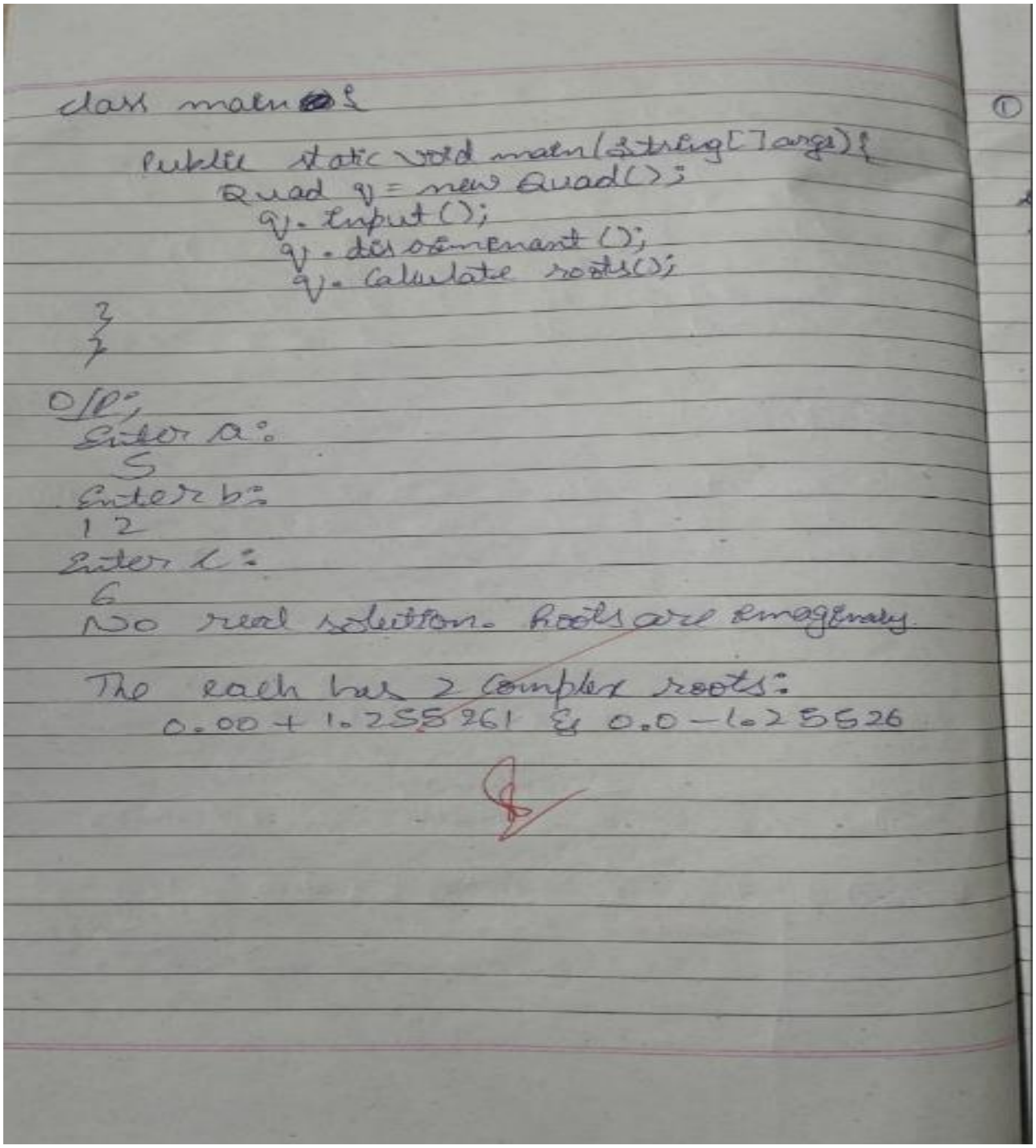
```
        System.out.println("Put: " + n);

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

        notify();

    }

}

class Producer implements Runnable {
    Q q;
    Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
    }
    public void run() {
        int i = 0;
        while (i < 15) {
            q.put(i++);
        }
    }
}
```

```
class Consumer implements Runnable {
    Q q;
    Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run() {

        int i = 0;
        while (i < 15) {
            int r = q.get();
            System.out.println("consumed:" + r);
            i++;
        }
    }
}
public class Corrected {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
   } }
```



③ Write a java program to calculate roots of quadratic equation. Use appropriate methods to take input, and calculate the roots.

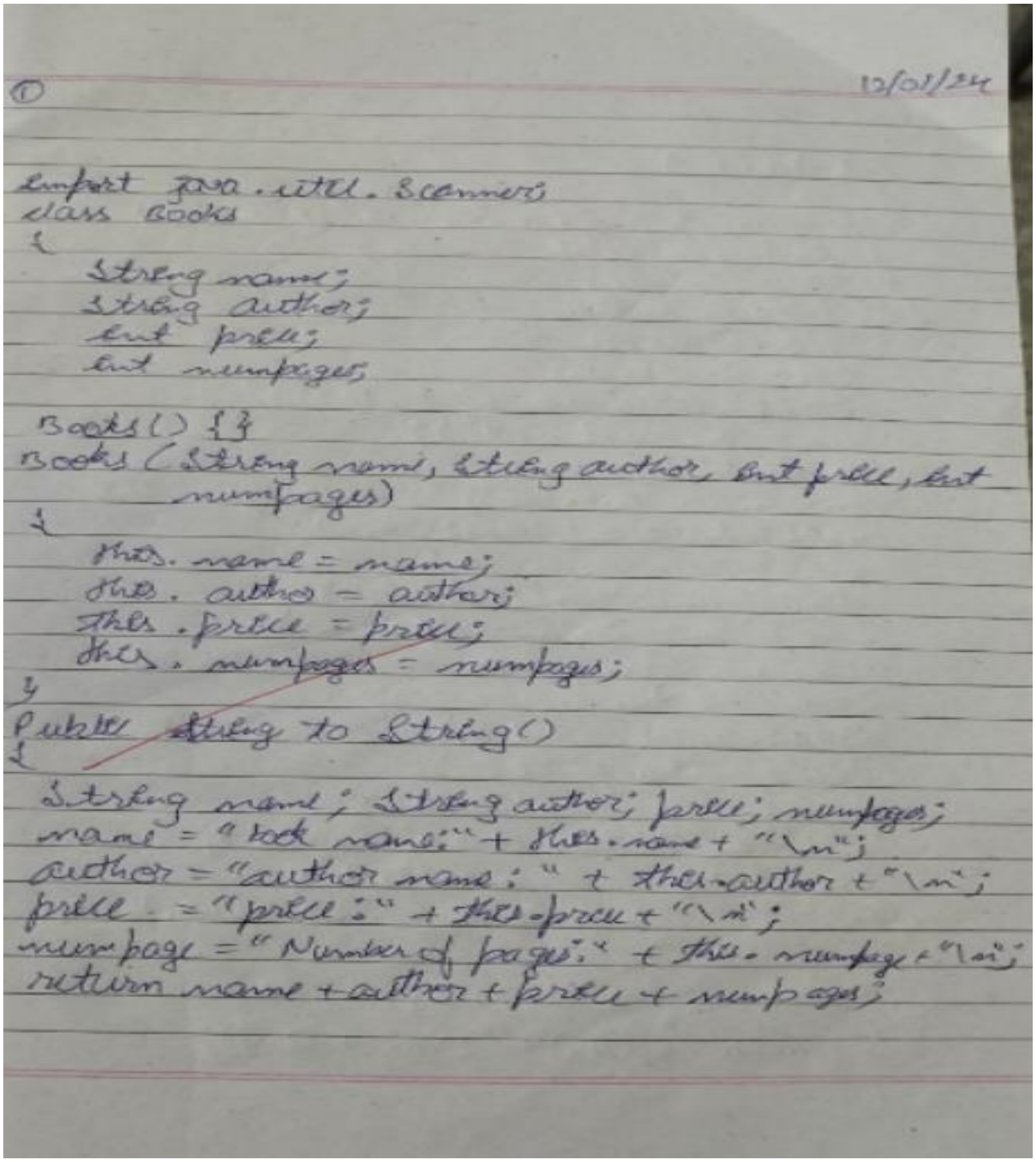
```
import java.util.Scanner;

class Quad {
        int a, b, c;
        double root1, root2; d;
        Scanner s = new Scanner(System.in)

void input ()
{
System.out.println("Quadratic equation is i
                         form: ax^2 + bx + c")
System.out.println("Enter a:");
  a = s.nextInt();
System.out.print("Enter b:");
  b = s.nextInt();
 System.out.print("Enter c:");
  c = s.nextInt();
   }

 void descrimenant() {
         d = (b*b)-(4*a*c);
 }
Void calulate Roots(){
    if (d>0)
  {
   System.out.println("Roots are real and
    unequal");
   root 1 = (-b + Math. sqrt(d)/(2*a));
   root 2 =(-b - Math. sqrt(d)/(2*a));
   System. out. println ("First root is:"+root1);
   System. out. println("second root is:"+root2);
  }

else if (d==0)
{
 System. out. println ("Roots are real and equal");
 root 1 = (-b + Math.sqrt(d)/(2*a));
 System.out. println ("Root:"+ root1);
}

else
{
System. out. println("No real solutions. Roots
                    are imaginary");
double real = -b/(2*a);
double imaginary = Math. sqrt(-d)/(2*a);

stem. out. println ("The equation has two complex roots:
                " + real + " + " + imaginary + "i and
                + real + "-" + imaginary + "i");

}
}
}
```



```
class main{
    public static void main(String[] args){
        Quad q = new Quad();
          q. input();
            q. discriminant();
              q. calculate roots();
  }
  }
```

