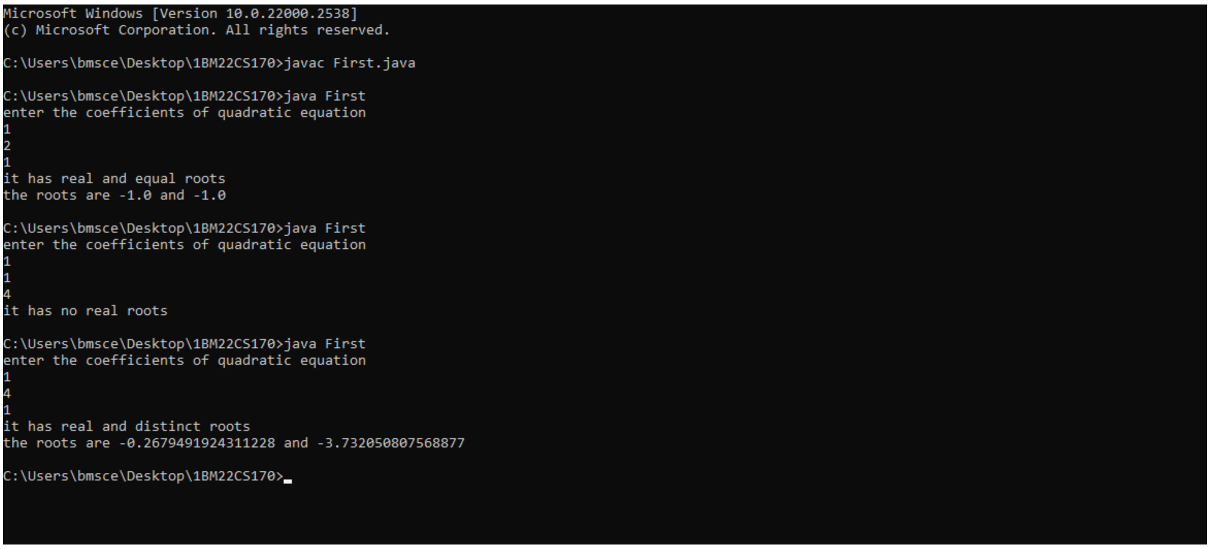
1.Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c= 0. Read in a, b, c and use the quadratic formula. If the discriminant b2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class First{
public static void main(String args[]){
int a,b,c;
double r1,r2,d;
System.out.println("enter the coefficients of quadratic equation");
Scanner s1= new Scanner(System.in);
a=s1.nextInt();
b=s1.nextInt();
c=s1.nextInt();
if(a==0){
System.out.println("coefficients are invalid");
}
else{
d=b*b-(4*a*c);
if(d>0){
System.out.println("it has real and distinct roots");
r1=((-b+Math.sqrt(d)))/(2*a);
r2=((-b-Math.sqrt(d)))/(2*a);
System.out.println("the roots are "+r1+" and "+r2);
}
else if(d==0){
System.out.println("it has real and equal roots");
r1=(-b)/(2*a);
System.out.println("the roots are "+r1+" and "+r1);
}
else{
System.out.println("it has no real roots");
}

}

}

}
```

Output:



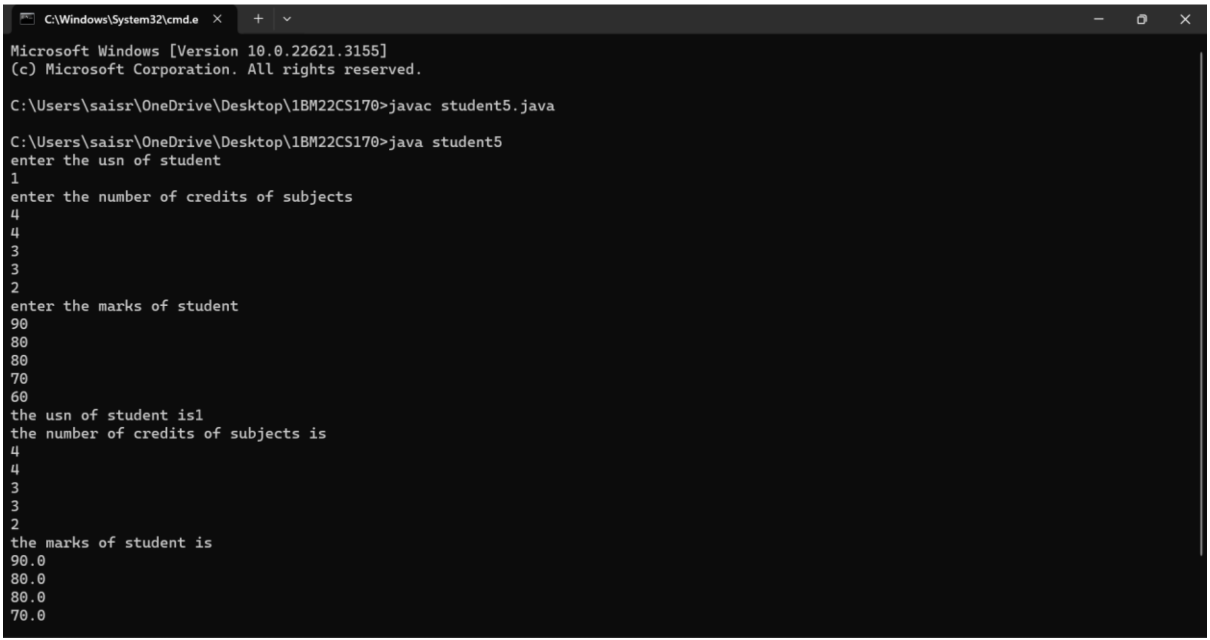
```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bmsce\Desktop\1BM22CS170>javac First.java

C:\Users\bmsce\Desktop\1BM22CS170>java First
enter the coefficients of quadratic equation
1
2
1
it has real and equal roots
the roots are -1.0 and -1.0

C:\Users\bmsce\Desktop\1BM22CS170>java First
enter the coefficients of quadratic equation
1
1
4
it has no real roots

C:\Users\bmsce\Desktop\1BM22CS170>java First
enter the coefficients of quadratic equation
1
4
1
it has real and distinct roots
the roots are -0.2679491924311228 and -3.732050807568877

C:\Users\bmsce\Desktop\1BM22CS170>
```

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

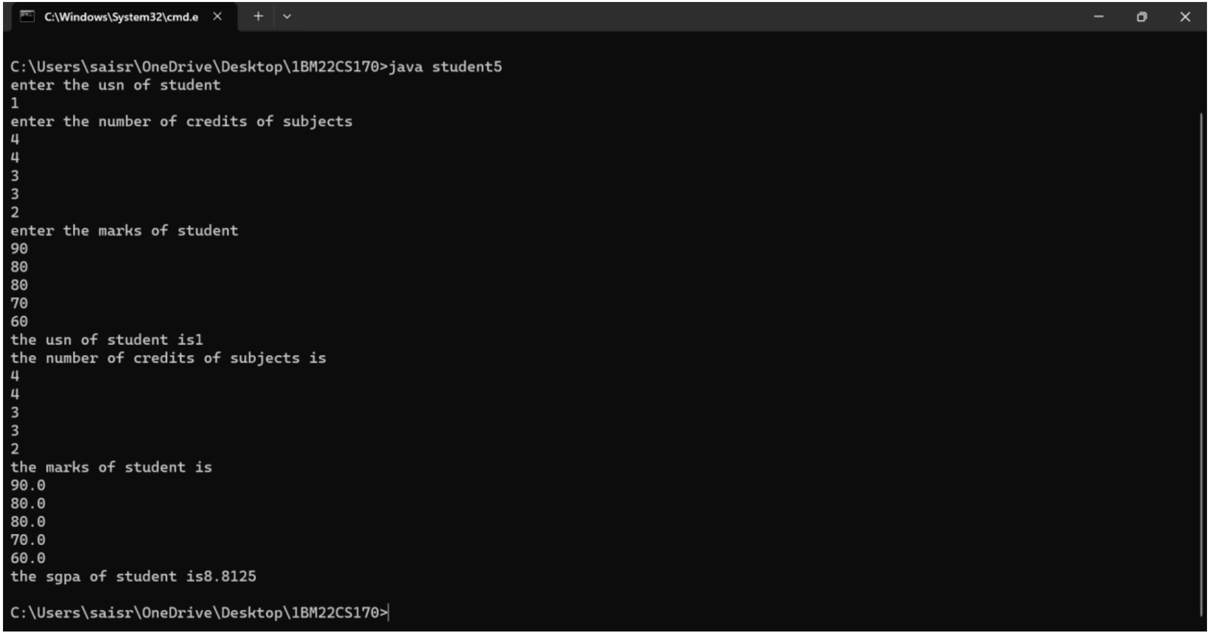
```
import java.util.*;
class student6{
int usn;
int credits[]=new int[5];
float marks[]=new float[5];
float sgpa;
void studdetails(int usn,int credits[],float marks[]){
this.usn=usn;
this.credits=credits;
this.marks=marks;
}
void printdetails(){
System.out.println("the usn of student is"+usn);
System.out.println("the number of credits of subjects is");
for(int i=0;i<5;i++){
System.out.println(credits[i]);
}
System.out.println("the marks of student is");
for(int i=0;i<5;i++){
System.out.println(marks[i]);
}
}
float studsgpa(){
int sum=0;
int creditssum=0;
for(int i=0;i<5;i++){
if(marks[i]>=90){
```

```
sum=sum+(credits[i]*10);
}
else if(marks[i]>=80){
sum=sum+(credits[i]*9);
}
else if(marks[i]>=70){
sum=sum+(credits[i]*8);
}
else if(marks[i]>=60){
sum=sum+(credits[i]*7);
}
else if(marks[i]>=50){
sum=sum+(credits[i]*6);
}
else if(marks[i]>=35){
sum=sum+(credits[i]*5);
}
else{
sum=sum+(credits[i]*0);
}
}
for(int i=0;i<5;i++){
creditssum=creditssum+credits[i];
}
sgpa=sum/(float)creditssum;
return sgpa;
}
}
class student5{
public static void main(String xx[]){
int usn;
int credits[]=new int[5];

float marks[]=new float[5];

student6 s=new student6();

Scanner s1=new Scanner(System.in);

System.out.println("enter the usn of student");

usn=s1.nextInt();

System.out.println("enter the number of credits of subjects");

for(int i=0;i<5;i++){

credits[i]=s1.nextInt();

}

System.out.println("enter the marks of student");

for(int i=0;i<5;i++){

marks[i]=s1.nextFloat();

}

s.studdetails(usn,credits,marks);

s.printdetails();

float sgpa=s.studsgpa();

System.out.println("the sgpa of student is"+sgpa);

}

}
```

Output:



```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>javac student5.java

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>java student5
enter the usn of student
1
enter the number of credits of subjects
4
4
3
3
2
enter the marks of student
90
80
80
70
60
the usn of student is1
the number of credits of subjects is
4
4
3
3
2
the marks of student is
90.0
80.0
80.0
70.0
```

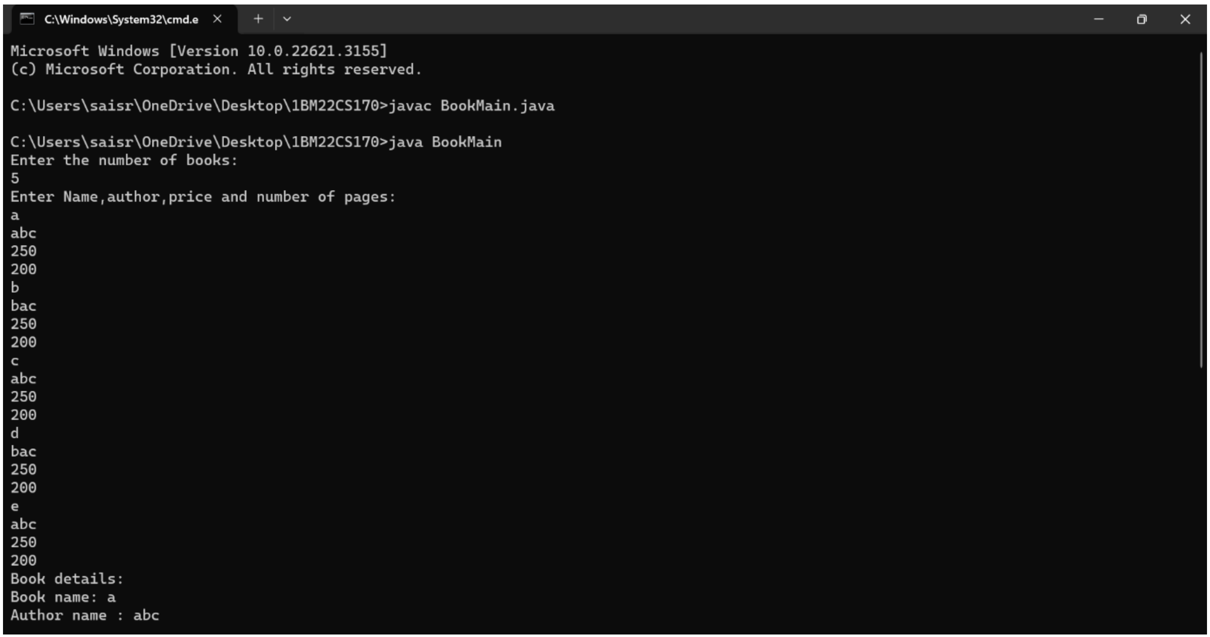


```
C:\Users\saisr\OneDrive\Desktop\1BM22CS170>java student5
enter the usn of student
1
enter the number of credits of subjects
4
4
3
3
2
enter the marks of student
90
80
80
70
60
the usn of student is1
the number of credits of subjects is
4
4
3
3
2
the marks of student is
90.0
80.0
80.0
70.0
60.0
the sgpa of student is8.8125

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>
```

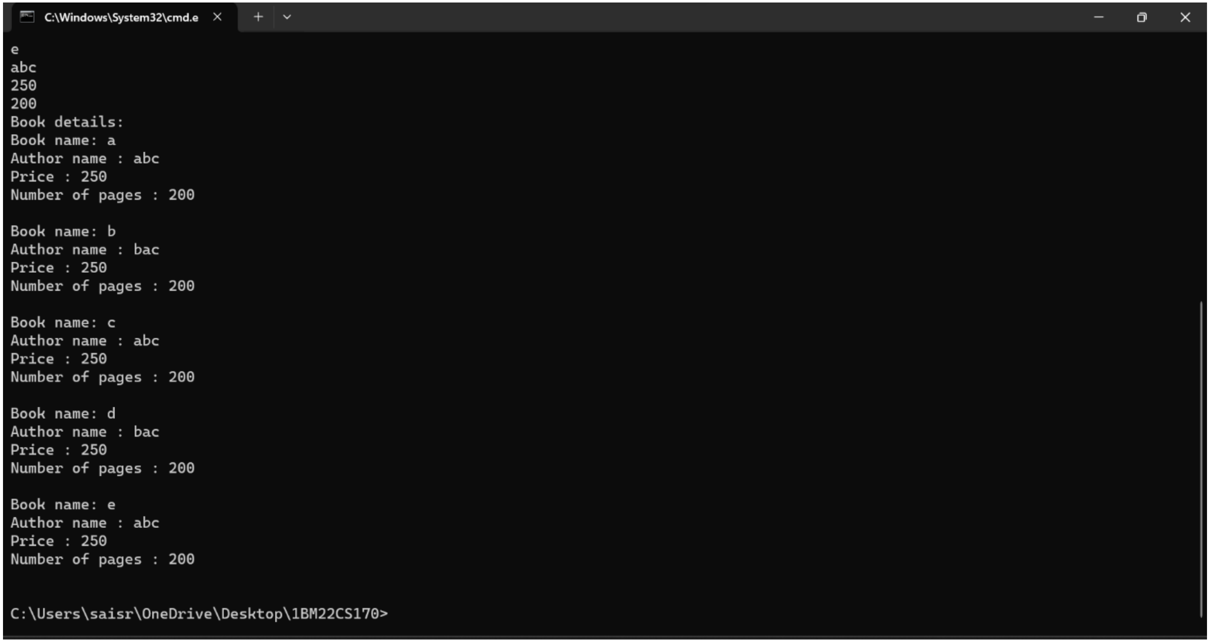
3. Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a to String() method that could display the complete details of the book. Develop a Java program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

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

Output:



```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>javac BookMain.java

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>java BookMain
Enter the number of books:
5
Enter Name,author,price and number of pages:
a
abc
250
200
b
bac
250
200
c
abc
250
200
d
bac
250
200
e
abc
250
200
Book details:
Book name: a
Author name : abc
```

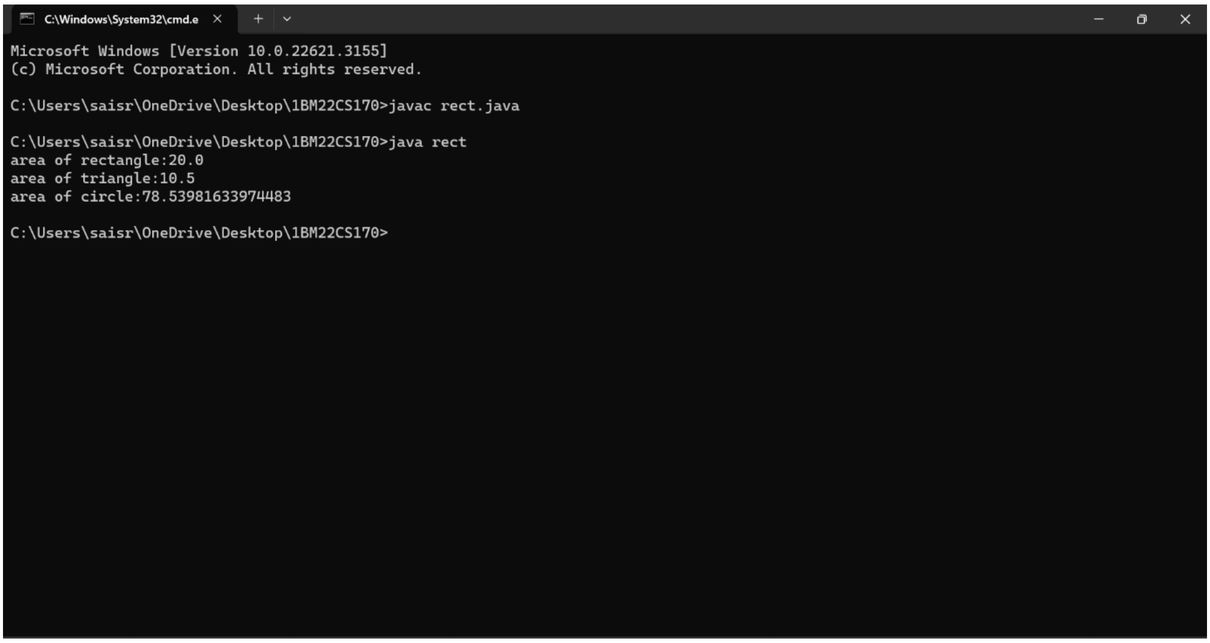


```
e
abc
250
200
Book details:
Book name: a
Author name : abc
Price : 250
Number of pages : 200

Book name: b
Author name : bac
Price : 250
Number of pages : 200

Book name: c
Author name : abc
Price : 250
Number of pages : 200

Book name: d
Author name : bac
Price : 250
Number of pages : 200

Book name: e
Author name : abc
Price : 250
Number of pages : 200


C:\Users\saisr\OneDrive\Desktop\1BM22CS170>
```

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.

