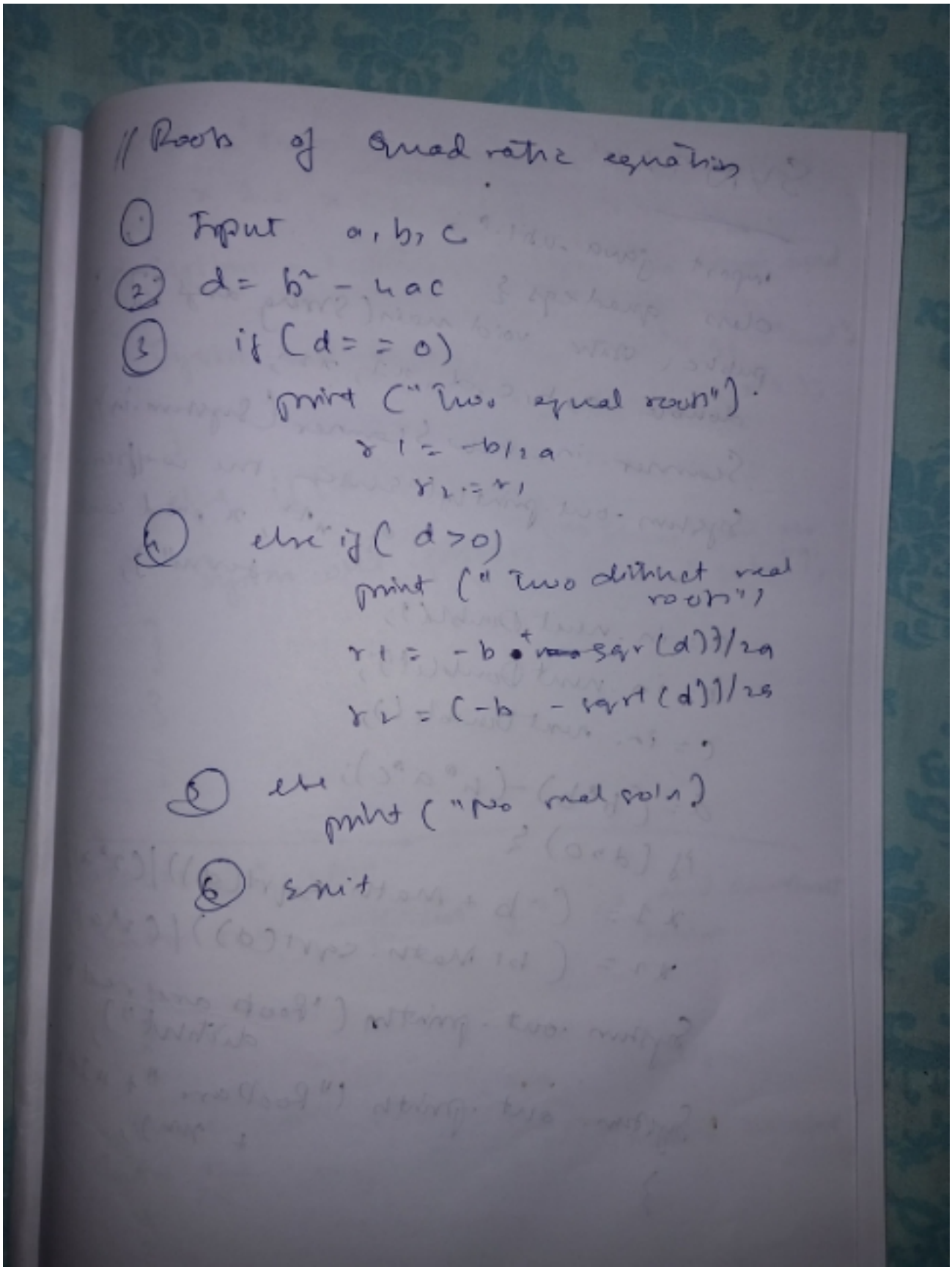
1	I	Develop a Java program that prints all real solutions to the quadratic equation $ax^2+bx+c = 0$. Read in a, b, c and use the quadratic formula. If the discriminate b^2-4ac is negative, display a message stating that there are no real solutions.