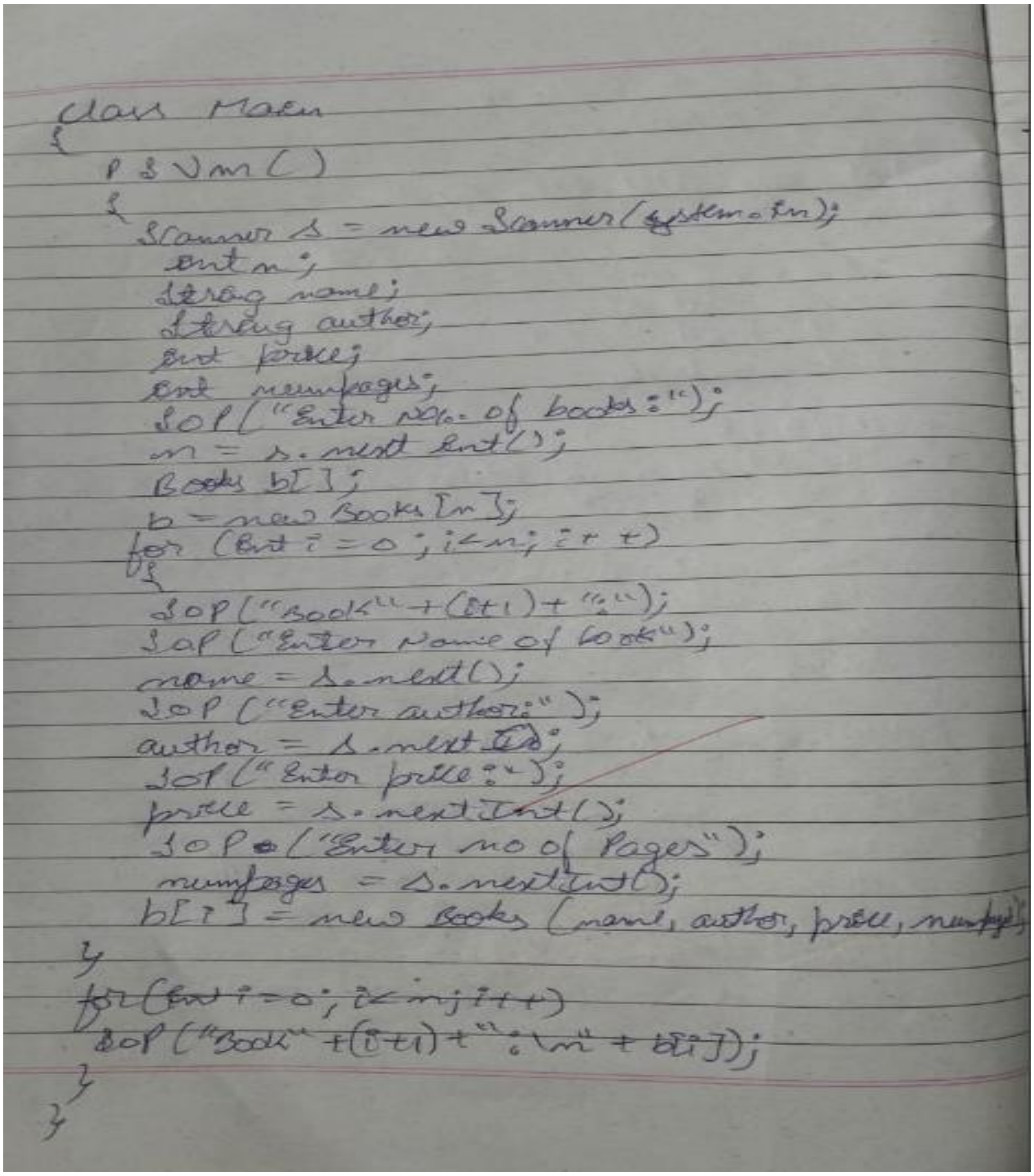
O/P:

```
Enter a:
 5
Enter b:
12
Enter c:
 6
No real solution. Roots are imaginary.

The each has 2 complex roots:
    0.00 + 1.255261 & 0.0 - 1.25526
```



① 12/01/24

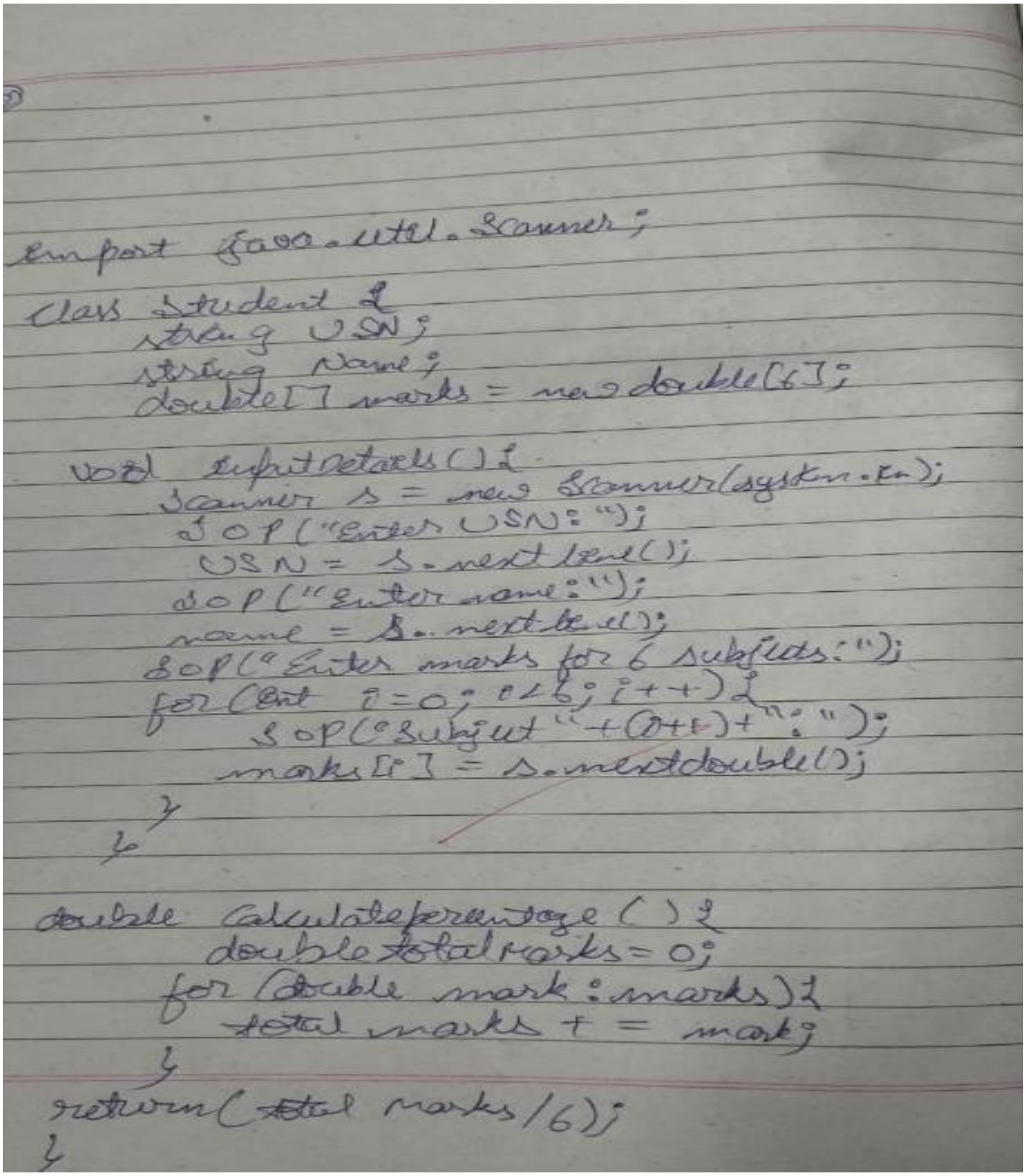
```
import java.util.Scanner;
class Books
{
    String name;
    String author;
    int price;
    int numpages;

 Books() {}
Books (String name, String author, int price, int
        numpages)
 {
    this.name = name;
    this.author = author;
    this.price = price;
    this.numpages = numpages;
}
Public String to String()
{
 String name; String 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.numpage + "\n";
 return name + author + price + numpages;
```



```
class Main
{
  psvm()
  {
   Scanner s = new Scanner(system.in);
    int n;
    String name;
    String author;
    int price;
    int numpages;
    SOP("Enter no. of books:");
    n = s.next Int();
    Books b[];
    b = new Books[n];
   for (int i = 0; i<n; i++)
   {
    SOP("Book" + (i+1) + ":");
    SOP("Enter Name of book");
    name = s.next();
    SOP("Enter author:");
    author = s.next();
    SOP("Enter price:");
    price = s.nextInt();
    SOP("Enter no of Pages");
    numpages = s.nextInt();
    b[i] = new Books(name, author, price, numpg);
  }
  for(int i=0; i<n;i++)
   SOP("Book" +(i+1) + ":\n" + b[i]);
  }
}
```