```
abstract class shape{
protected int dimension1;
protected int dimension2;
public shape(int dimension1,int dimension2){
this.dimension1=dimension1;
this.dimension2=dimension2;
}
public abstract void printarea();
}
class rectangle extends shape{
public rectangle(int length,int width){
super(length,width);
}
public void printarea(){
double area=dimension1*dimension2;
System.out.println("area of rectangle:"+area);
}
}
class triangle extends shape{
public triangle(int base,int height){
super(base,height);
}
public void printarea(){
double area=0.5*dimension1*dimension2;
System.out.println("area of triangle:"+area);
}
}
class circle extends shape{
public circle(int radius){
super(radius,0);
}
public void printarea(){
double area=Math.PI*dimension1*dimension1;
System.out.println("area of circle:"+area);
}
}
public class rect{
public static void main(String xx[]){
rectangle r=new rectangle(2,10);
triangle t=new triangle(3,7);
circle c=new circle(5);
r.printarea();
t.printarea();
c.printarea();
}
}
```

Output:



```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>javac rect.java

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>java rect
area of rectangle:20.0
area of triangle:10.5
area of circle:78.53981633974483

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>
```

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

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

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

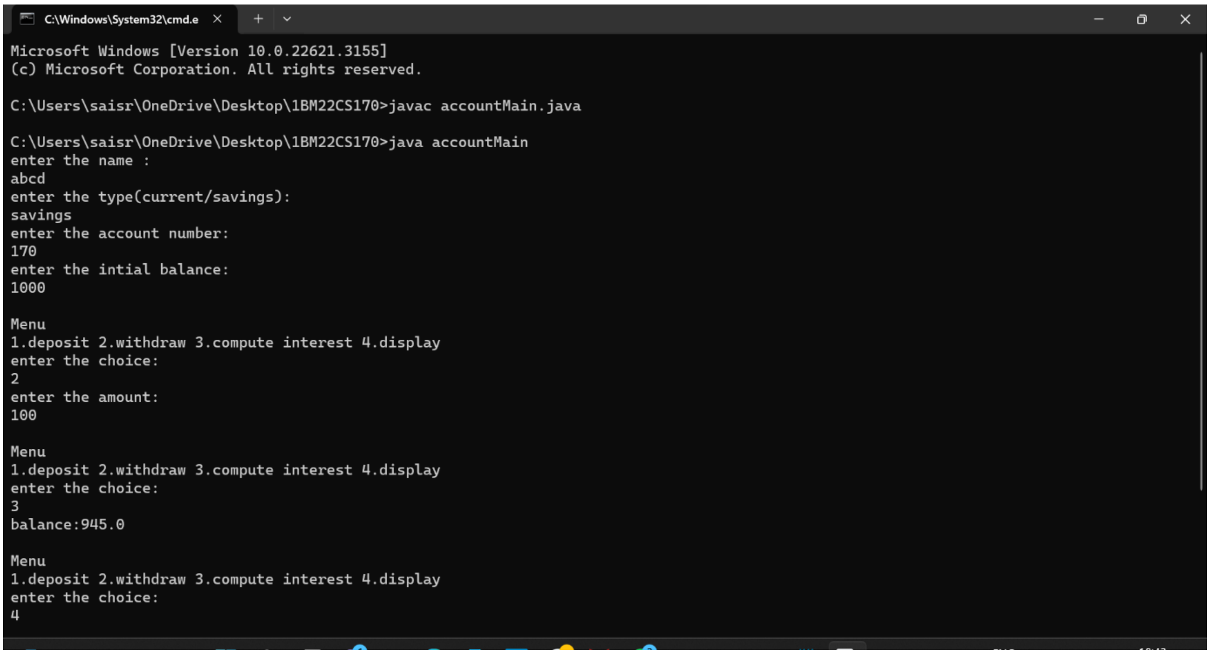
```
import java.util.Scanner;
class account
{
        String name;
        int accno;
        String type;
        double balance;

        account(String name,int accno,String type,double balance)
        {
                this.name=name;
                this.accno=accno;
                this.type=type;
                this.balance=balance;
        }
        void deposit(double amount)
        {
                balance+=amount;
```

```
		}
		void withdraw(double amount)
		{
			if((balance-amount)>=0)
			{
				balance-=amount;
			}
			else
			{
				System.out.println("insufficient balance,cant withdraw");
			}
		}

		void display()
		{

		System.out.println("name:"+name+"accno:"+accno+"type:"+type+"balance:"+balance);
		}
	}
	class savAcct extends account
	{


		private static double rate=5;
		savAcct(String name,int accno,double balance)
		{
			super(name,accno,"savings",balance);


		}

		void interest()
		{
        balance+=balance*(rate)/100;
        System.out.println("balance:"+balance);
    }


}
class curAcct extends account
{

    private double minBal=500;
    private double serviceCharges=50;

    curAcct(String name,int accno,double balance)
    {
        super(name,accno,"current",balance);

    }


    void checkmin()
    {

        if(balance<minBal)
        {
            System.out.println("balance is less than min balance,service charges
            imposed: "+serviceCharges);
            balance-=serviceCharges;
            System.out.println("balance is:"+balance);
        }

    }
```

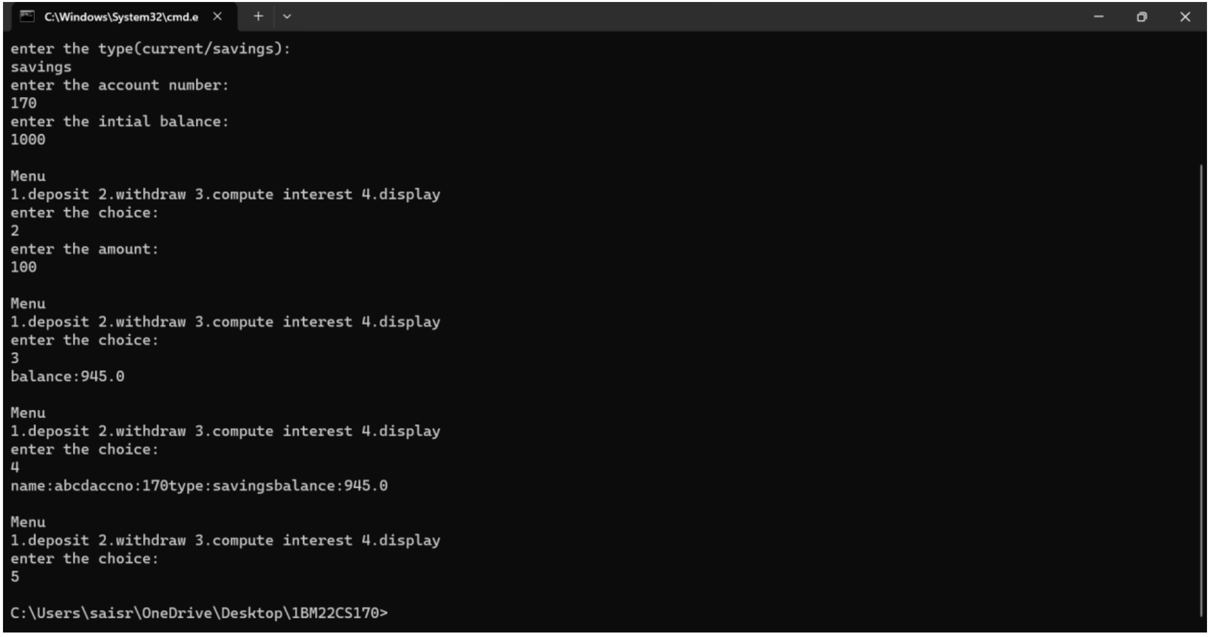
```
}
class accountMain
{
        public static void main(String a[])
        {
                Scanner s=new Scanner(System.in);
                System.out.println("enter the name :");
                String name=s.next();
                System.out.println("enter the type(current/savings):");
                String type=s.next();
                System.out.println("enter the account number:");
                int accno=s.nextInt();
                System.out.println("enter the intial balance:");
                double balance=s.nextDouble();
                int ch;
                double amount1,amount2;
                account acc=new account(name,accno,type,balance);
                savAcct sa=new savAcct(name,accno,balance);
                curAcct ca=new curAcct(name,accno,balance);
                while(true)
                {
                        if(acc.type.equals("savings"))
                        {
                                System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute
                                interest 4.display");
                                System.out.println("enter the choice:");
                                ch=s.nextInt();
                                switch(ch)
                                {
                                        case 1:System.out.println("enter the amount:");
                amount1=s.nextInt();
                sa.deposit(amount1);
                break;
            case 2:System.out.println("enter the amount:");
                amount2=s.nextInt();
                sa.withdraw(amount2);
                break;
            case 3:sa.interest();
                break;
            case 4:sa.display();
                break;
            case 5:System.exit(0);
            default:System.out.println("invalid input");
                break;
        }
}
else
{
        System.out.println("\nMenu\n1.deposit 2.withdraw  3.display");
        System.out.println("enter the choice:");
        ch=s.nextInt();
        switch(ch)
        {
            case 1:System.out.println("enter the amount:");
                amount1=s.nextInt();
                ca.deposit(amount1);
                break;
            case 2:System.out.println("enter the amount:");
                amount2=s.nextInt();
                ca.withdraw(amount2);
                ca.checkmin();
```

```
                                                            break;


                                                    case 3:ca.display();
                                                            break;
                                                    case 4:System.exit(0);
                                                    default:System.out.println("invalid input");
                                                            break;
                                            }
                                    }
                            }
                    }
            }
```

Output:



```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>javac accountMain.java

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>java accountMain
enter the name :
abcd
enter the type(current/savings):
savings
enter the account number:
170
enter the intial balance:
1000

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
2
enter the amount:
100

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
3
balance:945.0

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
4
```



```
enter the type(current/savings):
savings
enter the account number:
170
enter the intial balance:
1000

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
2
enter the amount:
100

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
3
balance:945.0

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
4
name:abcdaccno:170type:savingsbalance:945.0

Menu
1.deposit 2.withdraw 3.compute interest 4.display
enter the choice:
5

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>
```

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

```
package cie;
import java.util.Scanner;
public class student{
String usn;
String name;
int semester;
public void studdetails(){
Scanner s1=new Scanner(System.in);
System.out.println("enter the usn of student");
usn=s1.next();
System.out.println("enter the name of student");
name=s1.next();
System.out.println("enter the semester of student");
semester=s1.nextInt();
}
public void printdetails(){
System.out.println("the usn of student is"+usn);
System.out.println("the name of student is"+name);
System.out.println("the semester of student is"+semester);
}
}

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

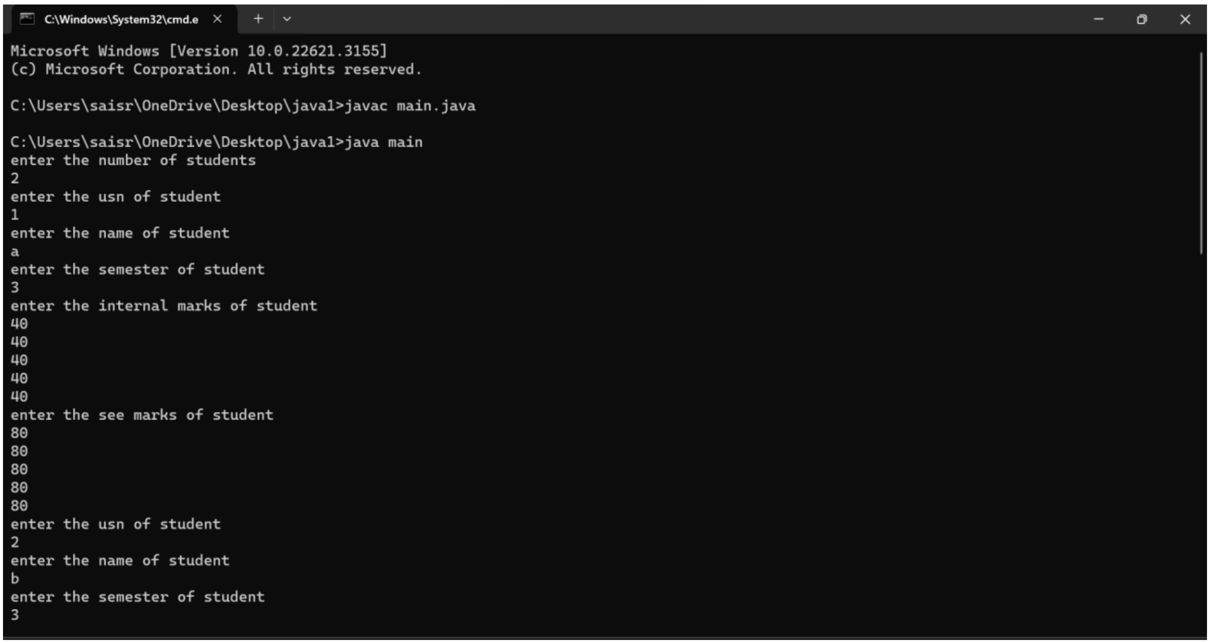
```
public class internals extends student{
public float studmarks[]=new float[5];
public void studmarks(){
Scanner s1=new Scanner(System.in);
System.out.println("enter the internal marks of student");
for(int i=0;i<5;i++){
studmarks[i]=s1.nextFloat();
}
}
public void printmarks(){
System.out.println("the marks of student is");
for(int i=0;i<5;i++){
System.out.println(studmarks[i]);
}
}
}
```

```
package see;
import java.util.Scanner;
import cie.student;
public class external extends student{
public float seemarks[]=new float[5];
public void seemarks(){
Scanner s2=new Scanner(System.in);
System.out.println("enter the see marks of student");
for(int i=0;i<5;i++){
seemarks[i]=s2.nextFloat();
}
}
public void printseemarks(){
System.out.println("the see marks of student is");
```

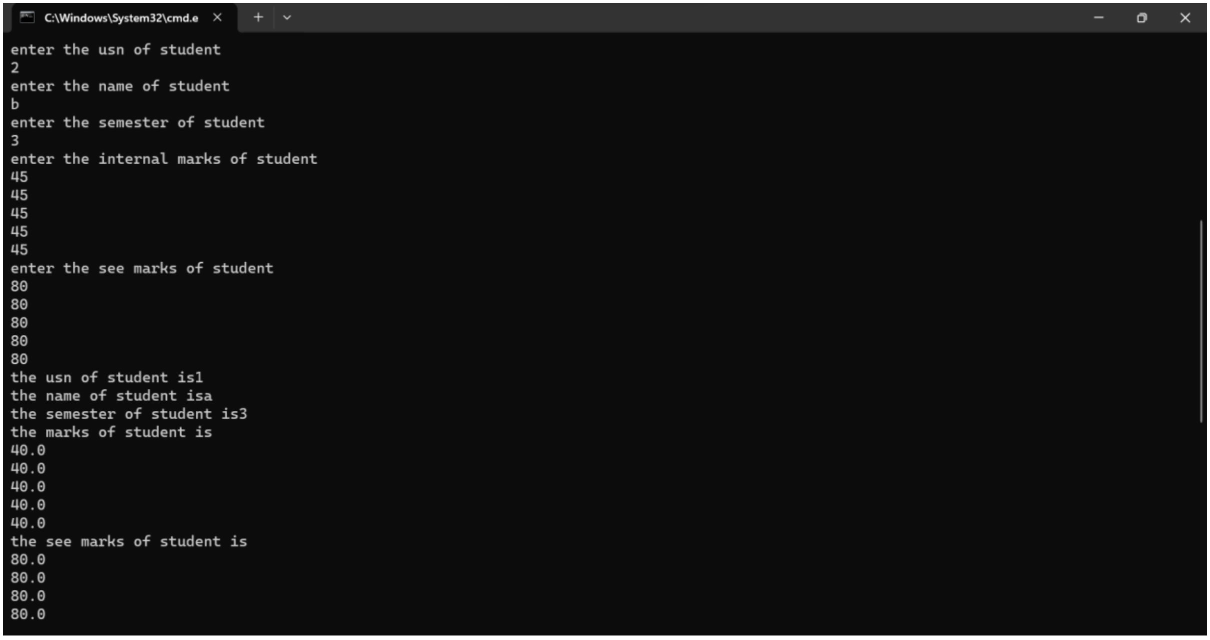
```
for(int i=0;i<5;i++){
System.out.println(seemarks[i]);
}
}
}


import java.util.Scanner;
import cie.student;
import cie.internals;
import see.external;
class main{
public static void main(String xx[]){
int n;
Scanner s3=new Scanner(System.in);
System.out.println("enter the number of students");
n=s3.nextInt();
internals internalmarks[]=new internals[n];
external externalmarks[]=new external[n];
for(int i=0;i<n;i++){
internalmarks[i]=new internals();
externalmarks[i]=new external();
internalmarks[i].studdetails();
internalmarks[i].studmarks();
externalmarks[i].seemarks();
}
float finalmarks[][]=new float[n][5];
for(int i=0;i<n;i++){
for(int j=0;j<5;j++){
finalmarks[i][j]=internalmarks[i].studmarks[j]+((externalmarks[i].seemarks[j])/2);
}
}
```

```
for(int i=0;i<n;i++){
internalmarks[i].printdetails();
internalmarks[i].printmarks();
externalmarks[i].printseemarks();
}
for(int i=0;i<n;i++){
System.out.println("the final marks of "+i+" student is");
for(int j=0;j<5;j++){
System.out.println(finalmarks[i][j]);
}
}
}
}
```