// Roots of quadratic equation

1. Input a, b, c
2. d = b^2 - 4ac
3. if (d == 0)
   print ("Two equal roots").
   r1 = -b/2a
   r2 = r1
4. else if (d > 0)
   print ("Two distinct real roots")
   r1 = (-b + sqrt(d))/2a
   r2 = (-b - sqrt(d))/2a
5. else
   print ("No real roots")
6. exit

SYNTAX:

```
import java.util.*;
class quadeqs {
public static void main(String args[]) {
 double a, b, c, d, x1, x2;
 Scanner in = new Scanner(System.in);
 System.out.println("Enter the coefficient
                     of x^2, x and constant
                     term");
 a = in.nextDouble();
 b = in.nextDouble();
 c = in.nextDouble();
 d = (b*b) - (4*a*c);
 if (d>0) {
 x1 = (-b + Math.sqrt(d)) / (2*a);
 x2 = (-b - Math.sqrt(d)) / (2*a);
 System.out.println("Roots are real &
                     distinct");
 System.out.println("Roots are " + x1 + " and "
                     + x2);
 }
```

```
else if (d == 0){
    x1 = x2 = (-b/(2*a));
System.out.println("Roots are real and
                    equal");
System.out.println("Roots are "+x1+"and"+
                    x2);
}
else if(d<0){
  System.out.println("There are no
                    real solns");
}
}
}
```

EXPECTED OUTPUT:

```
Enter the coefficient of x^2, x, and constant
                    term
1
2
3
There are no real solutions
Enter the coefficient of x^2, x, and constant
1
2
1
Roots are real and equal
    Roots are -1.0 and -1.0
```