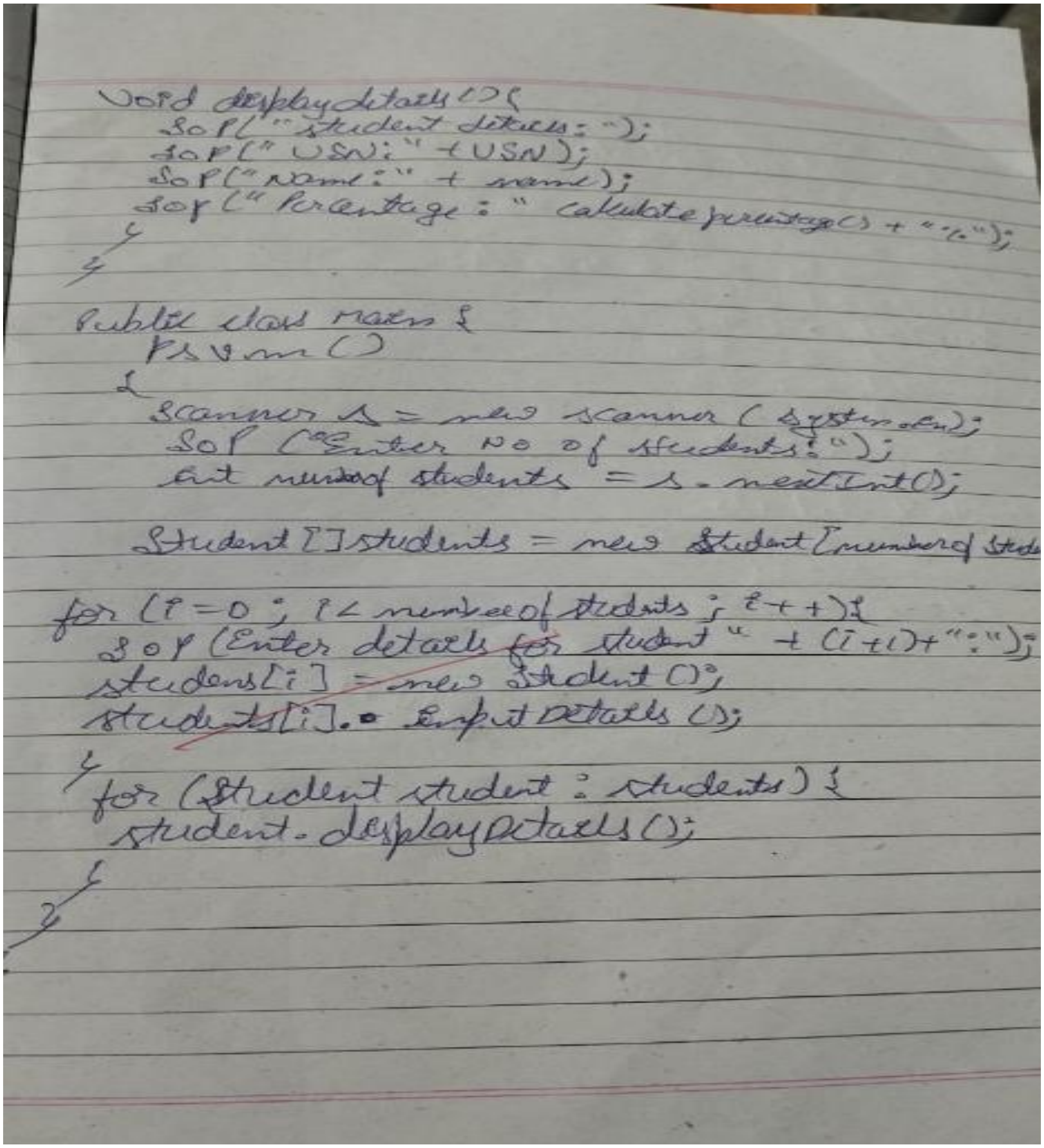
O/P:

```
Enter number of books: 2
Book1: enter the name of Book: OS
Enter author: KAMAL
Enter price: 360
Enter no of pages: 900

Book 2: enter the name of Book: Java
Enter author: Sachin
Enter price: 780
Enter No of pages: 390

Book 1:
book name: OS
authorName: KAMAL
Price :- 360
No of Pages: 900

Book 2:
book name: Java
Author name: Sachin
Price: 780
No of pages: 390
```