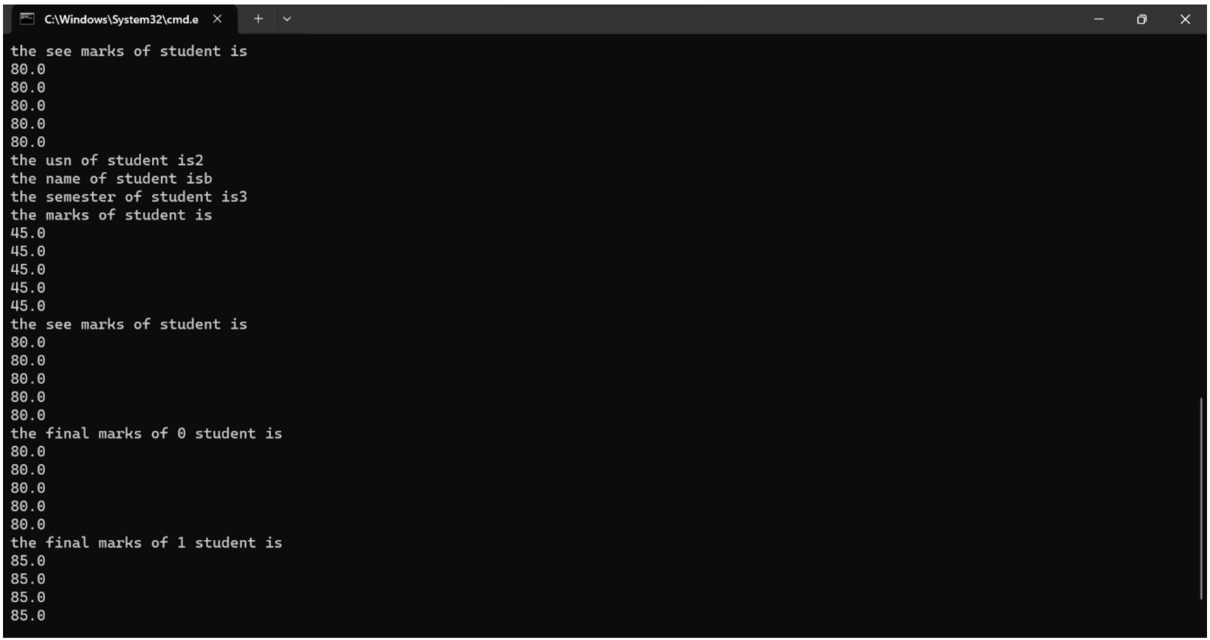
Output:



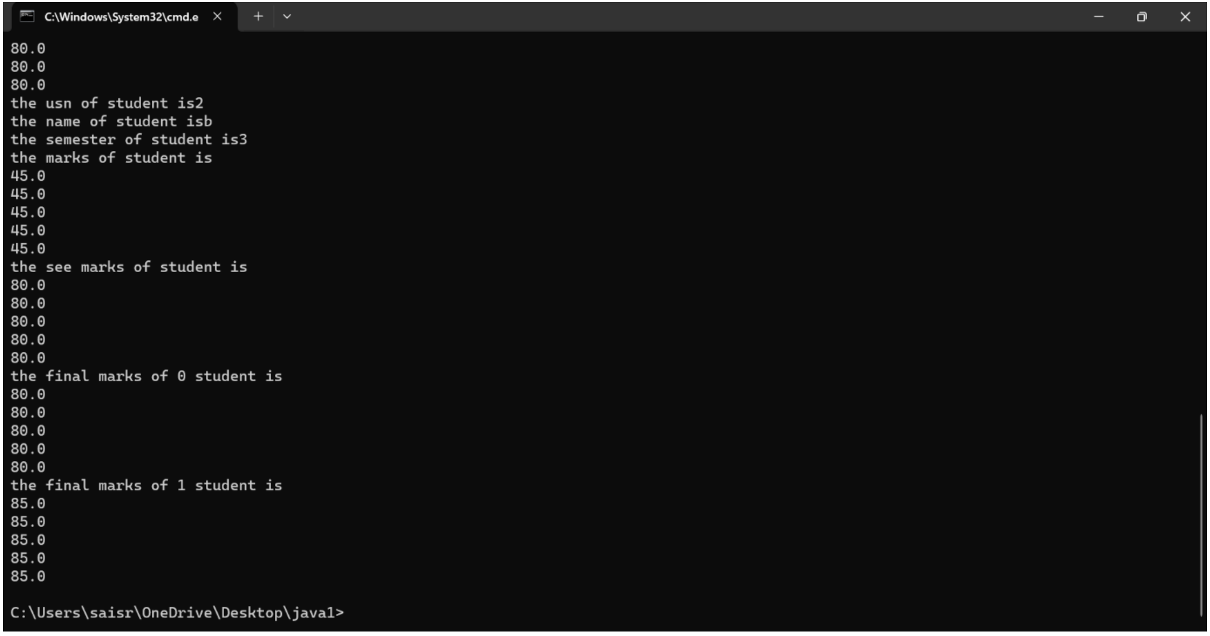
```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saisr\OneDrive\Desktop\java1>javac main.java

C:\Users\saisr\OneDrive\Desktop\java1>java main
enter the number of students
2
enter the usn of student
1
enter the name of student
a
enter the semester of student
3
enter the internal marks of student
40
40
40
40
40
enter the see marks of student
80
80
80
80
80
enter the usn of student
2
enter the name of student
b
enter the semester of student
3
```



```
enter the usn of student
2
enter the name of student
b
enter the semester of student
3
enter the internal marks of student
45
45
45
45
45
enter the see marks of student
80
80
80
80
80
the usn of student is1
the name of student isa
the semester of student is3
the marks of student is
40.0
40.0
40.0
40.0
40.0
the see marks of student is
80.0
80.0
80.0
80.0
```



```
the see marks of student is
80.0
80.0
80.0
80.0
80.0
the usn of student is2
the name of student isb
the semester of student is3
the marks of student is
45.0
45.0
45.0
45.0
45.0
the see marks of student is
80.0
80.0
80.0
80.0
80.0
the final marks of 0 student is
80.0
80.0
80.0
80.0
80.0
the final marks of 1 student is
85.0
85.0
85.0
85.0
```

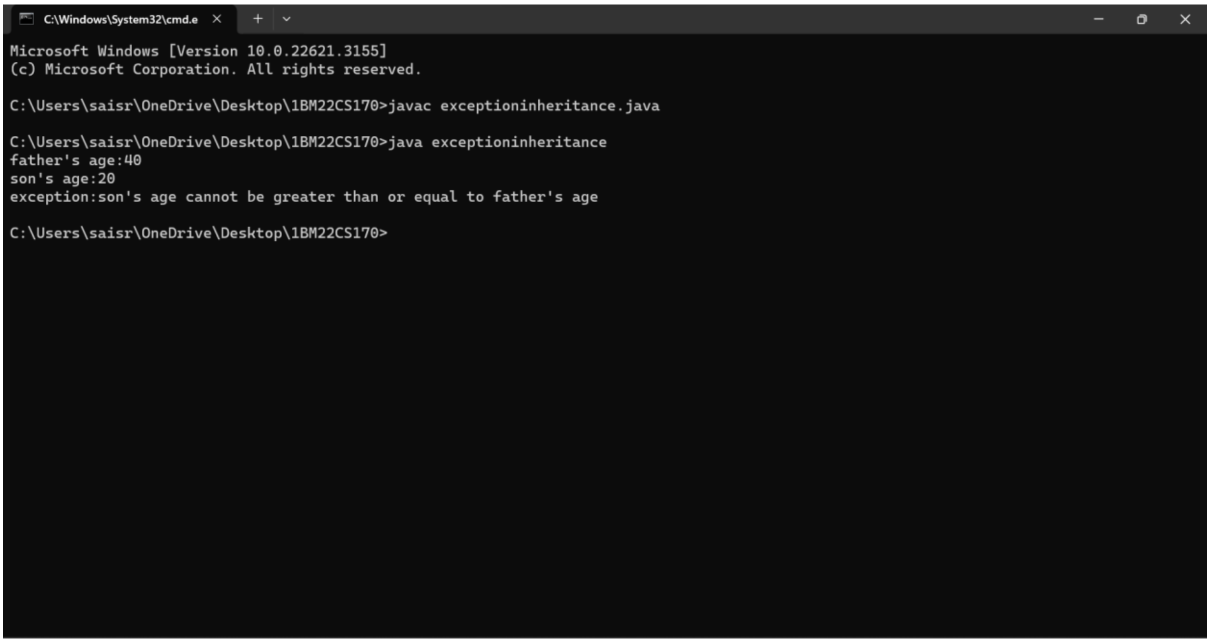


```
80.0
80.0
80.0
the usn of student is2
the name of student isb
the semester of student is3
the marks of student is
45.0
45.0
45.0
45.0
45.0
the see marks of student is
80.0
80.0
80.0
80.0
80.0
the final marks of 0 student is
80.0
80.0
80.0
80.0
80.0
the final marks of 1 student is
85.0
85.0
85.0
85.0
85.0

C:\Users\saisr\OneDrive\Desktop\java1>
```

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

```
class wrongage extends Exception{
public wrongage(String Message){
super(Message);
}
}
class father{
private int age;
public father(int age) throws wrongage{
if(age<0){
throw new wrongage("age cannot be negative");
}
this.age=age;
}
public int getage(){
return age;
}
}
class son extends father{
private int sonage;
public son(int fatherage,int sonage) throws wrongage{
super(fatherage);
if(sonage>=fatherage){
throw new wrongage("son's age cannot be greater than or equal to father's age");
}
this.sonage=sonage;
}
public int getsonage(){

return sonage;

}

}

public class exceptioninheritance{

public static void main(String xx[]){

try{

son son1=new son(40,20);

System.out.println("father's age:"+son1.getage());

System.out.println("son's age:"+son1.getsonage());

son invalidson=new son(30,35);

System.out.println("this line will not be reached");

}

catch(wrongage e){

System.out.println("exception:"+e.getMessage());

}

}

}
```

Output:

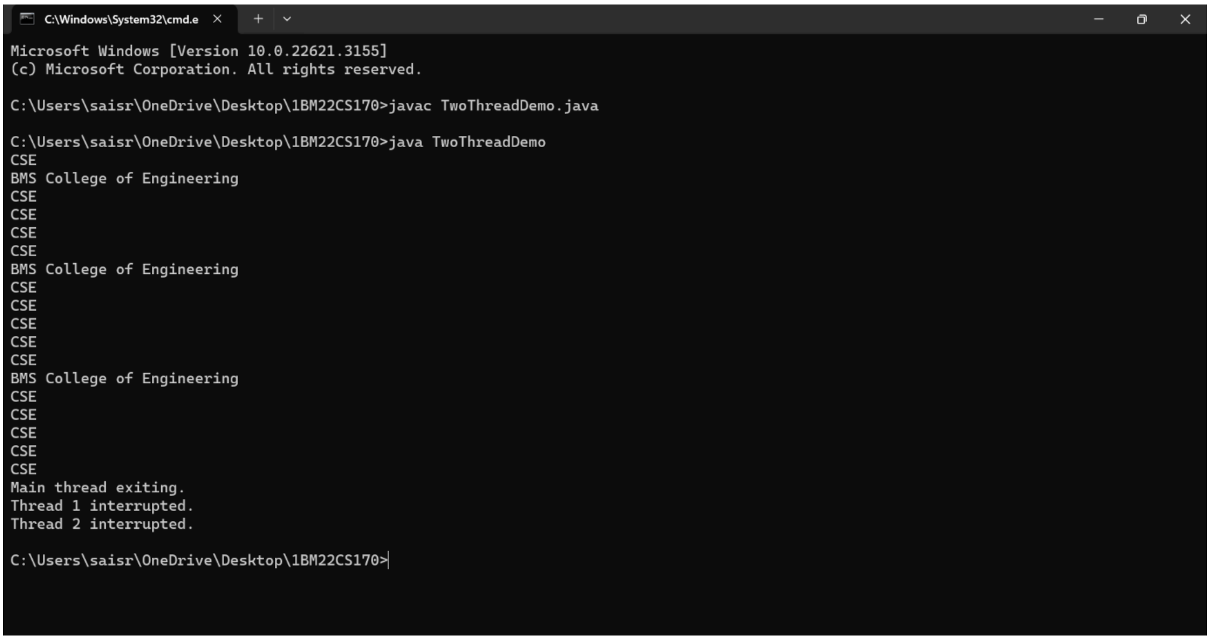


```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>javac exceptioninheritance.java

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>java exceptioninheritance
father's age:40
son's age:20
exception:son's age cannot be greater than or equal to father's age

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>
```

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 DisplayMessageThread extends Thread {
    private final String message;
    private final long interval; // in milliseconds

    DisplayMessageThread(String message, long interval) {
        this.message = message;
        this.interval = interval;
    }

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

public class TwoThreadDemo {
    public static void main(String[] args) {
        DisplayMessageThread thread1 = new DisplayMessageThread("BMS College of Engineering", 10000); // 10 seconds
        DisplayMessageThread thread2 = new DisplayMessageThread("CSE", 2000); // 2 seconds

        thread1.setName("Thread 1");
        thread2.setName("Thread 2");
        thread1.start();
        thread2.start();

        try {
            // Let the threads run for a while
            Thread.sleep(30000); // Let the program run for 30 seconds
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted.");
        }

        // Interrupt both threads to stop them
        thread1.interrupt();
        thread2.interrupt();

        System.out.println("Main thread exiting.");
    }
}
```

Output:



```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>javac TwoThreadDemo.java

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>java TwoThreadDemo
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
Main thread exiting.
Thread 1 interrupted.
Thread 2 interrupted.

C:\Users\saisr\OneDrive\Desktop\1BM22CS170>
```

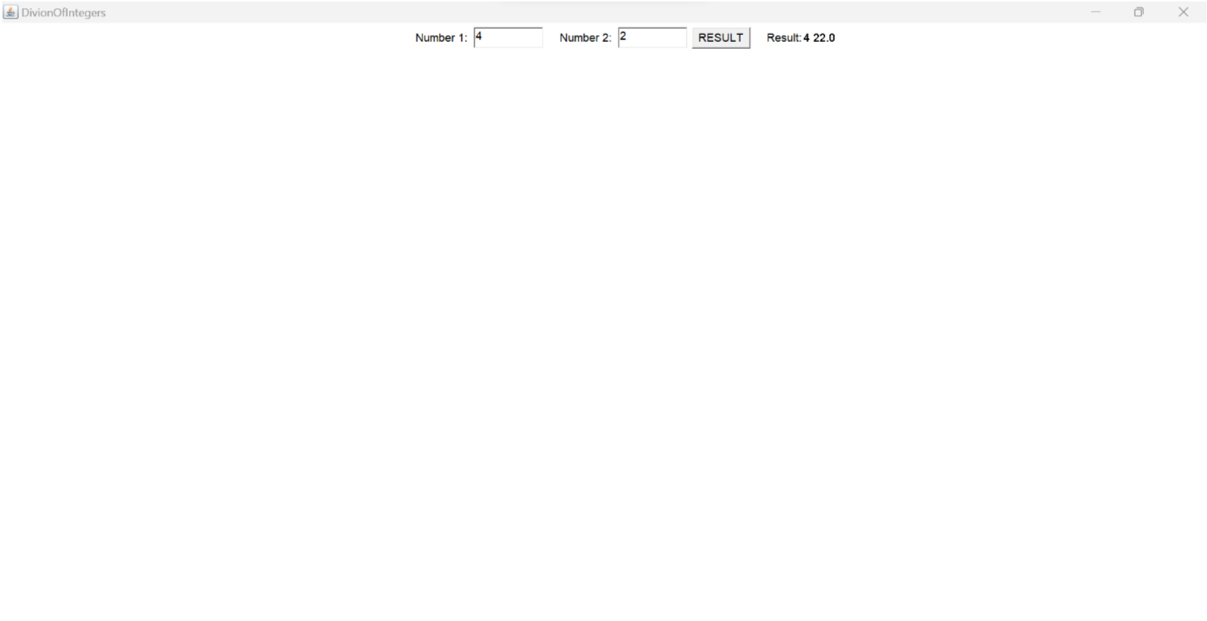
9.write a program thar creates a user interface to perform integer divisions. the user enters two numbers in the text fields, Num1 and Num2. the division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberformatException. If Num2 were zero, the program would throw an Arithmetic Exception display the exception in a message dialog box.

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

public class DivisionMain1 extends Frame implements ActionListener
{
	TextField num1,num2;
	Button dResult;
	Label outResult;
	String out="";
	double resultNum;
	int flag=0;

	public DivisionMain1()
	{
		setLayout(new FlowLayout());

		dResult = new Button("RESULT");
		Label number1 = new Label("Number 1:",Label.RIGHT);
		Label number2 = new Label("Number 2:",Label.RIGHT);
		num1=new TextField(5);
		num2=new TextField(5);
		outResult = new Label("Result:",Label.RIGHT);

		add(number1);
		add(num1);
		add(number2);
```

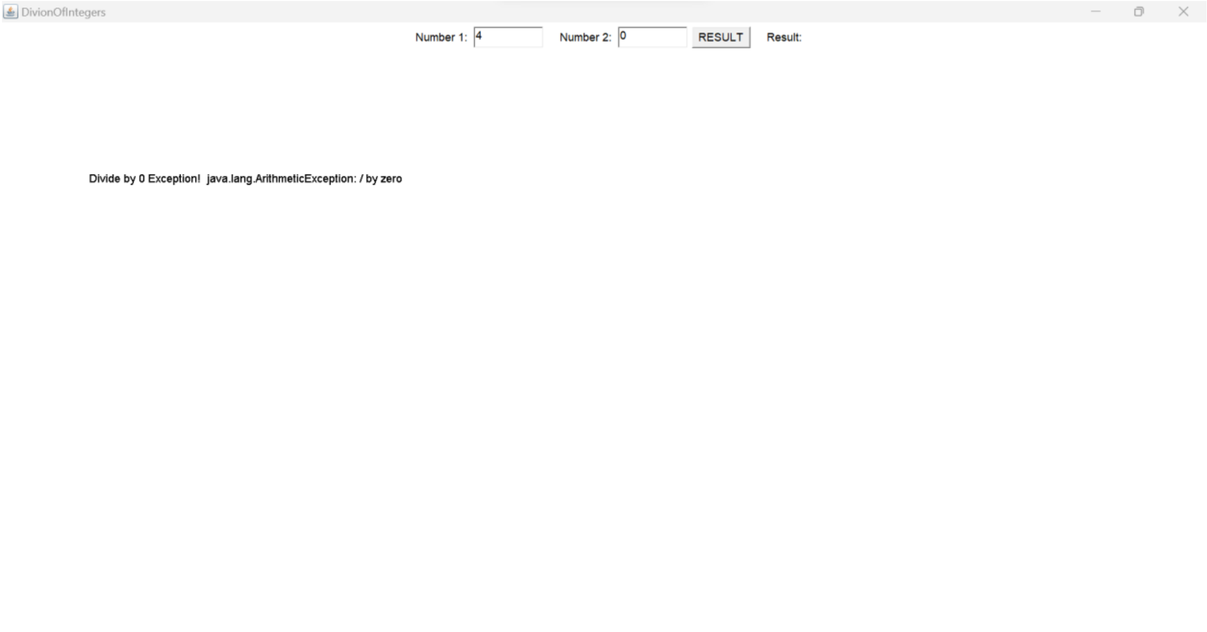
```
    add(num2);
    add(dResult);
    add(outResult);

    num1.addActionListener(this);
    num2.addActionListener(this);
    dResult.addActionListener(this);
    addWindowListener(new WindowAdapter()
    {
        public void windowClosing(WindowEvent we)
        {
            System.exit(0);
        }
    });
}

public void actionPerformed(ActionEvent ae)
{
    int n1,n2;
    try
    {
        if (ae.getSource() == dResult)
        {
            n1=Integer.parseInt(num1.getText());
            n2=Integer.parseInt(num2.getText());

            /*if(n2==0)
                throw new ArithmeticException();*/
            out=n1+" "+n2;
            resultNum=n1/n2;
            out+=String.valueOf(resultNum);
```

```
                        repaint();

                    }
                }
                catch(NumberFormatException e1)
                {
                    flag=1;
                    out="Number Format Exception!  "+e1;
                    repaint();
                }
                catch(ArithmeticException e2)
                {
                    flag=1;
                    out="Divide by 0 Exception!  "+e2;
                    repaint();
                }

            }

            public void paint(Graphics g)
            {
                if(flag==0)

            g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+outResult.getHeig
        ht()-8);
                else
                g.drawString(out,100,200);
                flag=0;
            }

            public static void main(String[] args)
            {
                    DivisionMain1 dm=new DivisionMain1();
                    dm.setSize(new Dimension(800,400));
                    dm.setTitle("DivionOfIntegers");
                    dm.setVisible(true);
          }

}
```

