VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JnanaSangama", Belgaum -590014, Karnataka.

LAB REPORT
on

OBJECT ORIENTED JAVA PROGRAMMING

Submitted by

Yapara karthikeya (1BM21CS249)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

B.M.S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
Oct 2022-Feb 2023

B. M. S. College of Engineering,

Bull Temple Road, Bangalore 560019

(Affiliated To Visvesvaraya Technological University, Belgaum)

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Lab work entitled "**OBJECT ORIENTED JAVA PROGRAMMING**" carried out by **Yapara karthikeya(1BM21CS249),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of Java Lab **- (22CS3PCOOJ)** work prescribed for the said degree.

Name of the Lab-Incharge
Designation
Department of CSE
BMSCE, Bengaluru

Dr. Jyothi S Nayak
Professor and Head
Department of CSE
BMSCE, Bengaluru

`

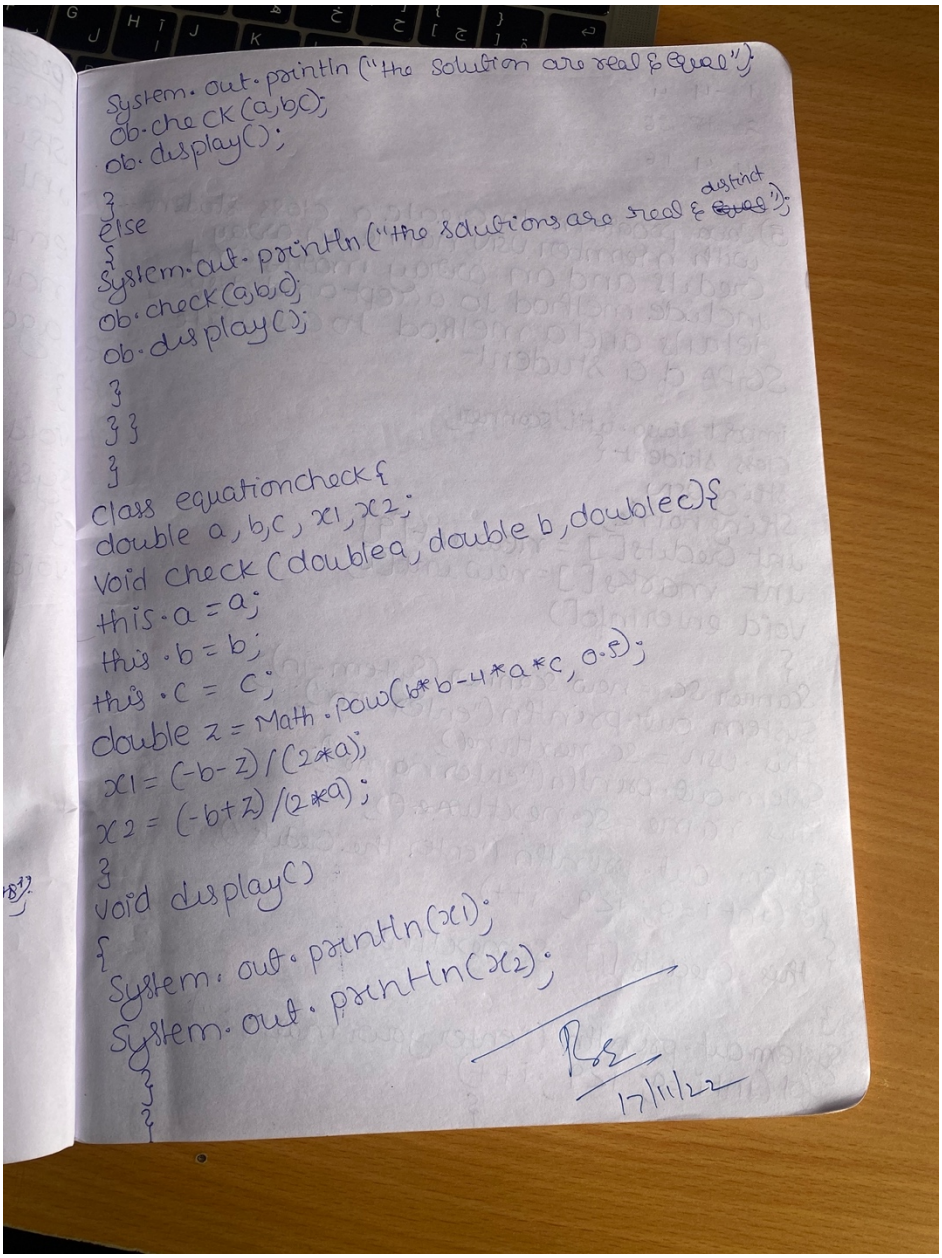
Index Sheet

Course Outcome

CO1	Apply the knowledge of java concepts to find the solution for a given solution
CO2	Analyze the given java application for correctness
CO3	Develop Java programs for a given requirement
CO4	Conduct practical experiments for demonstrating features of java