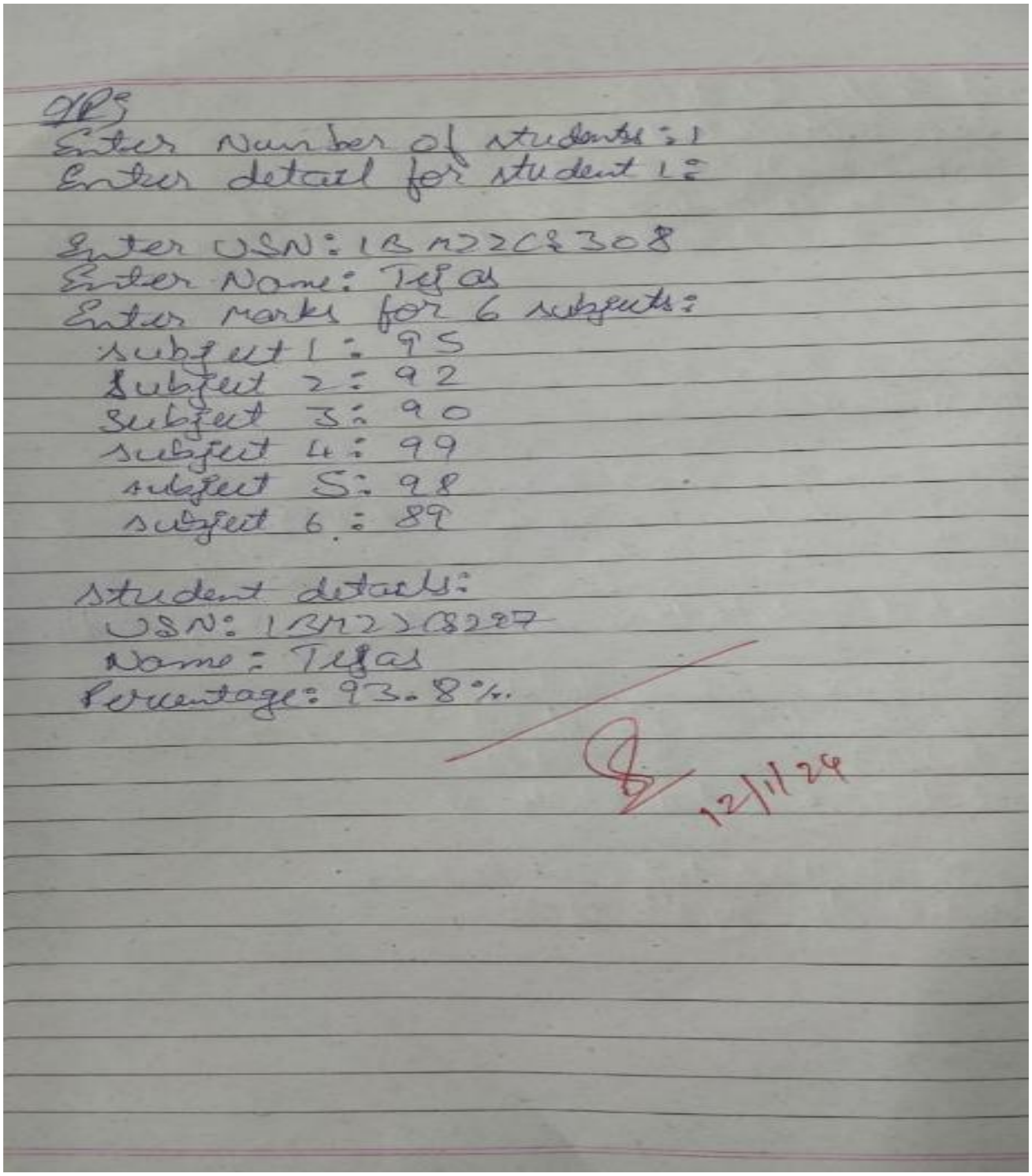


①

```
import java.util.Scanner;

class Student {
      string USN;
      string Name;
      double[] marks = new double[6];

   void inputDetails() {
       Scanner s = new Scanner(system.in);
        SOP("Enter USN:");
         USN = s.nextLine();
        SOP("Enter name:");
        name = s.nextLine();
       SOP("Enter marks for 6 subjects:");
       for(int i=0; i<6; i++){
            SOP("Subject "+(i+1)+":");
          marks[i] = s.nextdouble();
       }
   }

double calculatepercentage () {
         doubletotalmarks=0;
      for (double mark : marks){
         total marks + = mark;
      }
 return(total marks/6);
}
```



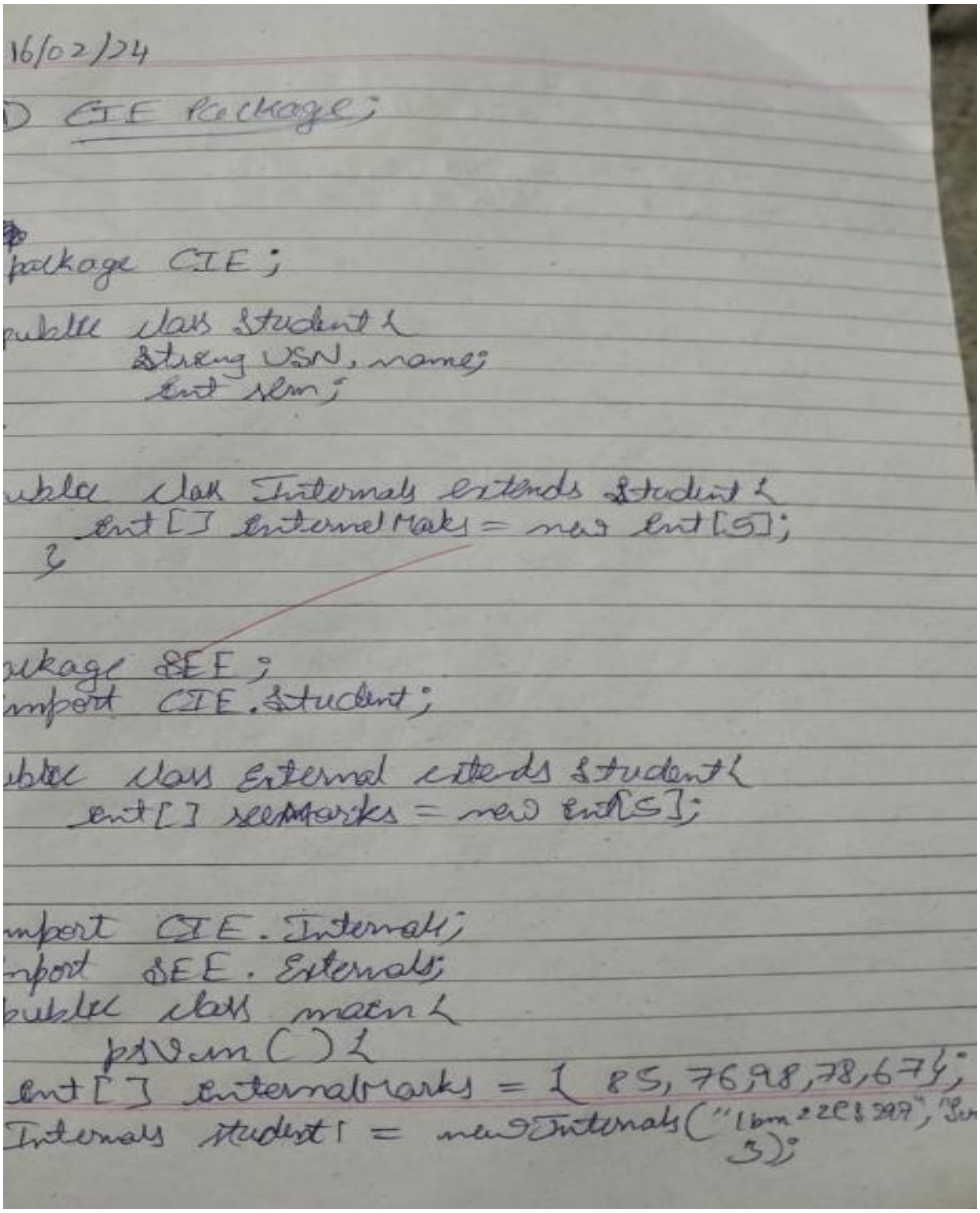
```
Void displaydetails(){
    SoP("student details:");
    SoP("USN:" +(USN);
    SoP("Name:" + name);
   SoP("Percentage:" calculatepercentage() + "%");
  }
}

Public class Main {
   PSVm()
  {
    Scanner s = new Scanner(System.in);
     SoP("Enter No of students:");
    int numberof students = s.nextInt();

   Student[]students = new Student[numberof stude

for (i=0; i< numberofstudents; i++){
  SoP(Enter details for student " +(i+1)+":");
  studens[i] = new Student();
  students[i]. inputDetails();
}
 for (Student student : students){
  student.displayDetails();
 }
}
```



O/P:

```
Enter Number of students: 1
Enter detail for student 1:

Enter USN: 1BM22CS308
Enter Name: Tejas
Enter marks for 6 subjects:
  subject 1: 95
  subject 2: 92
  subject 3: 90
  subject 4: 99
   subject 5: 98
   subject 6: 89

 student details:
  USN: 1BM22CS227
  Name: Tejas
 Percentage: 93.8%.
```