~~Import~~

Enter the coefficients of x^2, x, and constant term

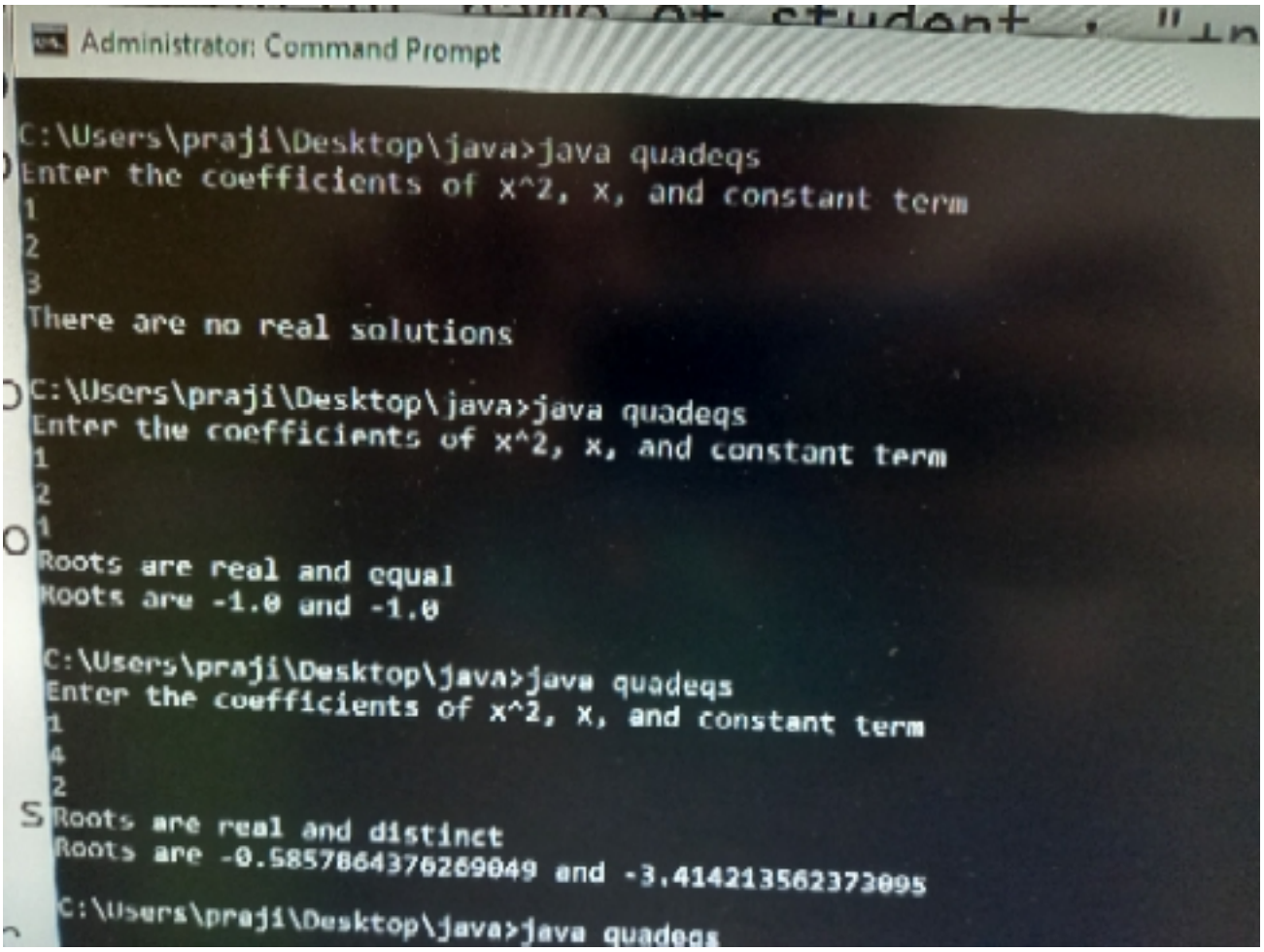
1

4

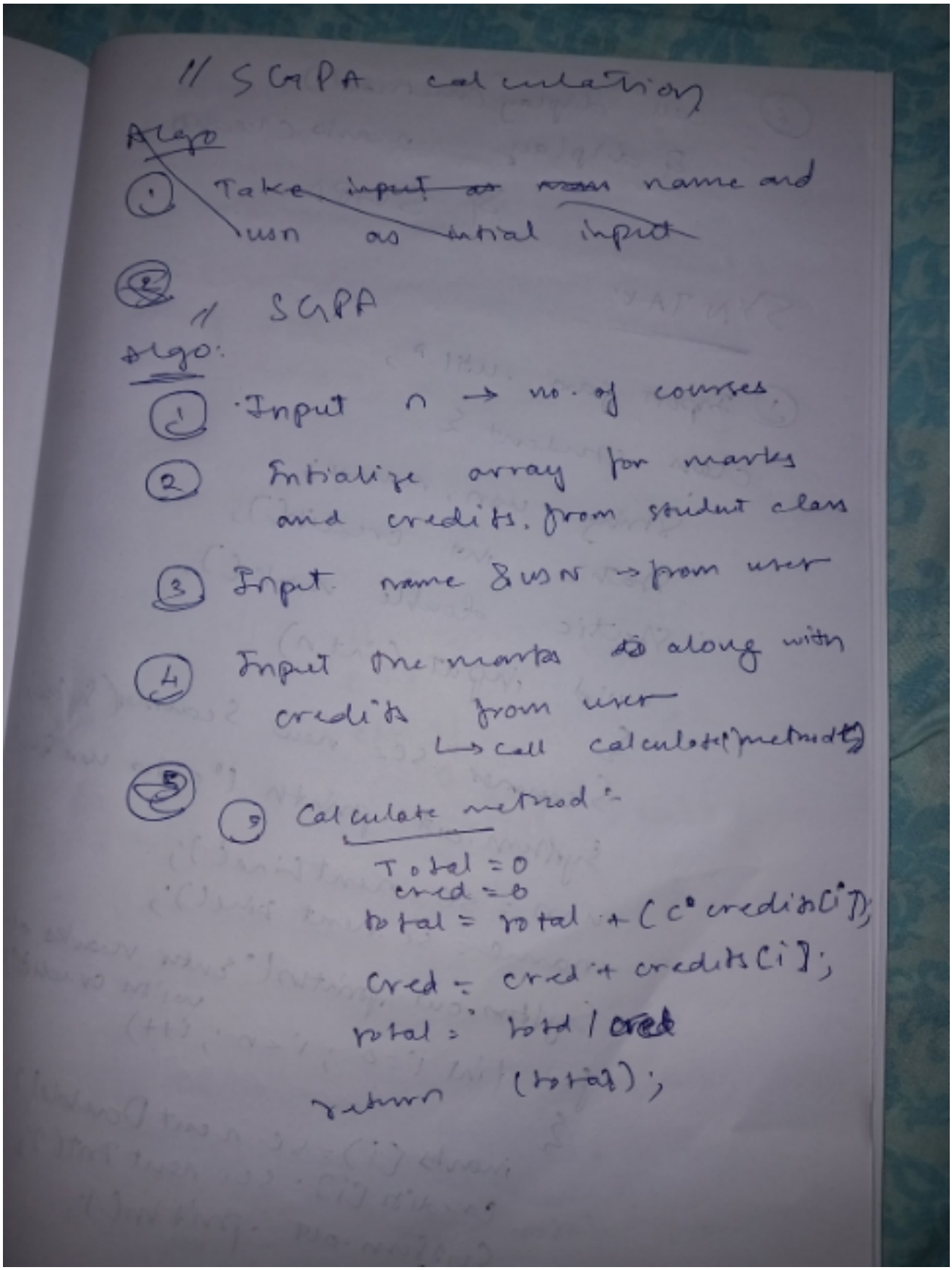
2

Roots are real and distinct

Roots are -0.585 and -3.4142


Administrator: Command Prompt
C:\Users\praji\Desktop\java>java quadeqs
Enter the coefficients of x^2, x, and constant term
1
2
3
There are no real solutions
C:\Users\praji\Desktop\java>java quadeqs
Enter the coefficients of x^2, x, and constant term
1
2
1
Roots are real and equal
Roots are -1.0 and -1.0
C:\Users\praji\Desktop\java>java quadeqs
Enter the coefficients of x^2, x, and constant term
1
4
2
Roots are real and distinct
Roots are -0.5857864376269049 and -3.4142135623730951
C:\Users\praji\Desktop\java>java quadeqs

2	II	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.



// SGPA calculation

Algo

1. Take name and usn as initial input

2. // SGPA

Algo:

1. Input n → no. of courses
2. Initialize array for marks and credits. from student class
3. Input name & USN → from user
4. Input the marks along with credits from user
   ↳ call calculate() method
5. Calculate method :-

```
Total = 0
cred = 0
total = total + (c* credits[i]);
cred = cred + credits[i];
total = total / cred
return (total);
```

(6) Call display (method)
& display marks, credits
& SGPA.

SYNTAX:

```
(1) Import java.util.*;
class student {
    String usn, name;
    Static int credits[];
    Static double marks[];
    void input (int n)
    {
    Scanner sc= new Scanner(System.in);
    System.out.println("enter usn & name");