Output:

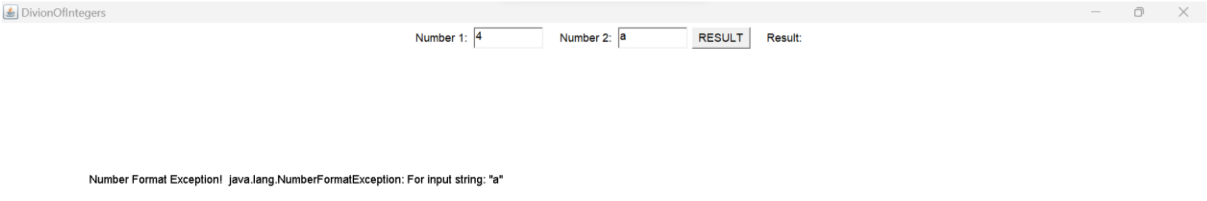


Number 1: 4 Number 2: 2 Result: 4 22.0



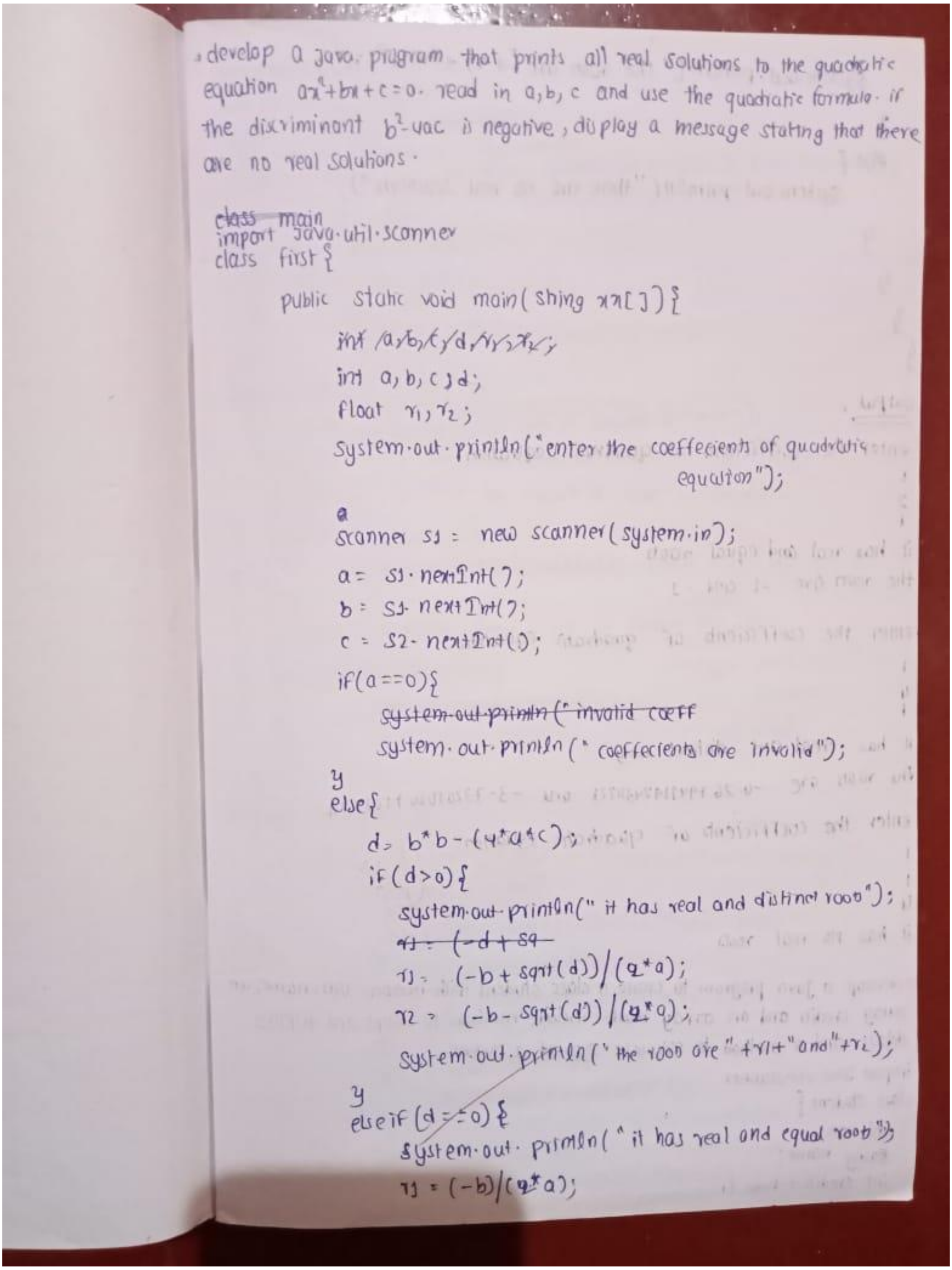
Number 1: 4 Number 2: 0 Result:

```
Divide by 0 Exception!  java.lang.ArithmeticException: / by zero
```



Number 1: 4 Number 2: a Result:

```
Number Format Exception!  java.lang.NumberFormatException: For input string: "a"
```



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 discriminant b^2-4ac is negative, display a message stating that there are no real solutions.

```
~~class main~~
import java.util.scanner
class first {

        public static void main(string xx[]){
            ~~int a,b,c,d,r1,r2;~~
            int a, b, c, d;
            float r1, r2;
            system.out.println("enter the coeffecients of quadratic
                                                  equation");
            a
            scanner s1 = new scanner(system.in);
            a= s1.nextInt();
            b= s1.nextInt();
            c = s2.nextInt();
            if(a==0){
                ~~system.out.println("invalid coeff~~
                system.out.println(" coeffecients are invalid");
            }
            else{
               d= b*b-(4*a*c);
               if(d>0){
                  system.out.println(" it has real and distinct roots");
                  ~~r1= (-d+sq~~
                  r1= (-b+sqrt(d))/(2*a);
                  r2= (-b-sqrt(d))/(2*a);
                  system.out.println(" the roots are "+r1+"and"+r2);
             }
             elseif (d==0){
                  system.out.println(" it has real and equal roots");
                  r1 = (-b)/(2*a);
```

```
    System.out.println(" the roots are "+r1 + "and "+r2);
  }
  else {
      System.out.println("there are no real solutions");
   }
  }
 }
}
```

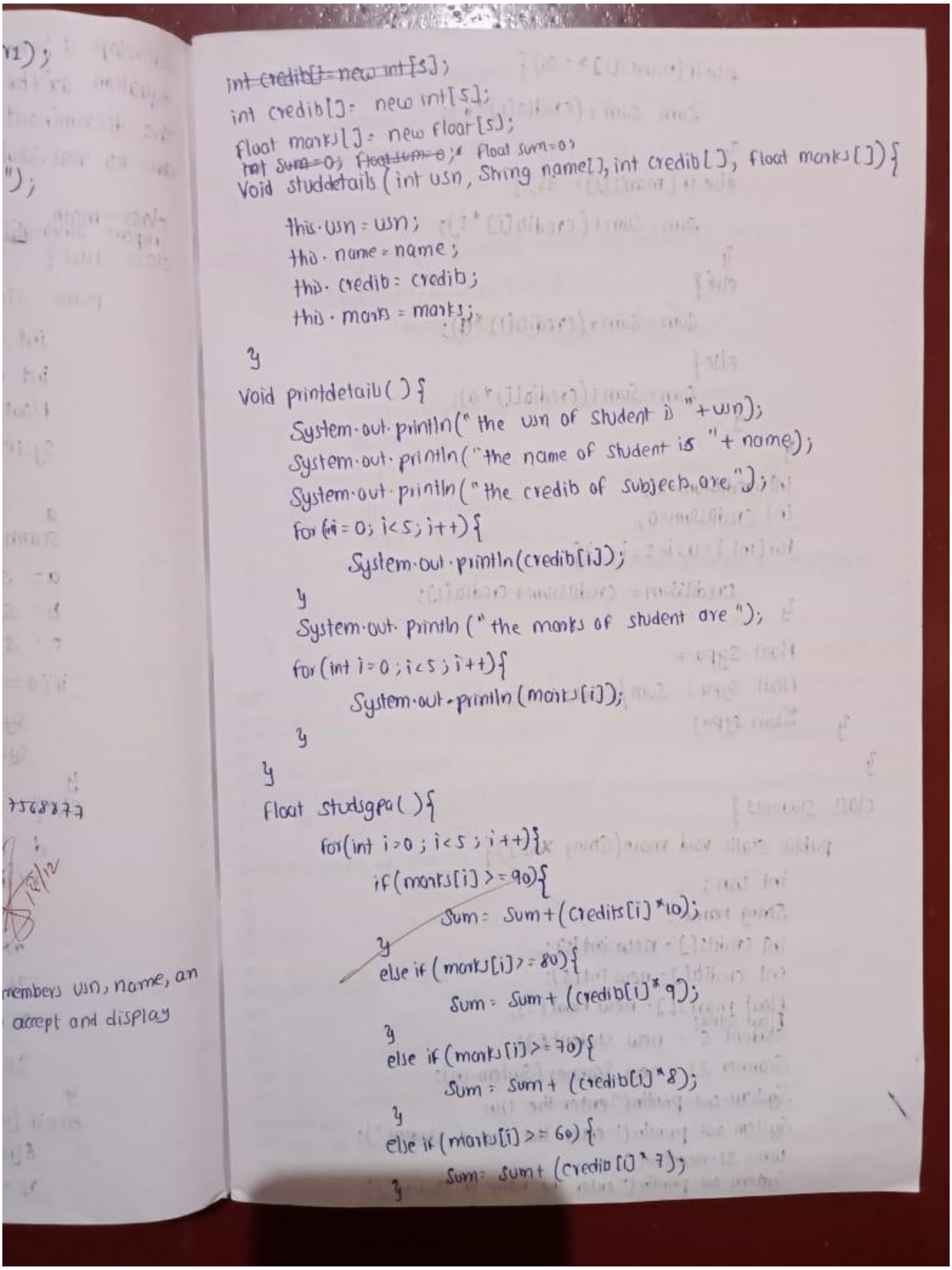
output

```
enter the coefficients of quadratic equation:
1
2
1
it has real and equal roots
the roots are  -1 and -1

enter the coefficients of quadratic equation:
1
4
1
it has real and distinct roots
the roots are  -0.2679491924311228 and -3.7320508075688777
enter the coefficients of quadratic equation
1
1
4
it has no real roots
```

* 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 student {
 ~~int~~ usn;
 String name;
 ~~int credits = new~~ i1
```

```
int ~~credits~~
int credit
float mar
~~int sum=~~
void stud
    this.(
    this.
    this.
    this.
}
void printde
   System
   System
   System
   for (i =
      S
   }
   System
   for (int i
      Sy
   }
}
float studsg
   for(int
      i
      }
      el
      }
      e
      }
      el
      }
```



r1);

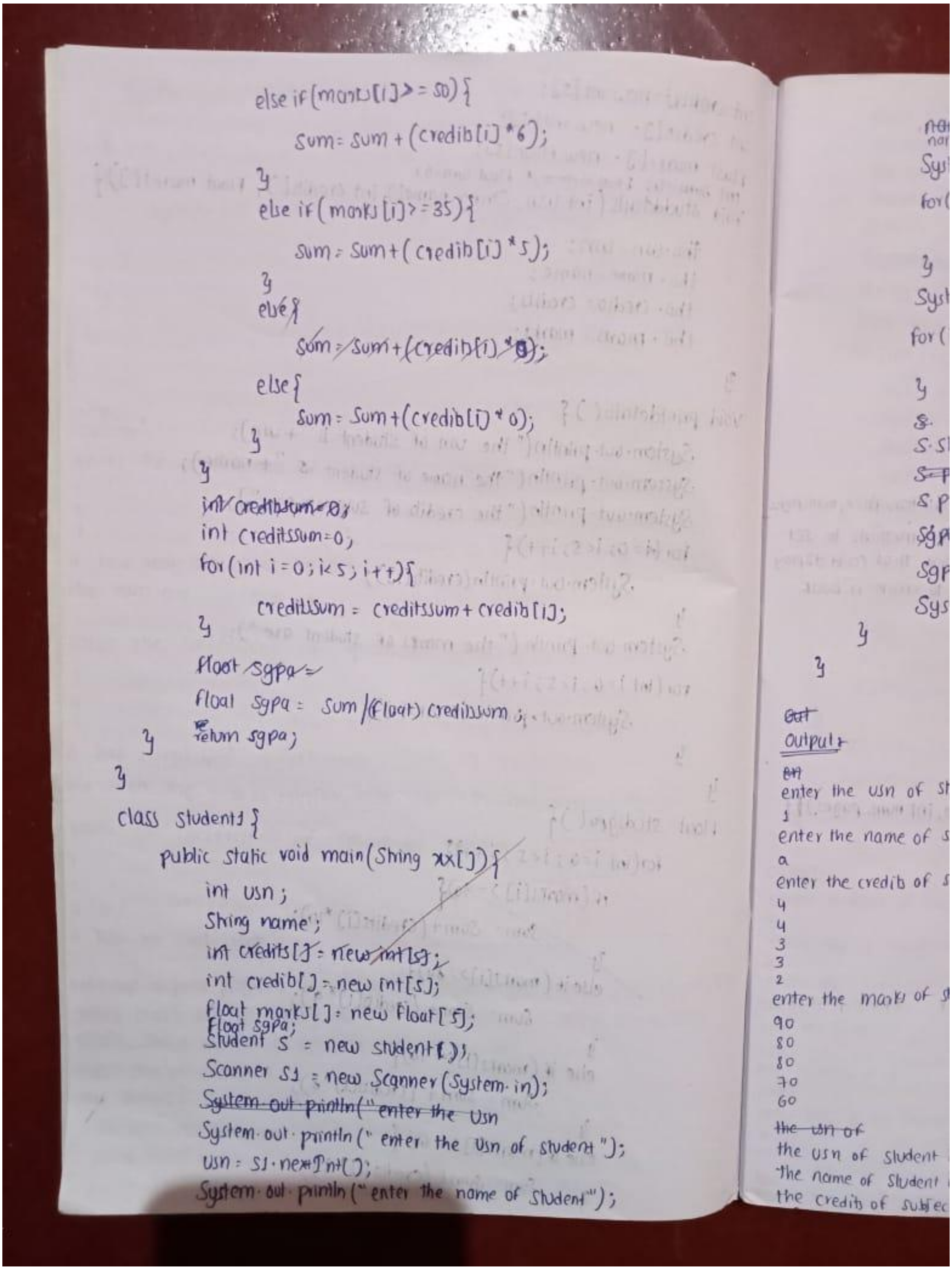
");

7568877

members usn, name, an

accept and display

```
~~int credits[]= new int~~[5];
int credits[]= new int[5];
Float marks[]= new Float[5];
~~int Sum=0; Float sum=0;~~ Float sum=0;
Void studdetails (int usn, String name[], int credits[], float marks[]){

    this.usn = usn;
    this.name = name;
    this.credits = credits;
    this.marks = marks;

}
Void printdetails(){
    System.out.println(" the usn of student is "+usn);
    System.out.println("the name of student is "+ name);
    System.out.println("the credits of subjects are");
    for (i = 0; i<5; i++){
        System.out.println(credits[i]);
    }
    System.out.println(" the marks of student are ");
    for (int i=0; i<5; i++){
        System.out.println (marks[i]);
    }
}
Float studsgpa(){
    for(int i>0; i<5; i++){
        if(marks[i]>=90){
            Sum= Sum+(Credits[i]*10);
        }
        else if (marks[i]>=80){
            Sum= Sum+ (credits[i]*9);
        }
        else if (marks[i]>=70){
            Sum= Sum+ (credits[i]*8);
        }
        else if (marks[i]>=60){
            Sum= Sum+ (credits[i]*7);
        }
```



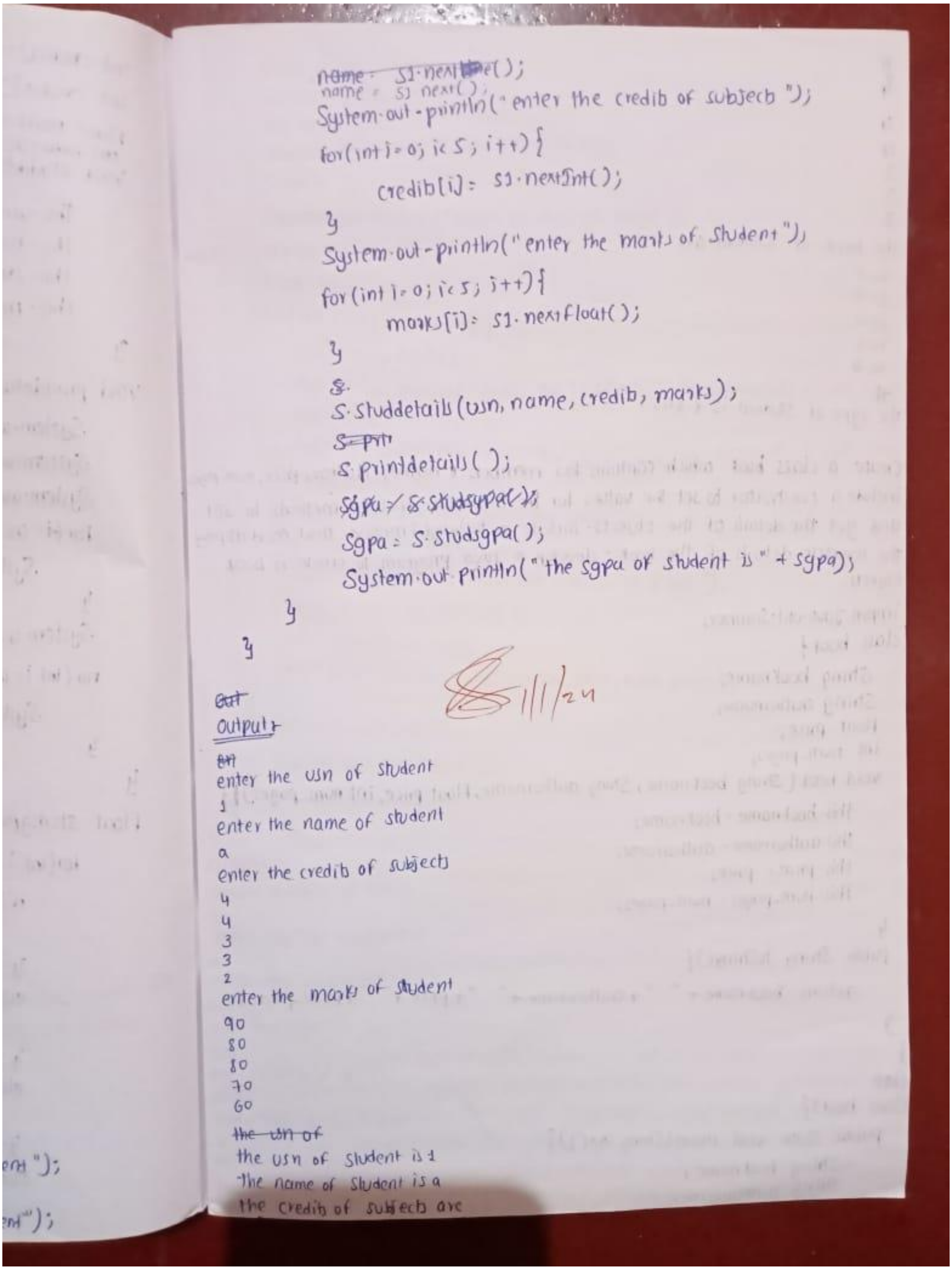
```
            else if(marks[i]>=50){
                sum=sum+(credib[i]*6);
            }
            else if(marks[i]>=35){
                sum=sum+(credib[i]*5);
            }
            else{
                ~~sum=sum+(credib[i]*0);~~
            else{
                sum=sum+(credib[i]*0);
            }
        }
        ~~int creditbsum=0;~~
        int creditssum=0;
        for(int i=0;i<5;i++){
            creditssum = creditssum+credib[i];
        }
        ~~float sgpa=~~
        float sgpa= sum/(float) creditssum;
    }   return sgpa;
}
class student1{
    public static void main(String xx[]){
        int usn;
        String name;
        ~~int credits[]=new int[5];~~
        int credib[]=new int[5];
        float marks[]=new float[5];
        float sgpa;
        student s = new student();
        Scanner s1 = new Scanner(System.in);
        ~~System.out.println("enter the usn~~
        System.out.println(" enter the usn of student ");
        usn=s1.nextInt();
        System.out.println("enter the name of student");
```