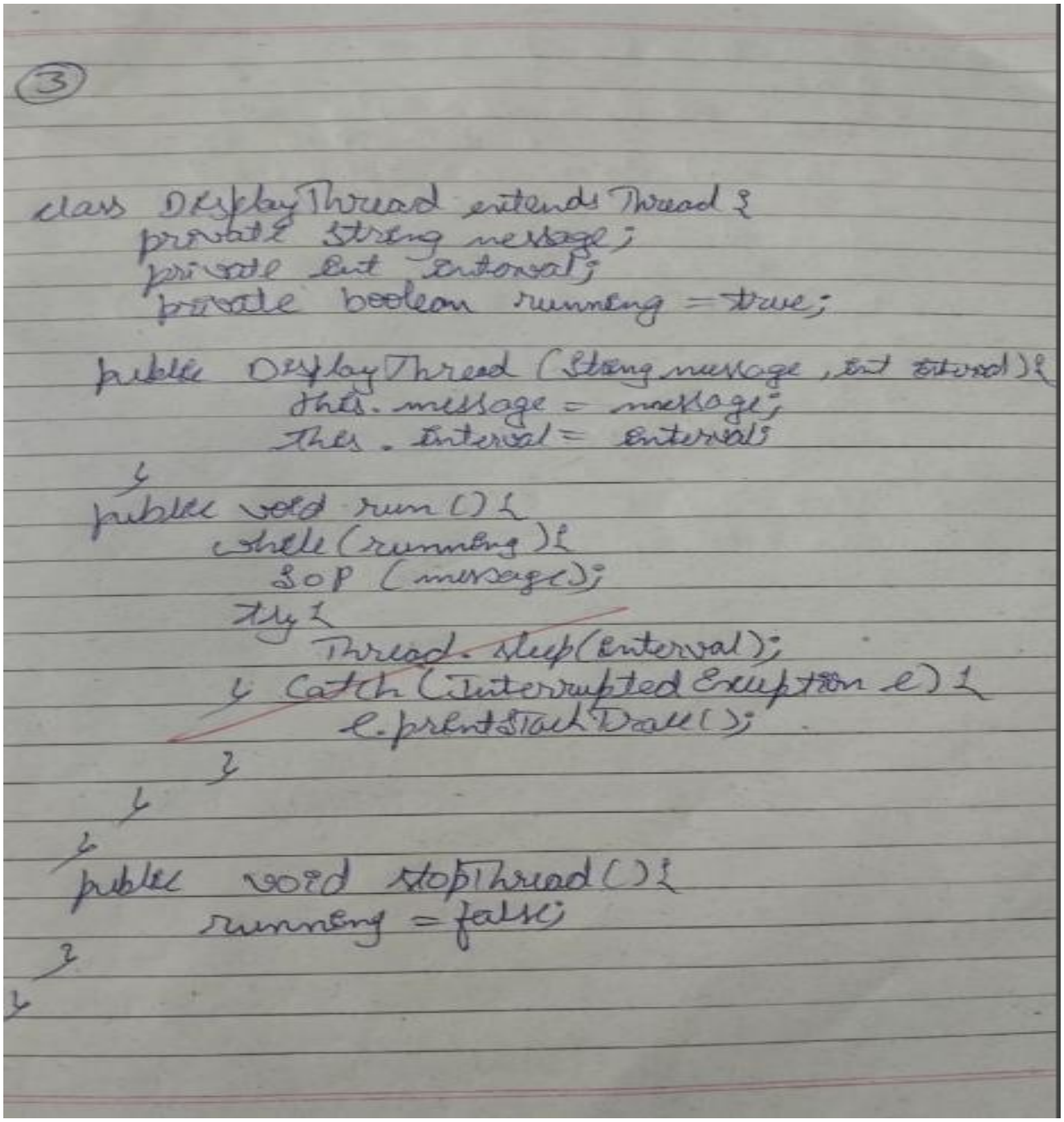
12/1/24



16/02/24

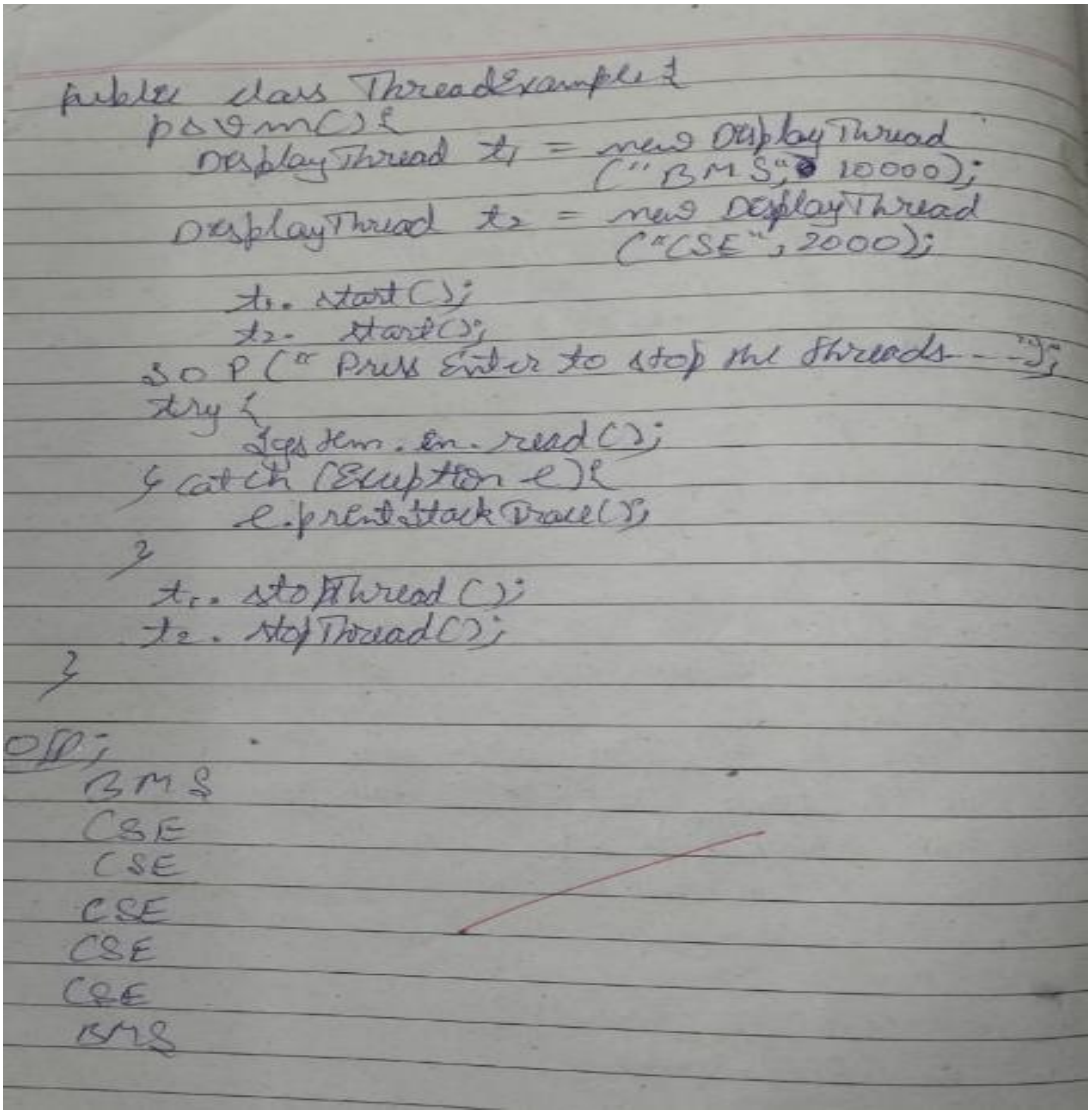
1) CIE Package;

```
package CIE;

public class Student {
        String USN, name;
        int sem;

ublic class Internals extends Student {
    int[] internalMarks = new int[5];
  }

ckage SEE;
import CIE.Student;

blic class External extends Student{
    int[] seeMarks = new int[5];

mport CIE.Internals;
nport SEE.Externals;
public class main {
        psvm () {
int[] internalMarks = { 85, 76,98,78,67};
Internals student1 = new Internals("1bm22CS297", "Su
                                        3);
```



③

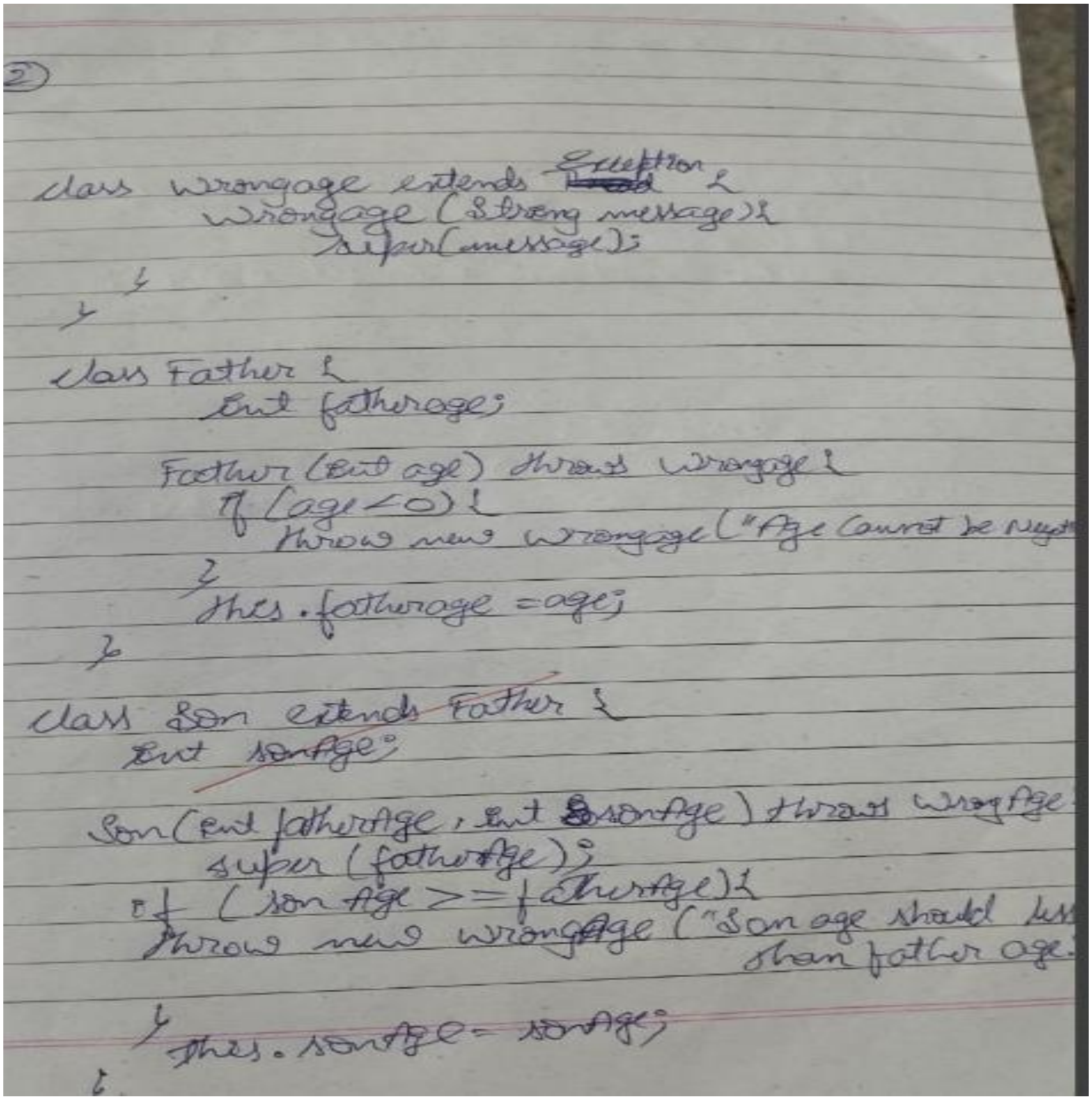
```
class DisplayThread extends Thread {
    private String message;
    private int interval;
    private boolean running = true;

  public DisplayThread (String message , int interval){
              this. message = message;
              this . interval = interval;
  }
  public void run (){
          while(running){
             SOP (message);
          try{
               Thread. sleep(interval);
          } Catch (InterruptedException e) {
                 e.printStackTrace();
          }
     }
  }
 public  void stopThread(){
       running = false;
 }
}
```



```
public class ThreadExample {
    psvm() {
      DisplayThread t1 = new DisplayThread
                          ("BMS", 10000);
      DisplayThread t2 = new DisplayThread
                          ("CSE", 2000);

         t1. start();
         t2. start();
    SOP(" Press Enter to stop the threads...");
    try {
          System. in. read();
    } catch (Exception e){
          e.printStackTrace();
    }
     t1. stopThread();
     t2. stopThread();
}
```