    usn = sc.nextLine();
    name = sc.nextLine();
    System.out.println("Enter marks along
                        with credits");
    for (int i=0 ; i<n ; i++)
    {
        marks[i] = sc.nextDouble();
        credits[i] = sc.nextInt();
        System.out.println();
    }
    }
```

```
double calculate(int n)
{
    int c, cred = 0;
    double tot, total = 0.0;
    for (int i = 0; i < n; i++)
    {
        tot = marks[i];
        if (tot >= 90)
            c = 10;
        else if (tot >= 80)
            c = 9
        else if (tot >= 70)
            c = 8
        else if (tot >= 60)
            c = 7
        else if (tot >= 50)
            c = 6
        else if (tot >= 40)
            c = 4
        else
            c = 0;
        total = total + (c * credits[i]);
        cred = cred + credits[i];
    }
    total = total/cred;
    return total;
}
```

```
void display(int n, double total)
{
System.out.println("name of student:"+
                                name);
System.out.println("no of subject:"+no);
System.out.println("marks of student with
                        Credits of course");
for(int i=0;i<n;i++)
{
System.out.println(marks[i]+" "+
                        credits[i]);
}
System.out.println("sgpa of student:"
                        +total);
}
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
student obj = new student();
System.out.println("enter no. of course");
int n = sc.nextInt();
credits = new int[n];
marks = new double[n];
obj.input(n);
double total = obj.calculate(n);
```