```
        ~~na~~
        na
        Sy
        for(

        }
        Sys
        for(

        }
        ~~S.~~
        S.S
        ~~S.P~~
        S.P
        sgp
        sgp
        Sys
    }
}
```

~~Out~~

Output:

```
~~en~~
enter the usn of st
1
enter the name of s
a
enter the credib of s
4
4
3
3
2
enter the marks of s
90
80
80
70
60
~~the usn of~~
the usn of student
the name of student
the credits of subjec
```



```
name = ~~s1.nextLine();~~
name = s1.next();
System.out.println("enter the credib of subjecb");
for(int i=0; i<5; i++){
      credib[i]= s1.nextInt();
}
System.out.println("enter the marks of student");
for(int i=0; i<5; i++){
      marks[i]= s1.nextFloat();
}
~~s.~~
s.studdetails(usn, name, credib, marks);
~~s.prin~~
s.printdetails( );
~~sgpa = s.studsgpa();~~
sgpa = s.studsgpa();
System.out.println("the sgpa of student is" + sgpa);
    }
}
```

1/1/24

~~Out~~

Output:-

```
~~en~~
enter the usn of student
1
enter the name of student
a
enter the credib of subjects
4
4
3
3
2
enter the marks of student
90
80
80
70
60
~~the usn of~~
the usn of student is 1
The name of student is a
the credib of subjects are
```

ent");

ent");

```
~~4~~
4
4
4
3
3
2
the marks of student are
90.0
80.0
80.0
70.0
60.0
~~#~~
the sgpa of student is 8.8125.
```

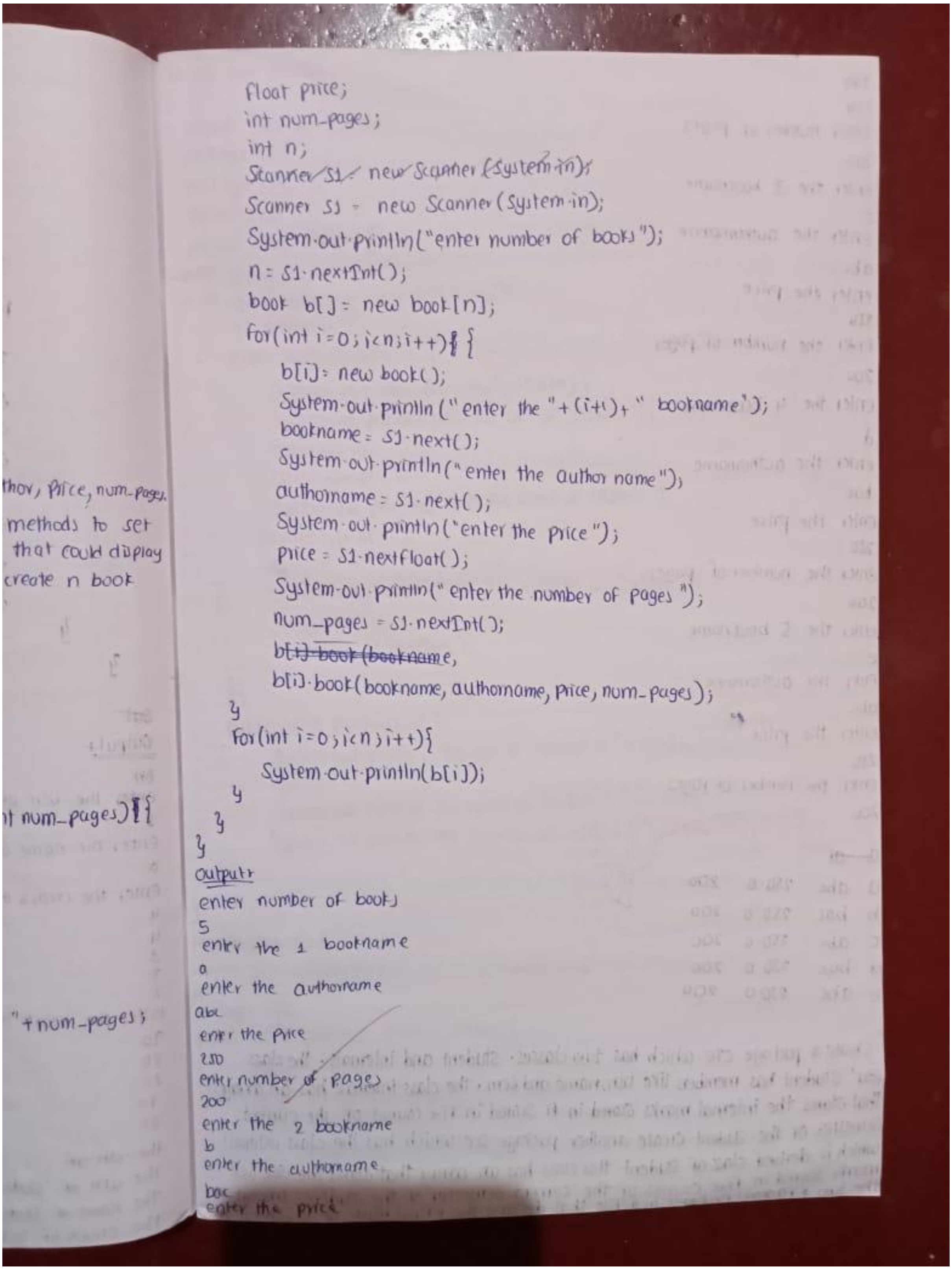
Create a class book which contains four members : name, author, Price, num_pages. include a constructor to set the values for the members. include methods to set and get the details of the objects. include a toString() method that could display the complete details of the book. develop a Java program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

```
import java.util.Scanner;
class book{
   String bookname;
   String authorname;
   Float price;
   int num_pages;
   void book(String bookname, String authorname, Float price, int num_pages){
      this.bookname = bookname;
      this.authorname = authorname;
      this.price = price;
      this.num_pages = num_pages;
   }
   public String toString(){
      return bookname + "  " + authorname + "  " + price + "  " + num_pages;
   }
}
~~class~~
class book1{
  public static void main(String xx[]){
      String bookname;
      String authorname;
```

```
      Float price
      int num_p
      int n;
      ~~Scanner S1~~
      Scanner S1
      System.out
      n = S1.next
      book b[] =
      for(int i=0;
         b[i] = ne
         System.ou
         bookname
         System.ou
         authornam
         System.ou
         price = S1.
         System.out
         num_pages
         ~~b[i] book(~~
         b[i].book(b
      }
      for(int i=0;i<n
         System.out.p
      }
   }
}
```

Output:

```
enter number of bool
5
enter the 1 booknar
a
enter the authorname
abc
enter the price
250
enter number of pages
200
enter the 2 bookname
b
enter the authorname
bac
enter the price
```



thor, Price, num_pages.
methods to set
that could display
create n book

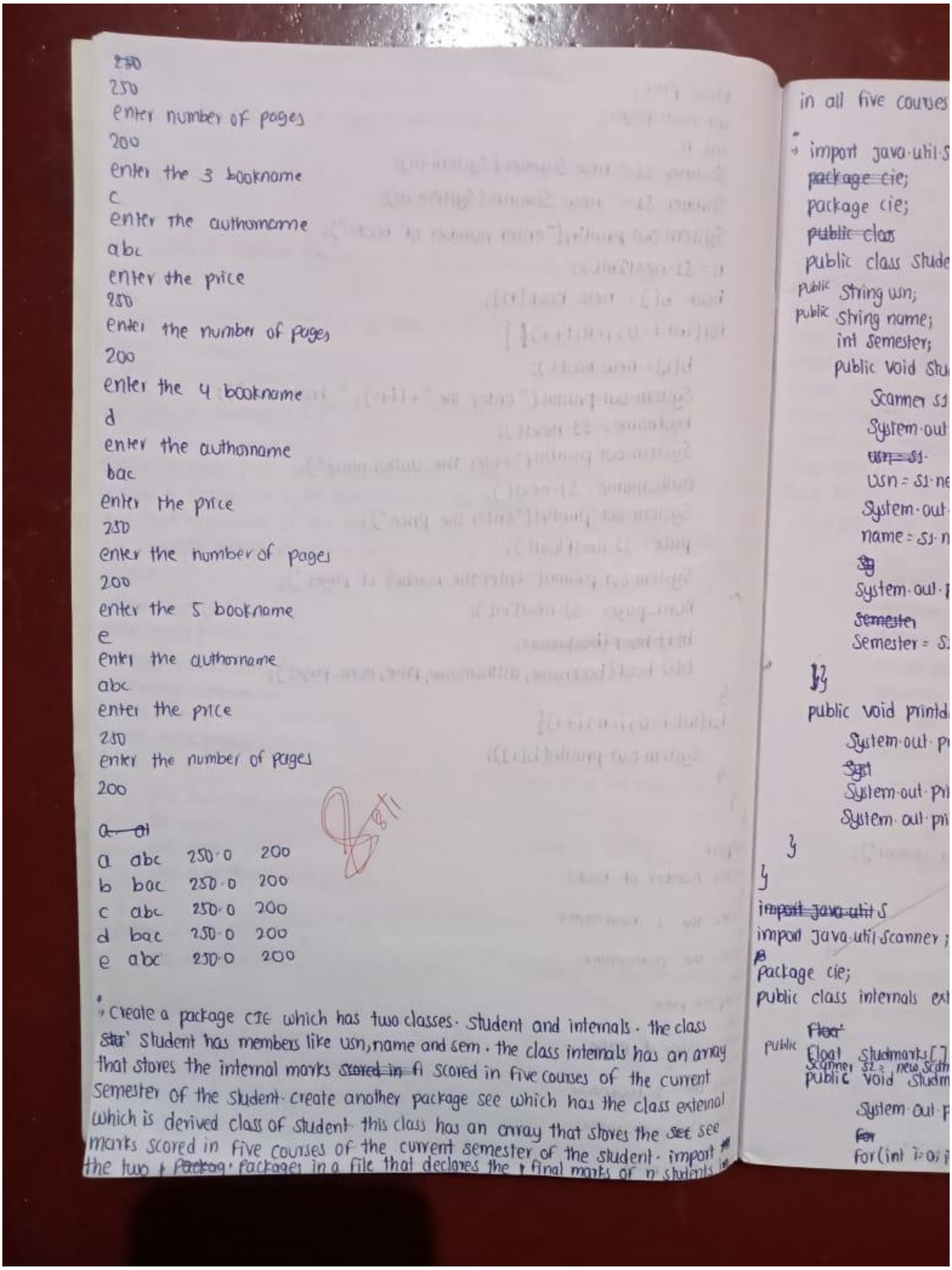
nt num_pages){

"+num_pages;

```
Float price;
int num_pages;
int n;
~~Stanner S1 = new Scanner (System.in);~~
Scanner S1 = new Scanner (System.in);
System.out.println("enter number of books");
n = S1.nextInt();
book b[] = new book[n];
For(int i=0; i<n; i++){
    b[i] = new book();
    System.out.println("enter the "+(i+1)+" bookname");
    bookname = S1.next();
    System.out.println("enter the author name");
    authorname = S1.next();
    System.out.println("enter the price");
    price = S1.nextFloat();
    System.out.println(" enter the number of pages ");
    num_pages = S1.nextInt();
    ~~b[i] book (bookname~~e,
    b[i].book(bookname, authorname, price, num_pages);
}
For(int i=0; i<n; i++){
    System.out.println(b[i]);
}
 }
}
```

Output:

```
enter number of books
5
enter the 1 bookname
a
enter the authorname
abc
enter the price
250
enter number of pages
200
enter the 2 bookname
b
enter the authorname
bac
enter the price
```



```
~~250~~
250
enter number of pages
200
enter the 3 bookname
c
enter the authorname
abc
enter the price
250
enter the number of pages
200
enter the 4 bookname
d
enter the authorname
bac
enter the price
250
enter the number of pages
200
enter the 5 bookname
e
enter the authorname
abc
enter the price
250
enter the number of pages
200

~~a a~~
a  abc  250.0  200
b  bac  250.0  200
c  abc  250.0  200
d  bac  250.0  200
e  abc  250.0  200
```

• Create a package CIE which has two classes. student and internals. the class ~~Stu~~ Student has members like usn, name and sem. the class internals has an array that stores the internal marks ~~scored in~~ fi scored in five courses of the current semester of the student. create another package see which has the class external which is derived class of student. this class has an array that stores the ~~see~~ see marks scored in five courses of the current semester of the student. import the two ~~Packag~~ packages in a file that declares the final marks of n students in all five courses

```
• import java.util.S
~~package cie;~~
package cie;
~~public clas~~
public class Stude
public String usn;
public String name;
    int semester;
    public void Stu
        Scanner s1
        System.out
        ~~usn = s1.~~
        usn = s1.ne
        System.out
        name = s1.n
        ~~Sy~~
        System.out.
        ~~Semester~~
        Semester = S
    }}
    public void printd
        System.out.p
        ~~Syst~~
        System.out.pri
        System.out.pri
    }
}
~~import java.util S~~
import java.util.Scanner;
package cie;
public class internals ext
        ~~Float~~
public  Float studmarks[]
        Scanner s2 = new Scan
        public void studm
            System.out.p
            ~~For~~
            for(int i=0;
```

internals - the class
internals has an array
uses of the current
has the class external
that stores the ~~set~~ see
the student. import ~~in~~
al marks of n students in

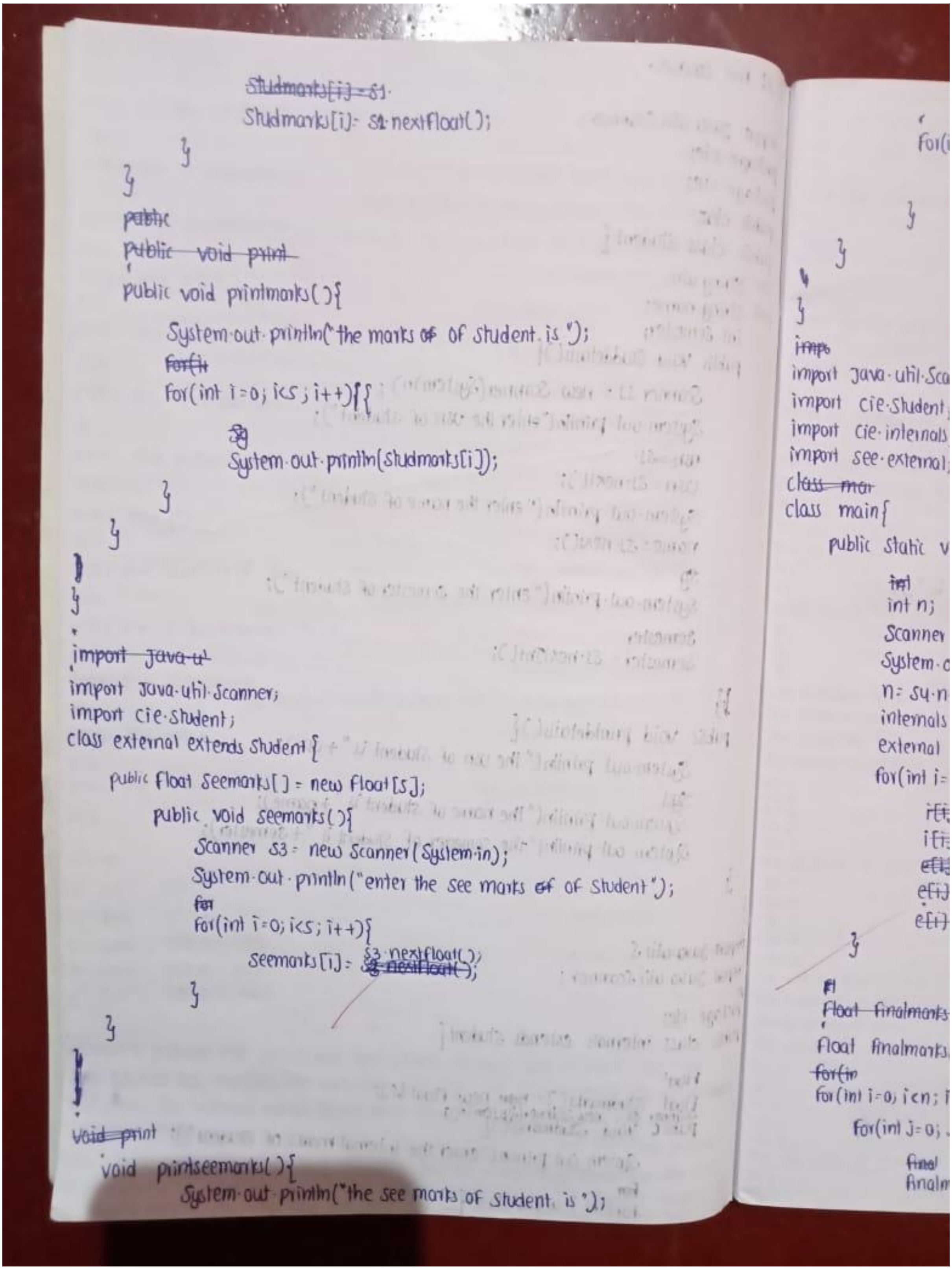
in all five courses-