O/P;

```
 BMS
 CSE
 CSE
CSE
CSE
CSE
BMS
```

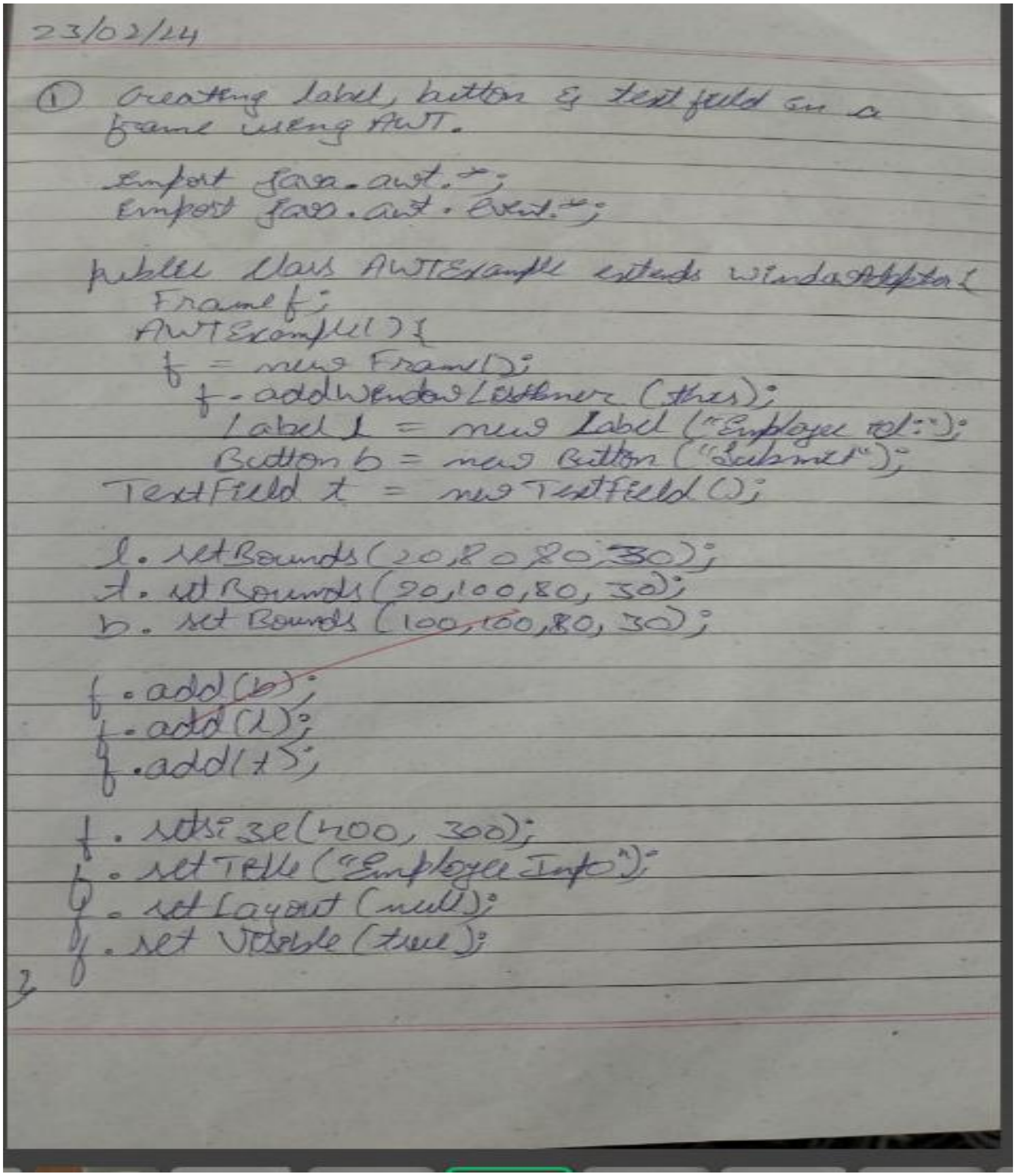


②

```
class wrongage extends Exception {
        wrongage (String message){
            super(message);
    }
}

class Father {
        int fatherage;

      Father (int age) throws wrongage {
        if (age<0){
            throw new wrongage ("Age cannot be Negat
        }
        this.fatherage =age;
  }

class Son extends Father {
    int sonAge;

  Son(int fatherAge, int sonAge) throws wrongAge
        super (fatherAge);
   if (sonAge >= fatherAge){
    throw new wrongAge ("Son age should less
                                than father age.
   }
    this.sonAge= sonAge;
}.
```

```
public class ExceptionInheritance Demo {
    psvm() {

    try {

    Father father = new Father(40);
    Son son = new Son(40,20);
    } catch (WrongAge e){
        SOP("Exception:" + e.getMessage());
    }
  }
```

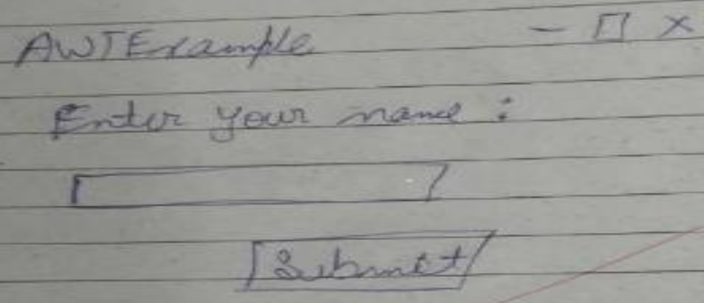
```
O/P :-
    Enter father's age : 20
    Enter Son's age : 40
  Exception Son's age cannot be greater's ag
```