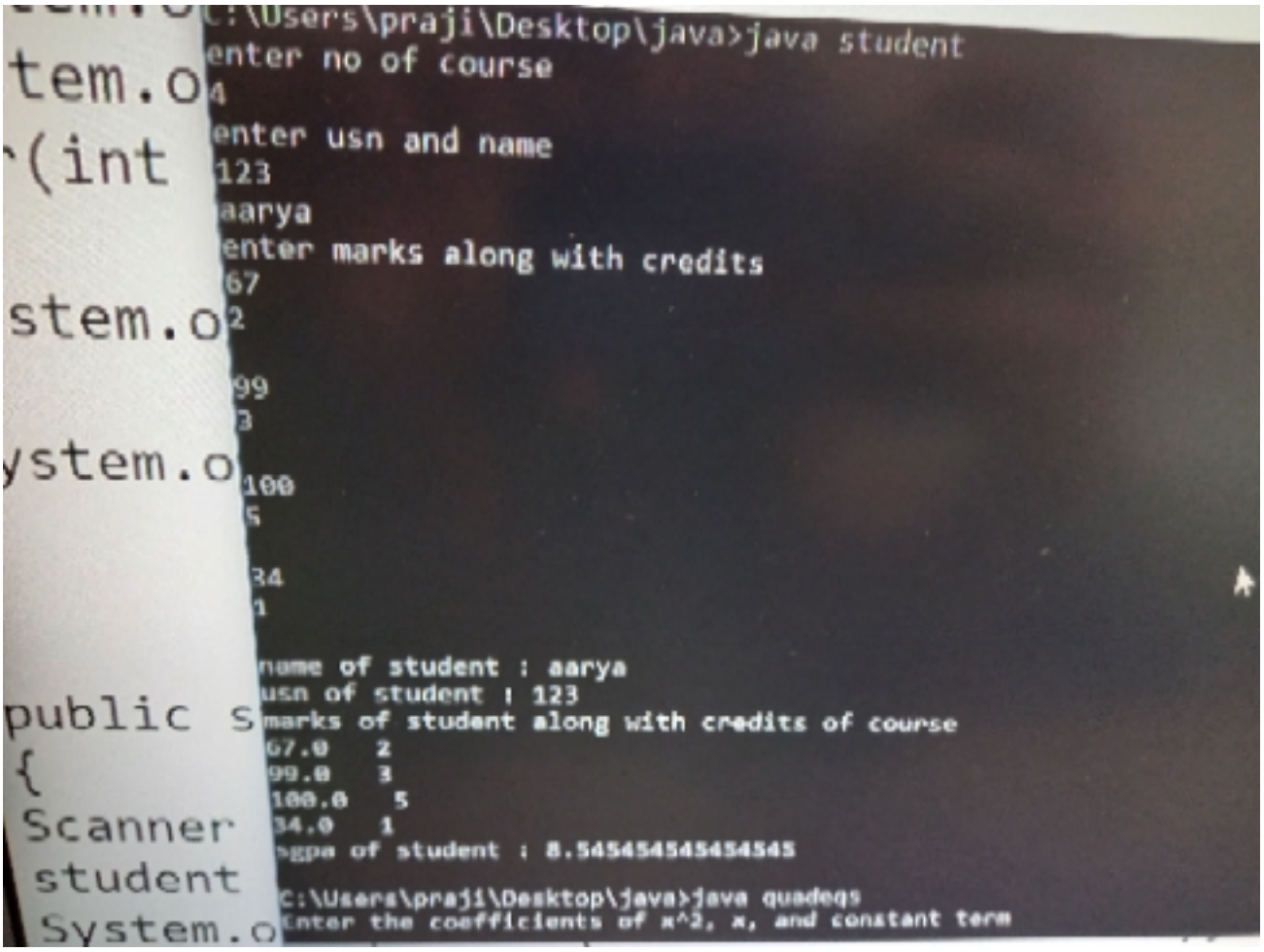
```
        obj.display(n,total);
    }
}
```

EXP OUTPUT:

```
Enter no. of course
4
enter usn and name
123
aarya
enter marks along with credits
67
2
89
3
100
5
34
1
name of student : aarya
usn of student : 123
marks of student along with
    credit of course
    67.0  2
    89.0  3
    100.0 5
```

34.01

sgpa of student :- 8.5454.


tem.o
(int
stem.o
ystem.o
public s
{
Scanner
student
System.o
C:\Users\praji\Desktop\java>java student
enter no of course
4
enter usn and name
123
aarya
enter marks along with credits
67
2
99
3
100
5
34
1
name of student : aarya
usn of student : 123
marks of student along with credits of course
67.0 2
99.0 3
100.0 5
34.0 1
sgpa of student : 8.545454545454545
C:\Users\praji\Desktop\java>java quadeqs
Enter the coefficients of x^2, x, and constant term

3	**II**	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.