```
import java.util.Scanner;
~~package cie~~;
package cie;
~~public clas~~
public class student{
public String usn;
public String name;
    int semester;
    public void Studdetails(){
        Scanner s1 = new Scanner(System.in);
        System.out.println("enter the usn of student");
        ~~usn = s1.~~
        usn = s1.next();
        System.out.println("enter the name of student");
        name = s1.next();
        ~~Sy~~
        System.out.println("enter the semester of student");
        ~~semester~~
        Semester = s1.nextInt();
    }}
    public void printdetails(){
        System.out.println("the usn of student is"+usn);
        ~~Syst~~
        System.out.println("the name of student is "+name);
        System.out.println("the semester of student is "+semester);
    }
}
~~import java.util~~ S
import java.util.Scanner;
~~B~~
package cie;
public class internals extends student{
        ~~Float~~
public  Float studmarks[] = ~~new~~ new Float [5];
        Scanner s2 = new Scanner(System.in);
        public void Studmarks(){
            System.out.println("enter the internal marks of Student");
            ~~For~~
            For(int i=0;i<5;i++){
```

in all five courses-

```
import java.util.Scanner;
~~package cie~~;
package cie;
~~public clas~~
public class student{
public String usn;
public String name;
    int semester;
    public void Studdetails(){
        Scanner s1 = new Scanner(System.in);
        System.out.println("enter the usn of student");
        ~~usn = s1.~~
        usn = s1.next();
        System.out.println("enter the name of student");
        name = s1.next();
        ~~Sy~~
        System.out.println("enter the semester of student");
        ~~semester~~
        Semester = s1.nextInt();
    }}
    public void printdetails(){
        System.out.println("the usn of student is" + usn);
        ~~Sys~~
        System.out.println("the name of student is" +name);
        System.out.println("the semester of student is" + semester);
    }
}
~~import java.util~~ S
import java.util.Scanner;
package cie;
public class internals extends student{
       ~~Float~~
public Float studmarks[] = ~~new~~ new float [5];
       Scanner s2 = new Scanner(System.in);
       public void Studmarks(){
           System.out.println("enter the internal marks of student");
           ~~for~~
           for(int i=0; i<5; i++){
```

internals. the class
internals has an array
...ses of the current
has the class external
that stores the ~~set~~ see
the student. import
al marks of n students in



```
                ~~Studmarks[i] = s1.~~
                Studmarks[i]= s1.nextFloat();
        }
    }
    ~~public~~
    ~~public void print~~
    public void printmarks(){
        System.out.println("the marks ~~of~~ of student is ");
        ~~for(i~~
        For(int i=0; i<5; i++){{
            ~~s3~~
            System.out.println(studmarks[i]);
        }
    }
}
}
import ~~java.u~~
import java.util.Scanner;
import cie.Student;
class external extends Student{
    public float seemarks[] = new float[5];
        public void seemarks(){
            Scanner s3 = new Scanner(System.in);
            System.out.println("enter the see marks ~~of~~ of student");
            ~~for~~
            for(int i=0;i<5;i++){
                seemarks[i]= s3.nextfloat();
                             ~~s3.nextFloat()~~;
            }
    }
}
~~void~~ print
    void printseemarks(){
        System.out.println("the see marks of student is ");
```

```
            For(i
            }
    }
}
~~impo~~
import java.util.Sca
import cie.Student;
import cie.internals
import see.external;
~~class mar~~
class main{
    public static v
        ~~int~~
        int n;
        Scanner
        System.o
        n= s4.n
        internals
        external
        for(int i=
            ~~i[i~~
            i[i
            ~~e[i~~
            e[i]
            e[i]
    }
    ~~F~~
    ~~Float finalmarks~~
    Float finalmarks
    ~~for(in~~
    For(int i=0; i<n; i
        For(int j=0;
            ~~final~~
            finalm
        for(int i=0; i<5; i++){
              System.out.println(seemarks[i]);
        }
    }
}
~~impo~~
import java.util.Scanner;
import cie.Student;
import cie.internals;
import see.external;
~~class mar~~
class main{
      public static void main(String xx[]){
            ~~int~~
            int n;
            Scanner sy = new Scanner(System.in);
            System.out.println("enter the number of students");
            n= sy.nextInt();
                      internalmarks[]
            internals ~~i[]~~ = new internals[n];
                      externalmarks[]
            external ~~e[]~~ = new external[n];
            for(int i=0; i<n; i++){
                  ~~i[i].studdetails();~~ internalmarks[i]= new internals();
                  ~~i[i].Studmarks();~~ externalmarks[i]= new external();
                                        internalmarks[i].studdetails();
                  ~~e[i]~~              internalmarks[i].studmarks();
                  ~~e[i].seemc~~        externalmarks[i].seemarks();
                  ~~e[i].seemarks();~~
            }

            ~~Fl~~
            ~~Float finalmarks[][]~~ =
            Float finalmarks[][]= new float[n][5];
            ~~for(in~~
            for(int i=0; i<n; i++){
                  for(int j=0; j<5; j++){
                        ~~final~~                internalmarks[i]              externalmarks[i]
                        finalmarks[i][j] = ~~i[i]~~.studmarks[j]+((~~e[i]~~.seemarks[j])/2);
        }
        }
}
~~for(int i=0;~~
for(int i=0; i<n; i++){
        internalmarks[i]. printdetails
        ~~i[i] printdetails(~~);
        internalmarks[i] printmarks( )
        ~~i[i] printmarks( )~~;
        externalmarks[i]. printseemarks( );
        ~~e[i] printseemarks( );~~
}
~~for(int i=0~~
for(int i=0; i<n; i++){
    ~~Syst~~
    System-out-println("the final marks of " + i + " student is ");
    ~~For~~
    For(int j=0;j<5;j++){
        ~~System-out-println(~~
        System-out-println(finalmarks[i][j]);
    }
  }
 }
}
}
```

output:

```
enter the number of students
2
enter the usn ~~of student~~
a
enter the usn of student
1
enter the name of student
a
enter the semester of student
3
enter the internal marks of student
40
40
40
40
40
enter the see mar
80
80
80
80
80
enter the usn of stu
2
enter the name of s
b
enter the semester of
3
~~enter the internalma~~
enter the internal m
45
45
45
45
45
enter the see marks of s
80
80
80
80
80
usn of student is 1
the name of student is
the semester of studen
the marks of student is
40.0
40.0
40.0
40.0
40.0
the see marks of student
80.0
80.0
80.0
80.0
80.0
the usn of student is 2
the name of student is b
the semester of student i
the marks of student is
45.0
45.0
45.0
45.0
45.0
the see marks of student i
80.0
80.0
```

```
40
40
40
40
enter the see marks of student
80
80
80
80
80
enter the usn of student
2
enter the name of student
b
enter the semester of student
3
```

~~enter the internalmarks of student~~

```
enter the internal marks of student
45
45
45
45
45
enter the see marks of student
80
80
80
80
80
usn of student is 1
the name of student is a
the semester of student is 3
the marks of student is
40.0
40.0
40.0
40.0
40.0
the see marks of student is
80.0
80.0
80.0
80.0
80.0
the usn of student is 2
the name of student is b
the semester of student is 3
the marks of student is
45.0
45.0
45.0
45.0
45.0
the see marks of student is
80.0
80.0
80.0
80.0
80.0
the finalmarks of 0 student is
80.0
80.0
80.0
80.0
80.0
the finalmarks of 1 student is
85.0
85.0
85.0
85.0
85.0
```

write a program that ~~dem~~ demonshates handling of exceptions in inheritence tree. create a base class called ~~father~~ " father and derived class son ~~whic~~ which ext the base class. ~~in~~ in father class ~~implement~~ impliment a constructor ~~whi~~ which ta the age and throws the ~~exce~~ exception ~~w~~ wrongage( ) when the ~~input~~ input age in ~~son~~ son class impliment a ~~constructor~~ constructor that cases both father and son's age and throws an ~~exception~~ exception if ~~son's~~ son's age is >= father's age

```
~~class wronga~~
~~class wrongage~~
                         Exception
                         ~~exceptioninheritancedemo~~ {
class wrongage extends ~~exception~~
        ~~wrongage(String m~~
    void wrongage (String message){
                ~~Supe~~
                Super(message);
        }
}
class father{
    private int age;
    public father(int age) throws wrongage {
        if(age <0){
            throw new wrongage("age cannot be negative");
        }
        this.age = age;
    }
    public int getage( ){
        return age;
    }
}
~~class son~~
~~class son~~
class son extends father{
    private int ~~sona~~ sonage ;
    public son(int fatherage , int sonage) throws wrongage{
        Super(fatherage);
```

```
        if(
        if(

        }
        this.s
    }
    ~~public int~~
    public Int
        return
    }
}
public class exce
    ~~publi~~
    public stati
        try{
            S
            S
            S
            S
            S
            Sy
            Son
            ~~Sys~~
            Sys
        }
        catch(
        
    }
  }
}
```

nitence tree.
e which extends
~~whi~~ which takes
~~input~~ input agesco
th father and
= father's age.

");

e{

```
        ~~if(sonage>=~~
        if(sonage>=fatherage){

              throw new wrongage("son's age cannot be greater than or
                    equal to father's age");

        }
        this.sonage = sonage;

    }
    ~~public int~~
    public int getsonage(){
        return sonage;
    }
}

public class exceptioninheritancedemo{

      ~~publi~~
      public static void main(string xx[]){

            try{
                ~~Son s~~
                ~~Son Son~~
                Son Son = new son(40,20);
                System.out.println("Father's age :" + son.getage());
                ~~Sy~~
                System.out.println("Son's age :"+ son.getsonage());
                Son invalidson = new son(30,35);
                ~~Syste~~
                System.out.println("this line will not be ~~reache~~ reached");
            }
            catch(wrongage e){
                  ~~So~~
                  ~~System~~                                      e.getMessage()
                  System.out.println("exception :" + ~~e.getme~~ ~~e.getmessaget~~ ));
            }
      }
}
```

Output:

```
father's age : 40
~~So~~
~~Son's~~
Son's age : 20
~~exception: son's Son's a~~
exception: Son's age cannot be greater than or ~~equal~~ equal to father's age.
```

if(.getmessage() is not used:

```
exception: wrongage: son's age cannot be greater than or equal to Father's ~~age~~ age
```

1. Create two packages - one named personal and other ~~named~~ named professional. the personal package has a class person ~~with~~ with details - name, age, address, emailid. this includes methods to accept and display ~~data~~ data from the user. the professional package has a class manager which is a subclass of person with details - id, deptid, qualification, salary. this includes methods to accept and display data from the user. ~~in~~ Import these two packages in a file which ~~cre~~ creates n objects of the manager class. print the names of the managers who are above 50 years of age and who earn above 50000.

```
package personal;
import java.util.Scanner;
public class person {
        String name;
        int age;
        String address;
        String emailid;
        ~~public void person~~
        public void persondetails(){
            Scanner s1 = new Scanner(System.in);
            System.out.println("enter the name of person");
            name= s1.next();
            System.out.println("enter the age of person");
            age= s1.nextInt();
            System.out.println("enter the address of person");
            address= s1.next();
            ~~System~~
            System.out.println("enter the emailid of person");
            emailid= s1.next();
    }
    public void printdetails(){
```

```
Sy
Sys
Sys
~~Sy~~
Sys
    }
;
}
~~pack~~
package professio
import java.util.S
import personal.pe
public class man
        int id;
        int dept
        String
        ~~double~~
        double s
        public v
            Scan
            ~~Sy~~
            Syster
            id= S2
            System
            ~~deptid~~
            deptid=
            System
            qualifica
            ~~System~~
            System.o
            Salary = S
    }
    public void p
        System.
        System
        System.
        System.
        item2[1].itemdetails(2,"b",250,2,2);
        item2[1].sell();
    }
    catch(reorder e){
        System.out.println("exception: "+e.getMessage());
    }
  }
}
```

output:

```
exception: quantity cannot be less than reorder ~~level~~. level.
```

Shape:

```
abstract class shape{
    ~~protect~~
    protected int dimension1;
    ~~protecte~~
    protected int dimension2;
    public shape(int dimension1, int dimension2){
        this.dimension1 = dimension1;
        this.dimension2 = dimension2;
    }
    public abstract void printArea();
}
class rectangle extends shape {
    ~~public~~
    public rectangle(int length, int width){
        Super(length, width);
    }
    public void printArea(){
        int area = dimension1 * dimension2;
        system.out.println("area of rectangle: "+area);
    }
}
class triangle extends shape {
    public triangle(int base, int height){
```

```
        Supe
    }
    public void
        double
        ~~System.~~
        System.
    }
}
class circle exte
    ~~pu~~
    public c
        Su
    }
    public voi
        double
        System
    }
}
public class re
    public stat
        recta
        ~~tri~~
        triang
        ~~circle~~
        circle
        r.prin
        t.prin
        c.prin
    }
}
```

Output:

```
area of rectangle:
area of triangle:
~~area of circle:~~
area of circle: 78.
```

```
        super(base, height);
    }
    public void printarea(){
        double area = 0.5* dimension1* dimension2;
        ~~System.out.println("~~
        System.out.println("area of triangle : "+area);

    }
}
class circle extends shape{

     ~~pt~~
     public circle(int radius){
          super(radius, 0);
     }
     public void printarea(){
          double area = Math.PI * dimension1 * dimension1;
          ~~S~~
          System.out.println("area of circle :"+area);
     }
}
public class rect{
     public static void main(string xx[]){

          rectangle r = new rectangle(2,10);
          ~~tr~~
          triangle t = new triangle(3,7);
          ~~circle c = new~~
          circle c = new circle(5))
          r.printarea();
          t.printarea();
          c.printarea();
     }
}
```

Output:

```
area of rectangle : 20
area of triangle : 10.5
~~area of circle :~~
area of circle : 78.53918
```

Output:

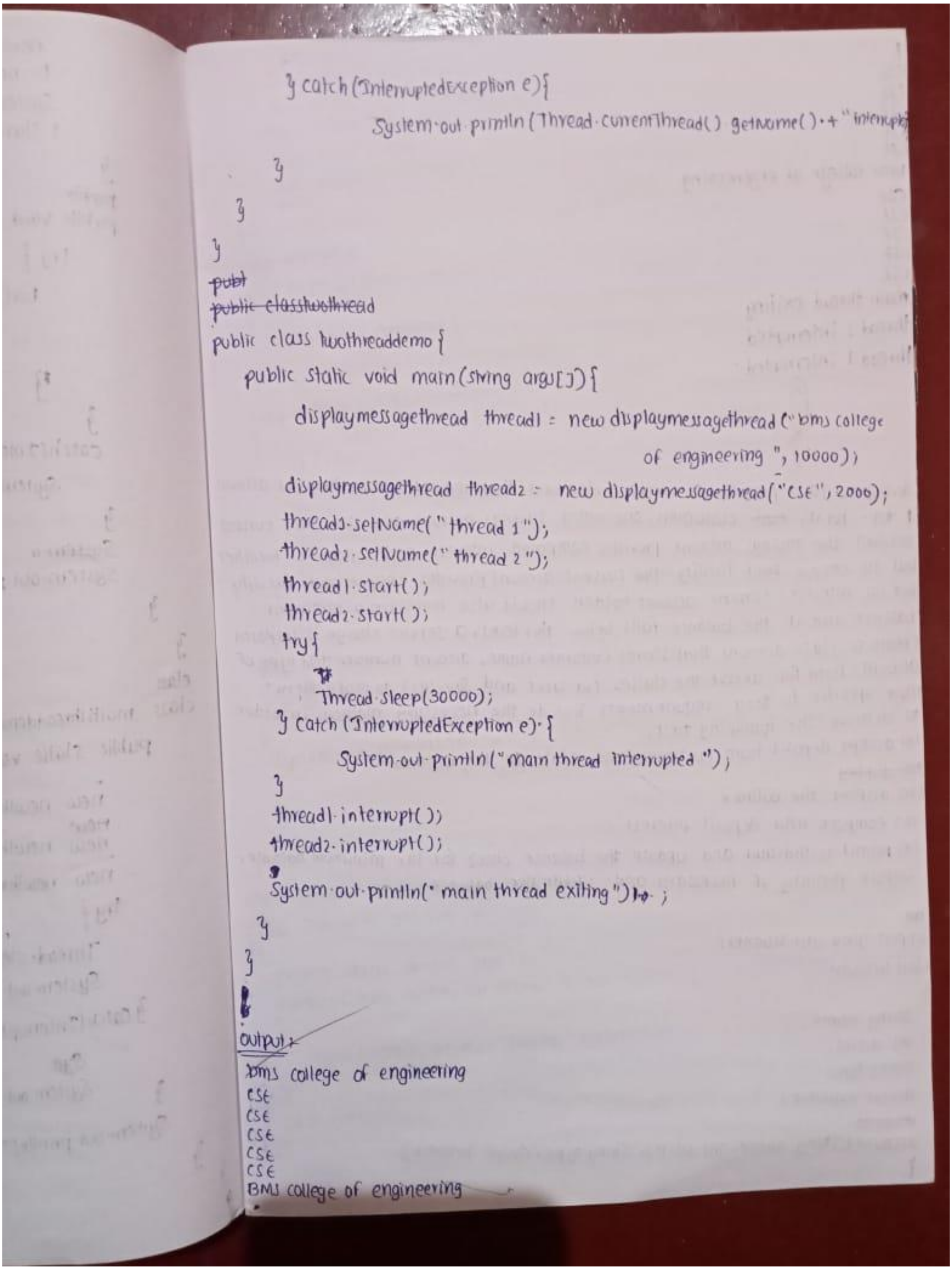
```
~~New thread : Thread[#21,one,5,main]~~
~~New thread :~~
new thread : Thread[#21,one,5,main]
new thread : Thread[#22,two,5,main]
new thread : Thread[#23,three,5,main]
two:5
three:5
one:5
two:4
three:4
one:4
two:3
one:3
three:3
three:2
one:2
two:2
~~one~~
one:1
three:1
two:1
three exiting
two exiting
one exiting
~~main thread is aw~~
main thread is aware
main thread exiting
```