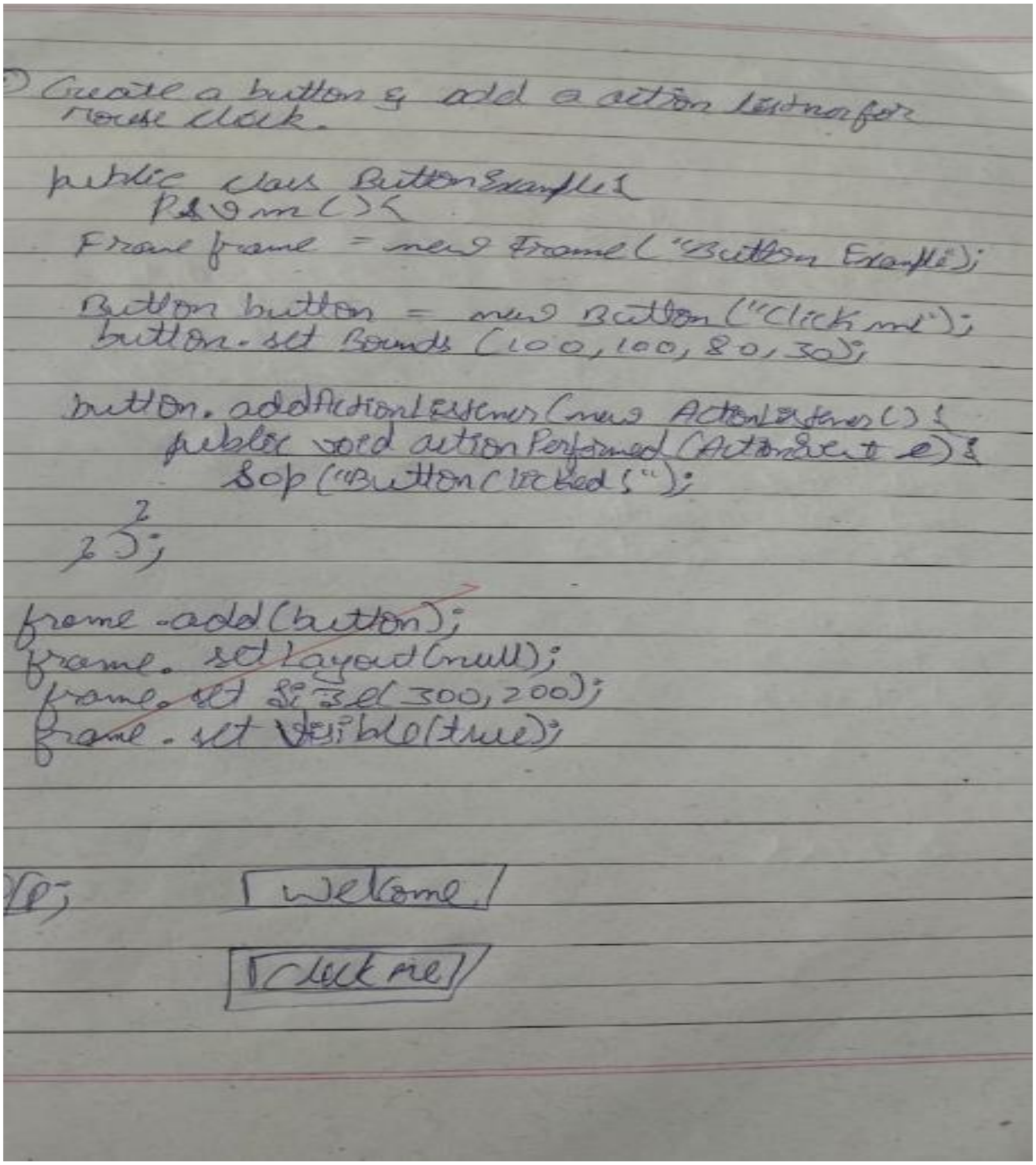


23/02/24

① Creating label, button & text field in a frame using AWT.

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

public class AWTExample extends WindowAdapter{
  Frame f;
  AWTExample(){
   f = new Frame();
    f.addWindowListener (this);
     Label l = new Label ("Employee id:");
     Button b = new Button ("Submit");
 TextField t = new TextField ();

 l. setBounds(20,80,80,30);
 t. setBounds(20,100,80, 30);
 b. set Bounds (100,100,80, 30);

 f.add(b);
 f.add(l);
 f.add(t);

 f. setSize(400, 300);
 f. setTitle ("Employee Info");
 f. setLayout (null);
 f. set Visible (true);
}
public void windowClosing (WindowEvent e){
        System.exit(0);
 }
public static void main (a ){
      AWTExample awt obj = new AWTExample();
 }
}
```



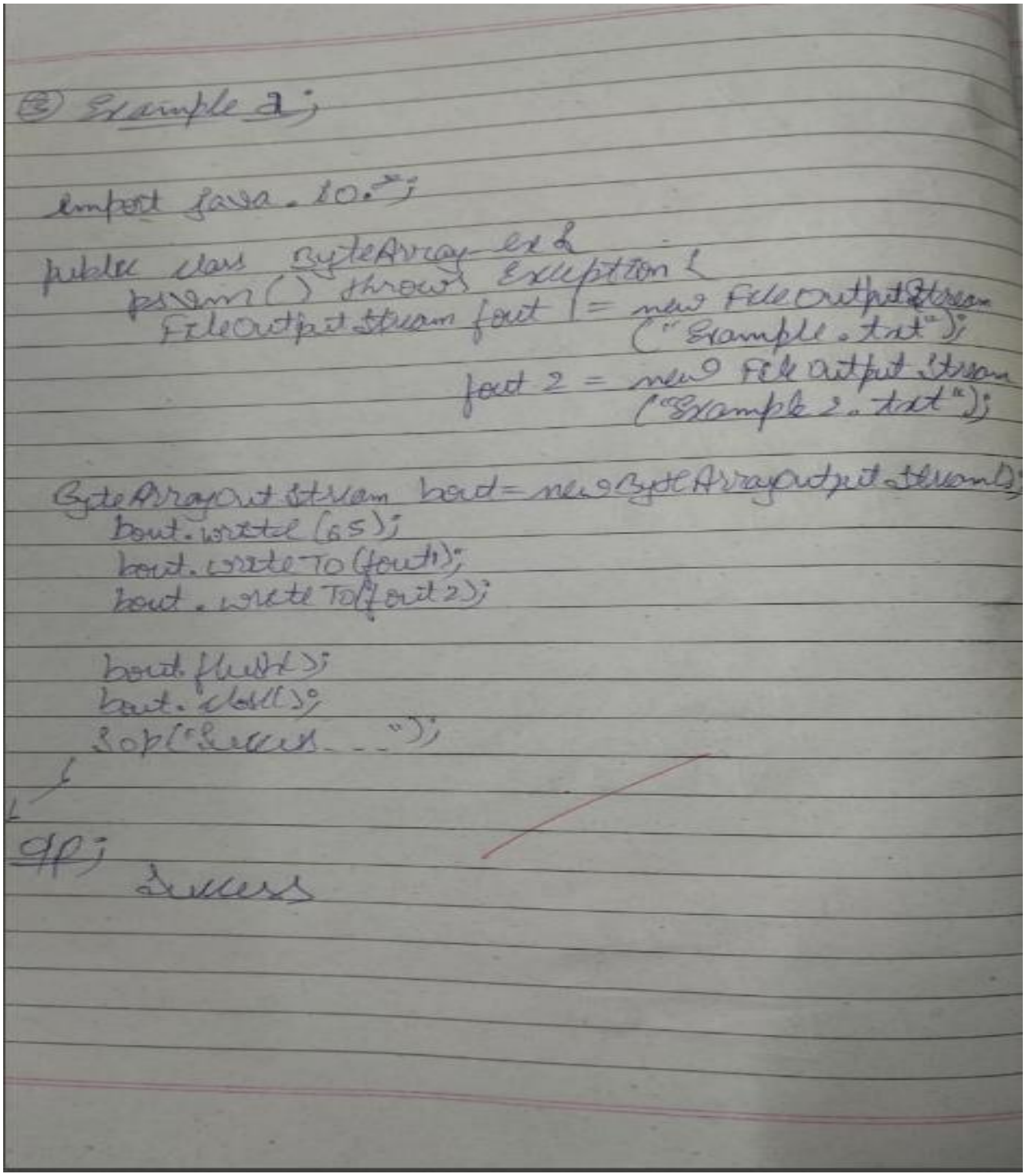


① Create a button & add a action listnerfor mouse click.