// Array of objects

Algorithm:

Step 1:
Create 'Book' class with specified instance variables & include constructor method Book() { }

Step 2:
Input n -> number of books.
& create object b1 of Book class
Book b1 = new Book();

Step 3:
create array of objects of Book class
Book[] arr = new Book[5];

Step 4:
Initialize each array element as an object & call getDetails method
arr[i] = new Book();
arr[i].getDetails();

Step 5:
Print details of books by using toString() method.

// Syntax:

```
import java.util.*;

class Book {
    String name;
    String author;
    double price;
    int num_pages;

Book(){}

public String toString(){
return("name :" + name + "author :" + author + "price:"
        + price + "no. of pages :" + num_
                                    Pages);
}

void getDetails(){
    Scanner in = new Scanner(System.in);
    System.out.println("Enter the name of ~~author~~ book:");
    name = in.next();
    System.out.println("Enter the price of book:\n");
    price = in.nextDouble();
    System.out.println("Enter name of author:\n");
    author = in.next();
    System.out.println("Enter no. of pages");
    num_pages = in.nextInt();

}
}
```

```
class BookArrayObj {
    public static void main (String args[]) {
        int n;
        String name;
        String author;

        Scanner in = new Scanner (System.in);
        System.out.println ("Enter the no. of books");
        n = in.nextInt();
        Book b1 = new Book();
        Book [] arr = new Book[n];
        for (int i=0 ; i<n; i++){
            arr[i] = new Book();
            arr[i].getDetails();
        }

        System.out.println ("The Details of books");
        for (int i=0 ; i<n ; i++ ){
            System.out.println (arr[i]);
        }
    }
}
```

```
C:\Users\praji\Desktop\java>java BookArrayObj
Enter the number of books:

2
Enter the name of   book:

atals
Enter the name of author:

ayn
Enter the price of book:

400.4576
Enter the number of pages of book:

450
Enter the name of   book:

name
Enter the name of author:

marcus
Enter the price of book:

45.0
Enter the number of pages of book:

760
The Details of the  books  are  :

 name : atals |  author : ayn | price : 400.4576 | number of pages : 450

 name : name |  author : marcus | price : 45.0 | number of pages : 760
```

4	III	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.

```
// Area using Abstract class
import java.util.*;
abstract class Shape{
    int a = 7;
    int b = 6;
abstract int printArea();
}

class Rectangle extends Shape{
    int printArea(){
        System.out.println("Area of rectangle is"
                + (a*b));

        return 0;

    }

}

class RightTriangle extends Shape{
    int printArea(){
    System.out.println("Area of right triangle is"+
                ((int)a*b*0.5));

        return 0;
    }

}
```

```
class Circle extends Shape{
    int printArea(){
    System.out.println("Area of rectangle is
                        + ((int)3.14*a*a));

    return 0;
    }
}
class Area{
public static void main(String args[]){
    Rectangle r = new Rectangle();
    RightTriangle t = new RightTriangle();
    Circle c = new Circle();
    Shape s;
    s = r;
    s.printArea();
    s = t;
    s.printArea();
    s = c;
    s.printArea();
    }
}
```

```
C:\Users\praji\Desktop\java>java Area
Area of rectangle is 42
Area of right triangle  is 21.0
Area of rectangle is 147
```

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 Curr-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.