```
~~class displaymessagethread extends th~~
class displaymessage thread extends Thread {
        private final String message;
        private final long interval;
        displaymessagethread(String message, long interval): {
                ~~this.messa~~
                this.message = message;
                this.interval = interval;
        }
        public void run(){
            try{
                while(true){
                    System.out.println(message);
                    Thread.sleep(interval);
                }
```

```
            } catch
            }
        }
}
~~publ~~
~~public classtwothrea~~
public class twothr
    public static v
        displaym

        displaymess
        thread1.set
        thread2.set
        thread1.sta
        thread2.sta
        try{
            Thread.s
        } catch (Inter
            System
        }
        thread1.interru
        thread2.interru
        System.out.print
    }
}
```

output:

```
~~bms college of engin~~
CSE
CSE
CSE
CSE
CSE
BMS college of engine
```



```
        } catch(InterruptedException e){
                System.out.println(Thread.currentThread() getName().+ "interrupt
        }
    }
}
~~publ~~
~~public classtwothread~~
public class twothreaddemo {
    public static void main(string args[]){
        displaymessagethread thread1 = new displaymessagethread ("bms college
                                                  of engineering ", 10000);
        displaymessagethread thread2 = new displaymessagethread("CSE",2000);
        thread1.setName("thread 1");
        thread2.setName("thread 2");
        thread1.start();
        thread2.start();
        try{
            Thread.sleep(30000);
        } catch (InterruptedException e).{
            System.out.println("main thread interrupted");
        }
        thread1.interrupt();
        thread2.interrupt();
        System.out.println("main thread exiting") ;
    }
}
```

output:

```
bms college of engineering
CSE
CSE
CSE
CSE
CSE
BMS college of engineering
```

```
}
CSE
CSE
CSE
CSE
CSE
bms college of engineering
CSE
CSE
CSE
CSE
CSE
main thread exiting
thread 2 interrupted
thread 1 interrupted
```

develop a Java program to create a class bank that maintains two kinds of account ~~for~~. for it's ~~cust~~ customer, one called savings account and the other current account. the savings account provides compound interest and withdrawl facilities but no cheque book facility. the current account provides cheque book facility but no interest. current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed create a class account that stores customer name, account number and type of account. from this derive the classes cur-acct and sav-acct to make them more specific to their requirements. include the necessary methods in order to acheive the following tasks:

(a) accept deposit from customer and update the balance

~~(b) display~~

(b) display the balance

(c) compute and deposit interest

(d) permit withdrawl and update the balance check for the minimum balance, impose penalty if necessary and update the balance

```
~~im~~
import java.util.Scanner;
class account
{
  String name;
  int accno;
  String type;
  double balance;
  ~~account~~
account(String name, int accno, String type, double balance)
{
```

```
      {
        this.name
        this.accno
        this.type
        this.balance

    }
    ~~void deposit(~~
    void deposit(do
    {
       balance += a
    }
    void withdraw(c
    {
       if((balance
       {
          balance -
       }
       else{
          System.o
       }
    }
   void display()
   {
    System.out.printl

    }
  }
~~class Savacct exten~~
class savacct exter
{
  private static doub
  Savacct(String name
  {
    super(name, a
  }
  void interest()
  {
   balance += balance
   }
```

s two kinds of account
and the other current
nd withdrawl facilities
cheque book facility
tain a minimum
ice charge is imposed
number and type of
ct to make them
ary methods in order

ance

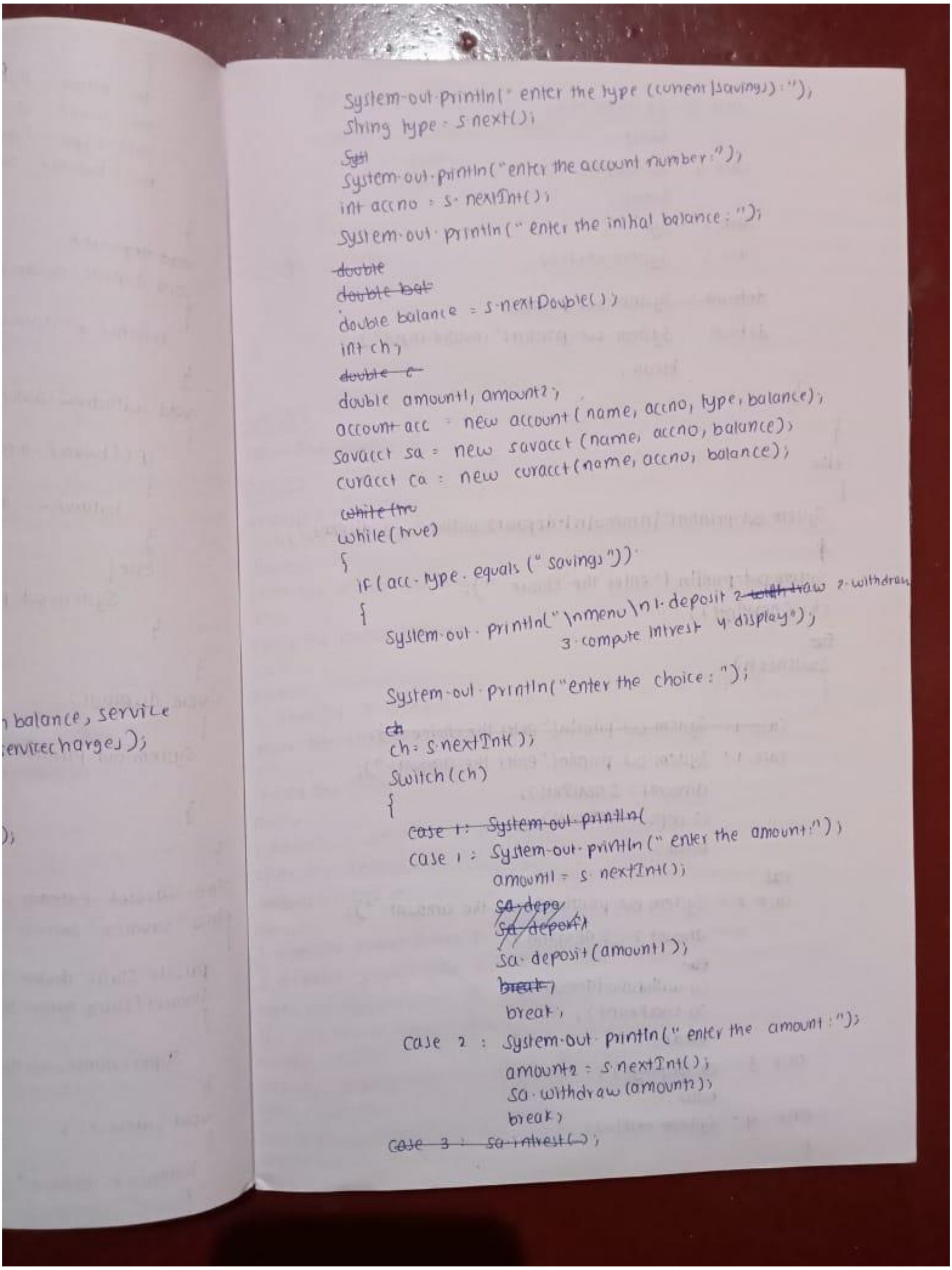
the minimum balance,

lance)

```
    {
     this.name = name;
     this.accno = accno;
     this.type = type;
     this.balance= balance;

}
~~void deposit(~~
void deposit (double amount)
{
   balance += amount;
 }
void withdraw (double amount)
{
     if((balance -amount)>=0)
     {
        balance -= amount;
     }
     else{
         System.out.println("insufficient balance, cant withdraw");
     }
 }
void display()
{
  System.out.println("name:"+ name +" accno:" + accno +" type:" +type
                            + " balance:" + balance);

   }
}
~~class savacct extends account~~
class savacct extends account
{
  private static double rate =5;
  savacct (String name, int accno, double balance)
  {
     super(name, accno, "savings",balance);
  }
  void interest( )
  {
   balance += balance * (rate)/100;
```

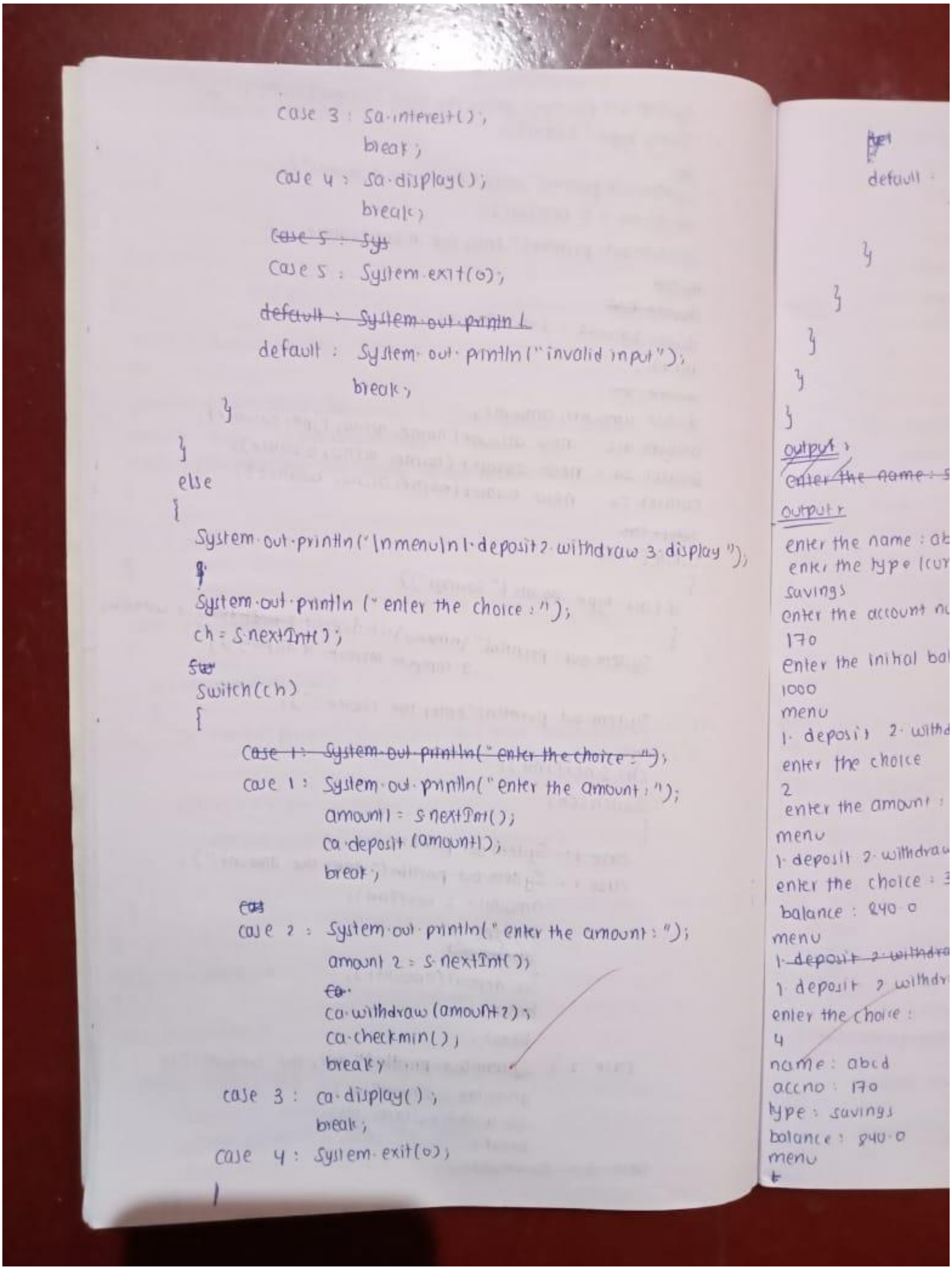
```
            System.out.println(" balance : " +balance);
    }
    }
}
~~class cur~~
class curacct extends account
{
  private double minbal = 500;
  ~~pen~~
  ~~private double servicecha~~
  private double servicecharges = 50;
  curacct(String name, int accno, double balance)
  {
     ~~sup~~
     super(name, accno, "current", balance);
  }
  }
  void checkmin()
  {
    if(balance < minbal)
    {
       System.out.println(" balance is less than min balance, service
                          charges imposed : "+ servicecharges);
       balance -= servicecharges;
       System.out.println(" balance is :" + balance);
    }
  }
}
~~class accountmain~~
~~{~~
class accountmain
{
  public static void main(String xx[]){
     Scanner s = new Scanner(system.in);
     System.out.println(" enter the name : ");
     String name = s.next();
     l
```

```
 System
 String
 ~~Syst~~
 System
 int acc
 System
 ~~double~~
 ~~double b~~
 double b
 int ch;
 ~~double c~~
 double a
account a
Savacct s
curacct c
~~while (tru~~
while(tru
{
 if(acc
 {
   Syste
   Syste
   ~~ch~~
   ch = S
   Switc
   {
    ~~case~~
    case
   case
~~case 3~~
```



```
n balance, service
ervicecharges);

);
```

```
System.out.println(" enter the type (current|savings):");
String type = s.next();
~~Sys~~
System.out.println("enter the account number:");
int accno = s.nextInt();
System.out.println(" enter the initial balance: ");
~~double~~
~~double bal~~
double balance = s.nextDouble();
int ch;
~~double c~~
double amount1, amount2;
account acc = new account ( name, accno, type, balance);
savacct sa = new savacct (name, accno, balance);
curacct ca = new curacct(name, accno, balance);
~~while tru~~
while(true)
{
  if(acc.type.equals(" savings"))
  {
    System.out.println("\nmenu\n1.deposit 2 ~~with draw~~ 2.withdraw
                      3.compute intrest 4.display");

    System.out.println("enter the choice: ");
    ~~ch~~
    ch= s.nextInt();
    Switch(ch)
    {
      ~~case 1: System.out.println(~~
      case 1 : System.out.println(" enter the amount:");
               amount1 = s.nextInt();
               ~~sa.depo~~
               ~~sa.deposit~~
               sa.deposit(amount1);
               ~~break;~~
               break;
     case 2 : System.out.println(" enter the amount :");
               amount2 = s.nextInt();
               sa.withdraw(amount2);
               break;
    ~~case 3 : sa.intrest(~~);
```



```
        case 3 : sa.interest();
                 break;
        case 4 : sa.display();
                 break;
        ~~case 5 : sys~~
        case 5 : System.exit(0);
        ~~default : System.out.println~~
       default : System.out.println("invalid input");
                 break;
    }
}
else
{
  System.out.println("\nmenu\n1.deposit 2.withdraw 3.display");
  System.out.println("enter the choice :");
  ch = s.nextInt();
  ~~sw~~
  switch(ch)
  {
      ~~case 1: System.out.println("enter the choice :");~~
      case 1 : System.out.println("enter the amount :");
               amount1 = s.nextInt();
               ca.deposit(amount1);
               break;
      ~~ca~~
      case 2 : System.out.println("enter the amount :");
               amount 2 = s.nextInt();
               ~~ca.~~
               ca.withdraw(amount2);
               ca.checkmin();
               break;
    case 3 : ca.display();
             break;
   case 4 : System.exit(0);
```

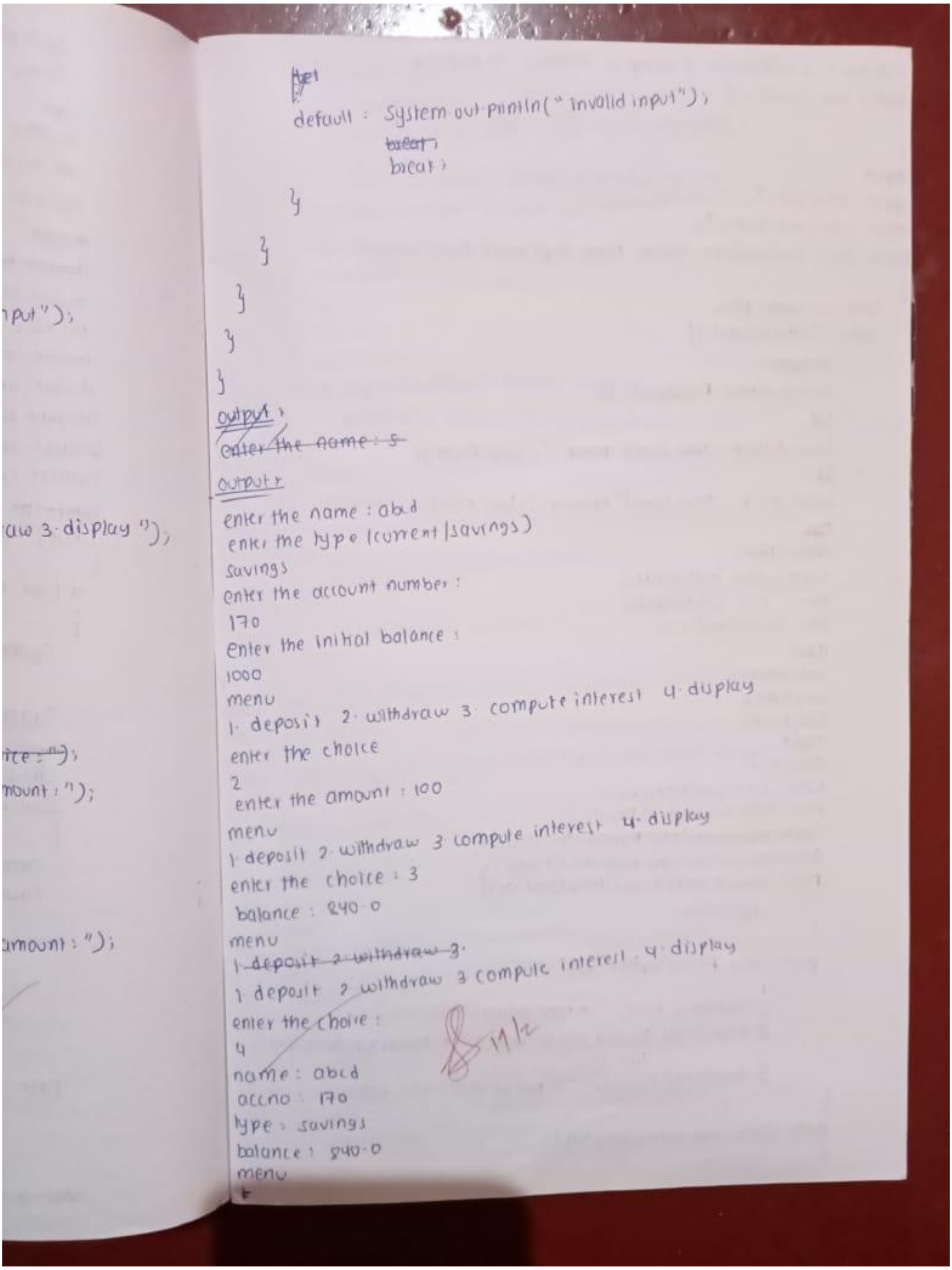
```
        default :