```
public class ButtonExample{
     PSVm(){
  Frame frame = new Frame("Button Example");

  Button button = new Button("Click me");
  button.set Bounds (100, 100, 80, 30);

 button.addActionListener(new ActionListener(){
     public void actionPerformed(ActionEvent e){
         Sop("Button Clicked!");
   }
 });

frame.add(button);
frame. setLayout(null);
frame.set Size(300,200);
frame.set Visible(true);
```

O/P; Welcome

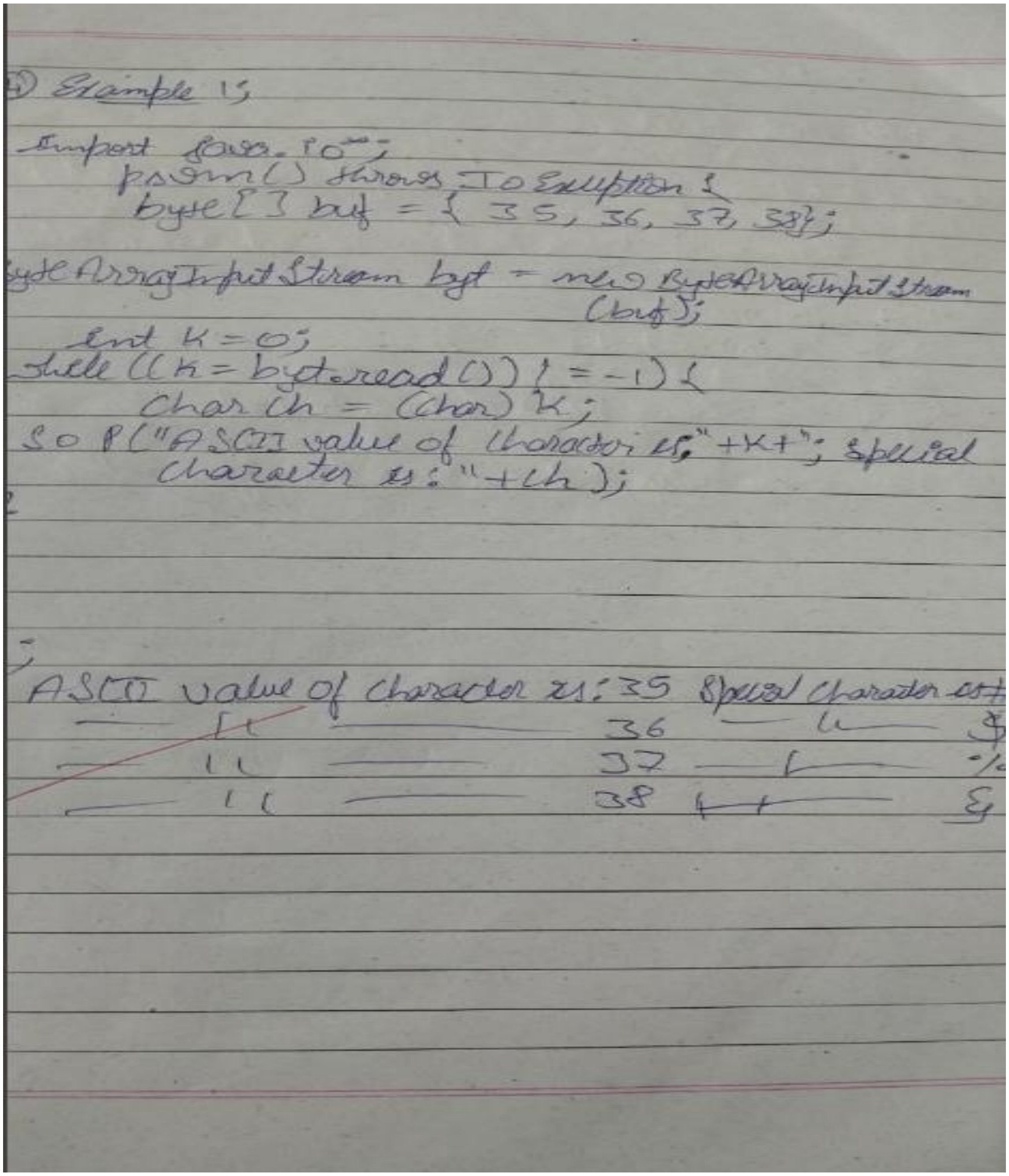
Click me



③ Example 2;

```
import java.io.*;

public class ByteArray ex2
    psvm() throws Exception {
      FileOutputStream fout1 = new FileOutputStream
                                   ("Example.txt");
                       fout 2 = new File Output Stream
                                   ("Example 2.txt");

 ByteArrayOutStream bout= new ByteArrayOutputStream();
   bout.write (65);
   bout.writeTo(fout1);
   bout.writeTo(fout2);

   bout.flush();
   bout.close();
   sop("Success ...");
  }
}
```

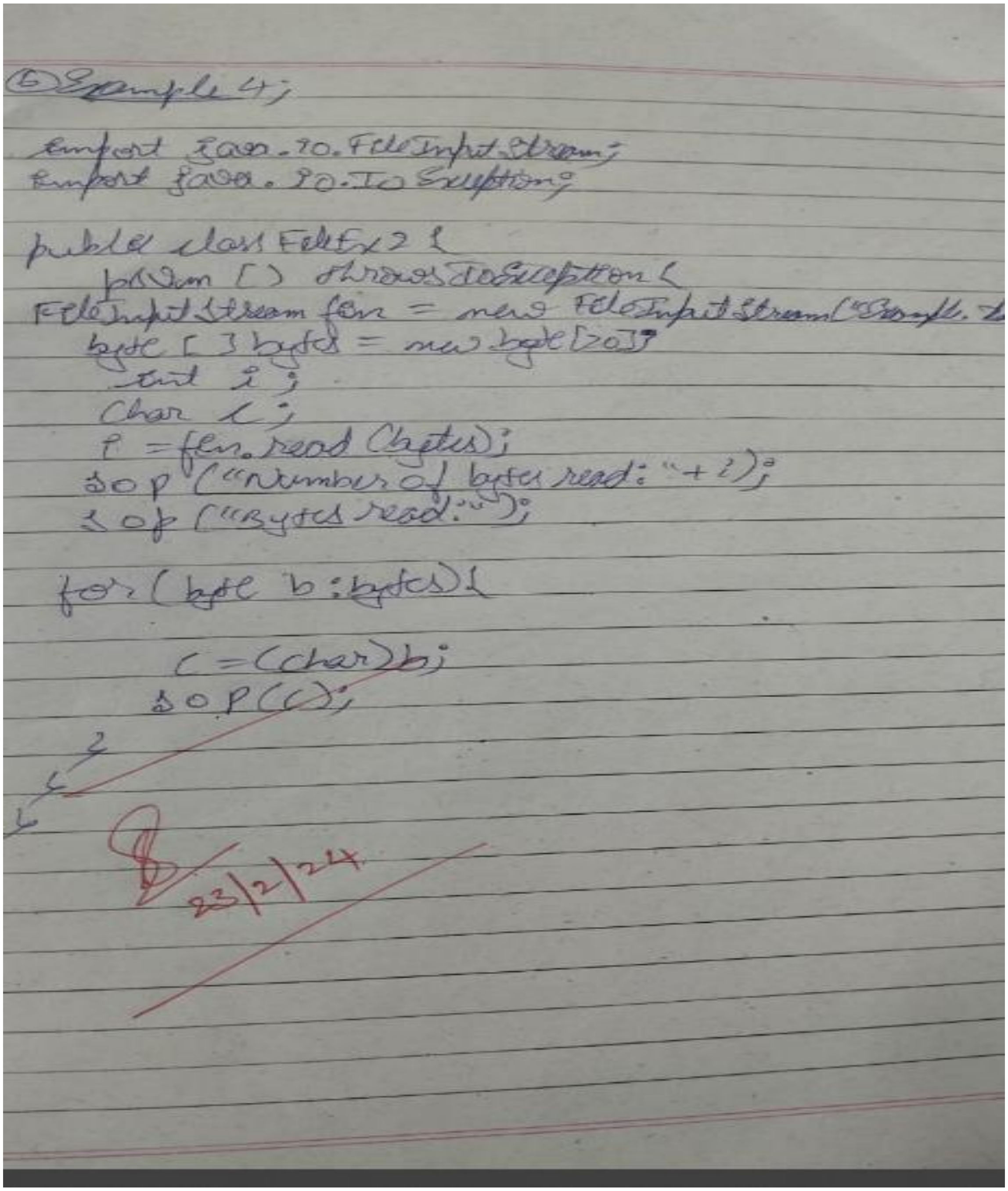
O/P;   Success



4) Example 1;

```
import java.io.*;
    psvm() throws IOException {
     byte[] buf = { 35, 36, 37, 38};

yte ArrayInputStream byt = new ByteArrayInputStream
                          (buf);
  int k = 0;
while ((k = byt.read()) != -1) {
     char ch = (char) k;
SOP("ASCII value of character is;" +k+ "; special
     character is:" +ch);
}
}
```

```
ASCII value of character is: 35 Special Character is #
  ——     ''     ————————       36 —— '' ——      $
  ——     ''     ————————       37 —— '' ——      %
  ——     ''     ————————       38 ——————        &
```



⑥ Example 4;

```
import java.io.FileInputStream;
import java.io.IOException;

public class FileEx2 {
    psvm [] throws IOException {
FileInputStream fin = new FileInputStream("Sample.t
   byte [ ] bytes = new byte[20];
    int i ;
    Char c ;
    i = fin.read (bytes);
   sop ("Number of bytes read: "+i);
   sop ("Bytes read: ");

  for (byte b : bytes) {

        c = (char) b;
      sop(c);
   }
 }
}
```

23/2/24