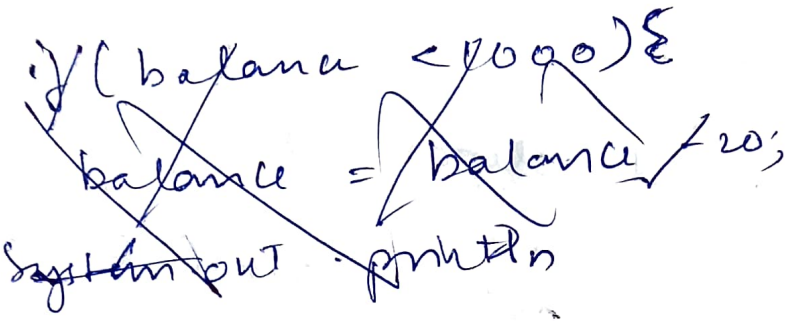
// Bank Account (Inheritance)

```
import java.util.*;
class Account{
    Scanner in = new Scanner(System.in);
    String cust_name;
    int acc_no;
    public int balance;
    int type_account;
    public int amount;

public void withdraw(){
    System.out.println("Enter ur account
                        balance:");
    balance = in.nextInt();
System.out.println("Enter the amount to be
                    withdrawn:");
    amount = in.nextInt();
        if(amount > balance){
    System.out.println("Insufficient Balance");
      display();
        }
    else {
      balance = balance - amount;
```

```
balance = balance - 20;
display();
}
}
}
public void deposit()
{
System.out.println("Enter ur acc balance");
balance = in.nextInt();
System.out.println("Enter the amount you
want to deposit:");
amount = in.nextInt();
balance = balance + amount;
display();
}
public void display()
{
System.out.println("your Account balance
is : " + balance);
}
}
```

```
class Current extends Account
{
                    withdraw
public void ~~Deposit~~()
{
```



```
if(balance <10000){
    balance = balance -20;
System.out.println
System.out.println("Enter ur account bal:");
Balance = in.nextInt();
System.out.println("Enter the amount to
                    be withdrawn:");
amount = in.nextInt();
    if (amount> balance{
System.out.println("Insufficient
                    Balance");
    display();
      }
    else {
    balance: balance - amount;
     if (balance < 10000){
        balance = balance-20;
System.out.println(" Service charge
    have been applied due
    to ~~less~~ bal less
        than min bal");
```

```
        displayCI();
        }
    }
}
class Saving extends Account {
public void computeInterest()
{
    int t;
    System.out.println("Enter ur ac bal:");
    balance = in.nextInt();
    System.out.println("Enter no time:");
    t = in.nextInt();
    balance = balance * (1+2*t);
    displayCI();
}
}
class Bank {
public static void main(String args[]){
    int choice;
Account a = new Account();
Current c = new Current();
Saving s = new Saving();
Scanner in = new Scanner(System.in);
```

```
for(;;){
System.out.println("Enter ur name:\n");
a.cust_name = in.next();
System.out.println("Enter ur Account:");
a.acc_no = in.nextInt();
System.out.println("Choose type of account:\n
                    1.Savings \n 2.Current");
a.type_account = in.nextInt();

System.out.println("Choose which operation
                    u want to perform:\n");
System.out.println("1.deposit \n 2.withdrawal
                    \n 3.deposit Interest
                    (only for savings)");

choice = in.nextInt();

switch(choice){
    case 1: if(a.type_account==2){
                c.deposit();
            }
            else{
                s.deposit();
            }
            break;
```

```
case 2: if (a.type-amount == 2) {
            c.withdraw();
            c.Penalty();
        }
        else {
            s.withdraw();
        }
        break;
case 3: s.computeInterest();
        break;
default: System.exit(0);
```

```
Enter ur name:

asd
Enter ur Account number:

123
Choose type of account :
 1.Savings
 2.Current

2
Choose which operation u want to perform:

 1.deposit
2.withdrawal
3.deposit Interest(Only for savings account)
2
Enter ur account balance:
900
Enter the amount u want to withdraw:
2
your Account balance is : 898
Balance is less than min bal hence service charge has been applied
your Account balance is : 878
Enter ur name:

1
Enter ur Account number:

123
Choose type of account :
 1.Savings
 2.Current

1
Choose which operation u want to perform:

 1.deposit
2.withdrawal
3.deposit Interest(Only for savings account)
1
Enter ur account balance:
789
Enter the amount u want to deposit:
34
your Account balance is : 823
Enter ur name:

asd
Enter ur Account number:
```

```
Enter ur Account number:

123
Choose type of account :
 1.Savings
 2.Current

1
Choose which operation u want to perform:

 1.deposit
2.withdrawal
3.deposit Interest(Only for savings account)
1
Enter ur account balance:
789
Enter the amount u want to deposit:
34
your Account balance is : 823
Enter ur name:

asd
Enter ur Account number:

123
Choose type of account :
 1.Savings
 2.Current

1
Choose which operation u want to perform:

 1.deposit
2.withdrawal
3.deposit Interest(Only for savings account)
3
Enter ur account balance:
3456
Enter the time interval:

2
your Account balance is : 17280
Enter ur name:
```