        }
      }
    }
  }
}
```

~~output :~~

~~enter the name : s~~

output :

```
enter the name : ab
enter the type (cur
savings
enter the account n
170
enter the initial ba
1000
menu
1. deposit 2. withd
enter the choice
2
enter the amount :
menu
1. deposit 2. withdraw
enter the choice : 3
balance : 840.0
menu
~~1. deposit 2. withdra~~
1. deposit 2. withdr
enter the choice :
4
name : abcd
accno : 170
type : savings
balance : 840.0
menu
```



```
nput");
aw 3. display ");
~~ice:");~~
mount:");
amount:");
```

```
            ~~bre~~
            default : System.out.println(" invalid input");
                        ~~break;~~
                        break;
            }
        }
    }
  }
}
```

~~output:~~
~~enter the name: s~~

output:

```
enter the name : abcd
enter the type (current/savings)
savings
enter the account number :
170
enter the initial balance :
1000
menu
1. deposit  2. withdraw 3. compute interest  4. display
enter the choice
2
enter the amount : 100
menu
1. deposit 2. withdraw 3 compute interest  4. display
enter the choice : 3
balance : 840.0
menu
~~1. deposit 2. withdraw 3.~~
1 deposit  2 withdraw 3 compute interest . 4. display
enter the choice :
4
name : abcd
accno : 170
type : savings
balance : 840.0
menu
```

19/2

```
1 deposit 2 withdraw 3 compute interest 4 display
enter the choice: 5
```

```
1
~~import~~
import java.awt.*;
import java.awt.event.*;
public class TextfieldDemo extends Frame implements Actionlistener
{
    Textfield name, pass;
    public TextfieldDemo(){
        ~~Setlayout~~
        Setlayout(new flowlayout());
        ~~Lab~~
        Label namep = new Label("Name: ", Label.RIGHT);
        ~~La~~
        Label passp = new Label("Password:", Label.RIGHT);
        ~~Nam~~
        ~~name = new~~
        name = new Textfield(12);
        pass = new Textfield(8);
        pass.setEchoChar('?');
        ~~add(~~
        add(namep);
        add(name);
        add(passp);
        ~~add(p~~
        add(pass);
        name.addActionListener(this);
        pass.addActionListener(this);
        ~~addWindowlistener(new MyWindowAdapter());~~
        addWindowlistener(new MyWindowAdapter());
        public void actionPerformed(ActionEvent ae){
            repaint();
        }
        public void paint(Graphics g){
            1
            g.drawString("Name: " + name.getText(), 100,200);
            g.drawString("selected text in name : " + name.getSelectedText(),
                100, 220);
            g.drawString("Password : " + pass.getText(), 100, 240);
    }
    public static void main(String xx[])
    {
```

```
        1
        ~~TextfieldDe~~
        TextfieldDe
        ~~a~~
        awin.setsiz
        awin.setTi
        ~~awin.setVi~~
        awin.setVis
    }
}
class MyWindowAd
    public void u
        Syste
    }
}
```

Output

```
name: [a|bc]        pass
name : abc
Select text in name: bc
pasword: aaa
```

```
1
import java.awt.*;
import java.awt.event.*;
~~P~~
public class DivisionMain1 ext
{
    Textfield num1, num2;
    Button dResult;
    Label outResult;
    String out = " ";
    ~~doub~~
    double resultNum;
    int Flag = 0;
    ~~public DivisionMain1()~~
    public DivisionMain1()
    {
        Setlayout(new Flowl
        ~~dResult =~~
        ~~dR~~
        dResult = new Button
        Label number1 = new
        |
        ~~TextfieldDemo~~
        TextfieldDemo  awin = new TextfieldDemo();
        ~~a~~
        awin.setsize(new Dimension(700,700));
        awin.setTitle("TF_label demo"),
        ~~awin.setvisible(~~
        awin.setvisible(true);
    }
}
class MyWindowAdapter extends WindowAdapter {
    public void windowClosing(WindowEvent we){
            System.exit(0);
    }
}
```

Output

name: a|bc   password: ???

name : abc

Select text in name: bc

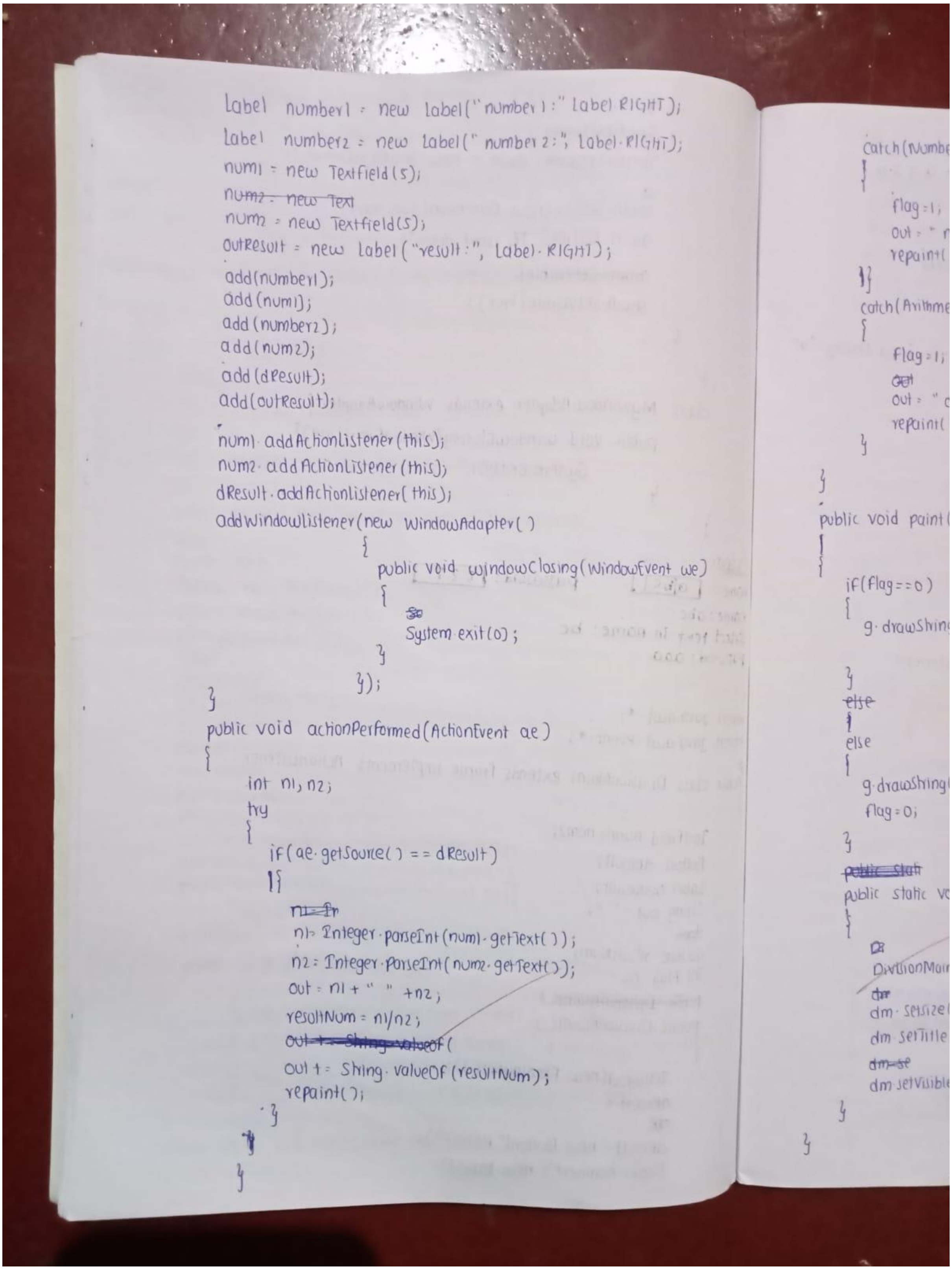
password: aaa

```
|
import Java.awt.*;
import Java.awt.event.*;
~~P~~
Public class DivisionMain1 extends frame implements Actionlistener
{
    Textfield num1, num2;
    Button dResult;
    Label outResult;
    String out = " ";
    ~~doub~~
    double resultNum;
    int flag = 0;
    ~~public DivisionMain1( )~~
    public DivisionMain1( )
    {
        Setlayout(new Flowlayout( ));
        ~~dResult =~~
        ~~dR~~
        dResult = new Button(" RESULT");
        ~~Label number1 = new Label("~~
```

);

.getSelectedText(),

240);



```
Label number1 = new label("number 1:" Label RIGHT);
Label number2 = new Label(" number 2:", Label.RIGHT);
num1 = new Textfield(5);
~~num2 = new Text~~
num2 = new Textfield(5);
outResult = new Label ("result:", Label. RIGHT);
add(number1);
add(num1);
add (number2);
add(num2);
add (dResult);
add(outResult);

num1.addActionListener(this);
num2.addActionListener(this);
dResult.addActionlistener( this);
addWindowlistener(new WindowAdapter()
                {
                  public void windowClosing(WindowEvent we)
                  {
                      ~~So~~
                      System.exit(0);
                  }
                });
}
public void actionPerformed(ActionEvent ae)
{
    int n1, n2;
    try
    {
      if(ae.getSource() == dResult)
      {
        ~~n1 = Fr~~
        n1= Integer.parseInt(num1.getText());
        n2= Integer.ParseInt(num2.getText());
        out = n1 + " " +n2;
        resultNum = n1/n2;
        ~~out += String.valueOf~~(
        out += String. valueOf (resultNum);
        repaint();
      }
    }
    }
```

```
    catch(Numbe
    {
        flag=1;
        out = " r
        repaint(
    }
    catch(Arithme
    {
        Flag=1;
        ~~Out~~
        out = " c
        repaint(
    }
}
public void paint(
{
    if(flag==0)
    {
      g.drawStrin

    }
    ~~else~~
    }
    else
    {
      g.drawString(
      flag=0;
    }
    ~~public stati~~
    public static vo
    {
        ~~Di~~
        DivisionMai
        ~~dm~~
        dm.setSize(
        dm.setTitle
        ~~dm.se~~
        dm.setVisible
    }
}
HT);
HT);
wEvent we)
```

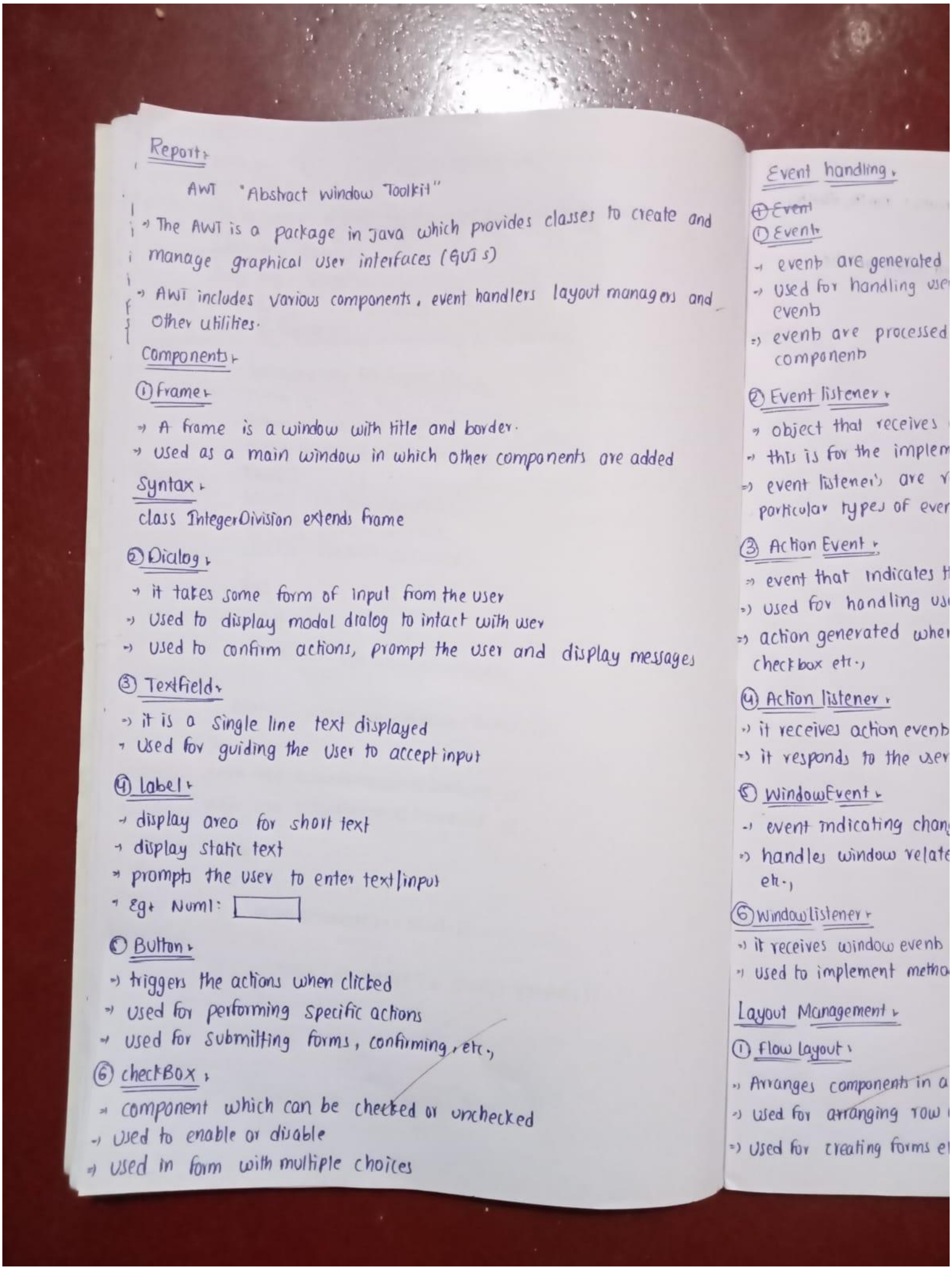
```
catch(NumberformatException e1)
{
{
    flag=1;
    Out = " number format exception! " +e1
    repaint();
}}
catch(ArithmeticException e2)
{
    Flag=1;
    ~~Out~~
    out = " divide by 0 exception! " +e2;
    repaint();
  }

}
public void paint(Graphics g)
{
{
  if(flag==0)
  {
    g.drawString(out, outResult.getx() + outResult.getwidth(), outResult.getY()
                    + outResult.getHeight()-8));
  }
  ~~else~~
  ~~{~~
  else
  {
    g.drawString(out, 100, 200);
    flag=0;
  }
  ~~public stati~~
  public static void main(String xx[])
  {
    ~~D~~
    DivisionMain1 dm = new DivisionMain1();
    ~~dm~~
    dm.setsize(new Dimension(800, 400));
    dm.setTitle(" DivisionOfIntegers ");
    ~~dm.se~~
    dm.setVisible(true);
  }
}
```

Output :

```
Number 1: 4          number 2 :  2   [result]  result : 4 2 2.0

number 1 : 4         number 2 : 0    [result]  result :

divide by 0 Exception! Java.lang. ArithmeticException : / by zero

number 1 : 4         number 2 : a    [result]  result :

number format exception! Java.lang.NumberFormatException : For input string: "a"
```

```
import Java.awt.*;
import Java.awt.event.*;
public class DivisionMain extends frame implements ActionListener
{
      Textfield num1, num2;
      Button dResult;
      label outResult;
      String out = "";
      double resultNum;
      int flag = 0;

      public DivisionMain()
      {
            setlayout(new Flowlayout());
            dResult = new Button("RESULT");
            label number1 = new label("Number 1:", label.RIGHT);
            label number2 = new label("Number2:", label.RIGHT);
            num1 = new Textfield(5);
            num2 = new Textfield(5);
            outResult = new label("Result: ", label.RIGHT);
            add(number1);
            add(num1);
            add(number2);
```

```
add(num2);
add(dResul
~~add(outRes~~
add(outResu
num1.addAct
num2.addAc
dResult.addA
addwindowlis

      }
public void a
{
      double n
      try
      {
        if(ae
        {
          n1=
          ~~n2=~~
          n2=
          out
          res
          ~~out~~
          out
          rep
      }
}
catch(Arithmet
{
    Flag = 1;
    out = " Di
    repaint()
}
```



Report:

AWT "Abstract window Toolkit"

- The AWT is a package in Java which provides classes to create and manage graphical user interfaces (GUI s)
- AWT includes various components, event handlers layout managers and other utilities.

Components:

① Frame:

- A frame is a window with title and border.
- used as a main window in which other components are added

Syntax:

class IntegerDivision extends Frame

② Dialog:

- it takes some form of input from the user
- Used to display modal dialog to intact with user
- Used to confirm actions, prompt the user and display messages

③ Textfield:

- it is a single line text displayed
- Used for guiding the User to accept input

④ label:

- display area for short text
- display static text
- prompts the user to enter text/input
- Eg: Num1: □

⑤ Button:

- triggers the actions when clicked
- used for performing specific actions
- used for submitting forms, confirming, etc.,

⑥ checkBox:

- component which can be checked or unchecked
- used to enable or disable
- used in form with multiple choices

Event handling:

~~① Event~~

① Event:

- event are generated
- used for handling use events
- events are processed components

② Event listener:

- object that receives
- this is for the implem
- event listener's are r particular types of ever

③ Action Event:

- event that indicates t
- used for handling us
- action generated whe checkbox etc.,

④ Action listener:

- it receives action events
- it responds to the user

⑤ WindowEvent:

- event indicating chan
- handles window relate etc.,

⑥ Windowlistener:

- it receives window events
- used to implement metho

Layout Management:

① Flow layout:

- Arranges components in a
- used for arranging row
- Used for creating forms e

create and

managers and

are added

play messages

Event handling :

~~① Event~~

① Events

-> events are generated by user actions

-> used for handling user input such as mouse clicks, keypress or window events

=> events are processed by event listeners attached to the relevant components

② Event listener :

-> object that receives and handles events

-> this is for the implementation of event driven behaviour in GUI's

=> event listener's are registered with specific components to listen for particular types of events

③ Action Event :

-> event that indicates that a component - defined action occured

=> used for handling user actions

=> action generated when user clicks on ~~cor~~ component like button, checkbox etc.,

④ Action listener :

=> it receives action events

=> it responds to the user action

⑤ WindowEvent :

-> event indicating change to the state of a window

=> handles window related events such as window opening, closing, resizing etc.,

⑥ WindowListener :

=> it receives window events

-> used to implement methods to handle window-related events

Layout Management :

① Flow layout :

=> Arranges components in a centered flow one after another

=> used for arranging row or column with equal spacing between them.

=> used for creating forms etc.,

② Border-Layout :

- Layout manager that divides the contamer into 5 regions: north, south, east, west and center
- used for arranging components at the ~~edge~~ edges and center of the container

Layout manager :

- responsible for arranging components in container

~~co~~

- control size and position of components.