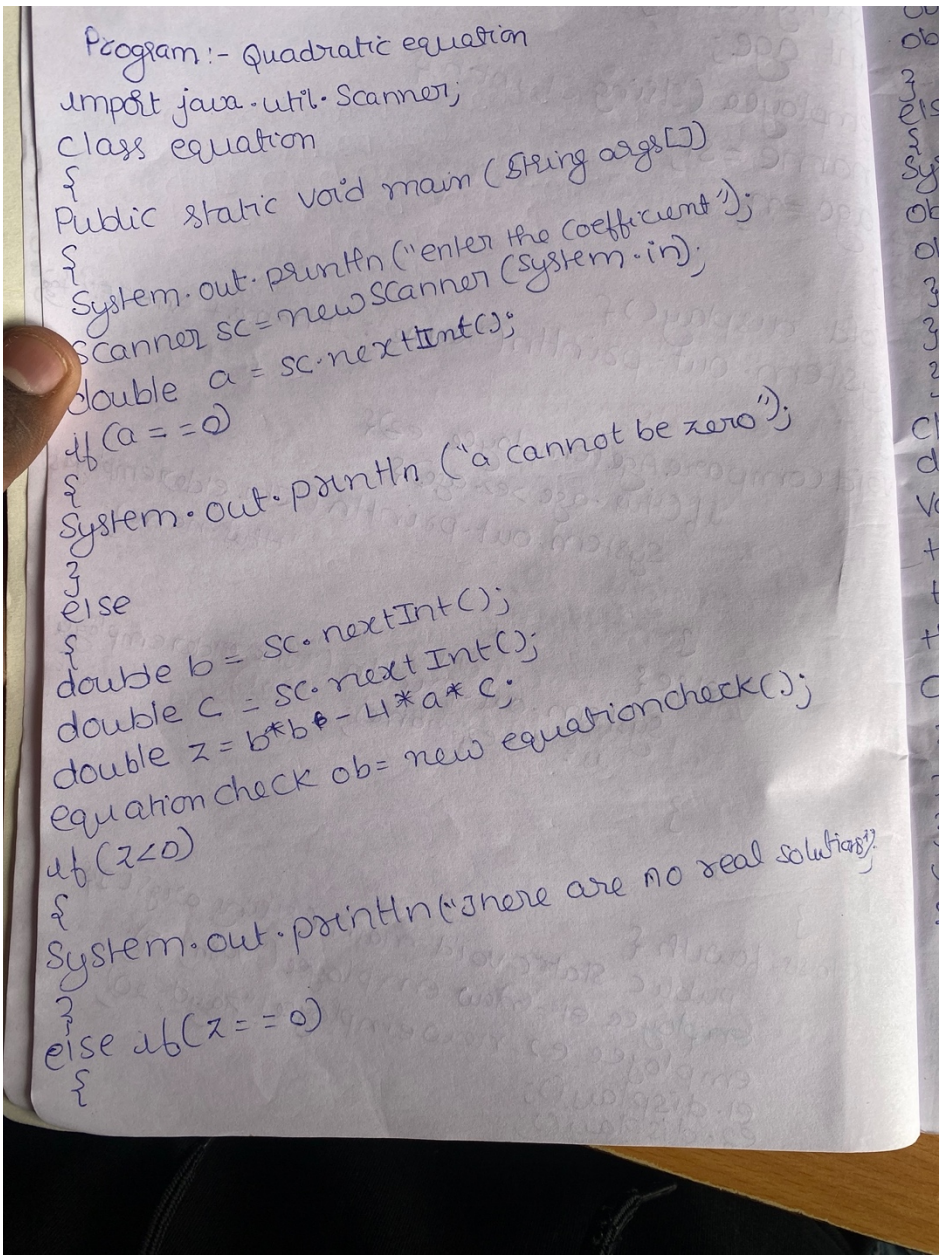
LAB PROGRAM 1:

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



```
System.out.println("the solution are real & Equal");
ob.checK(a,b,c);
ob.display();
}
else
{
System.out.println("the solutions are real & distinct");
ob.check(a,b,c);
ob.display();
}
}
}
class equationcheck{
double a,b,c,x1,x2;
void check(double a, double b, double c){
this.a=a;
this.b=b;
this.c=c;
double z=Math.pow(b*b-4*a*c,0.5);
x1=(-b-z)/(2*a);
x2=(-b+z)/(2*a);
}
void display()
{
System.out.println(x1);
System.out.println(x2);
}
}
```

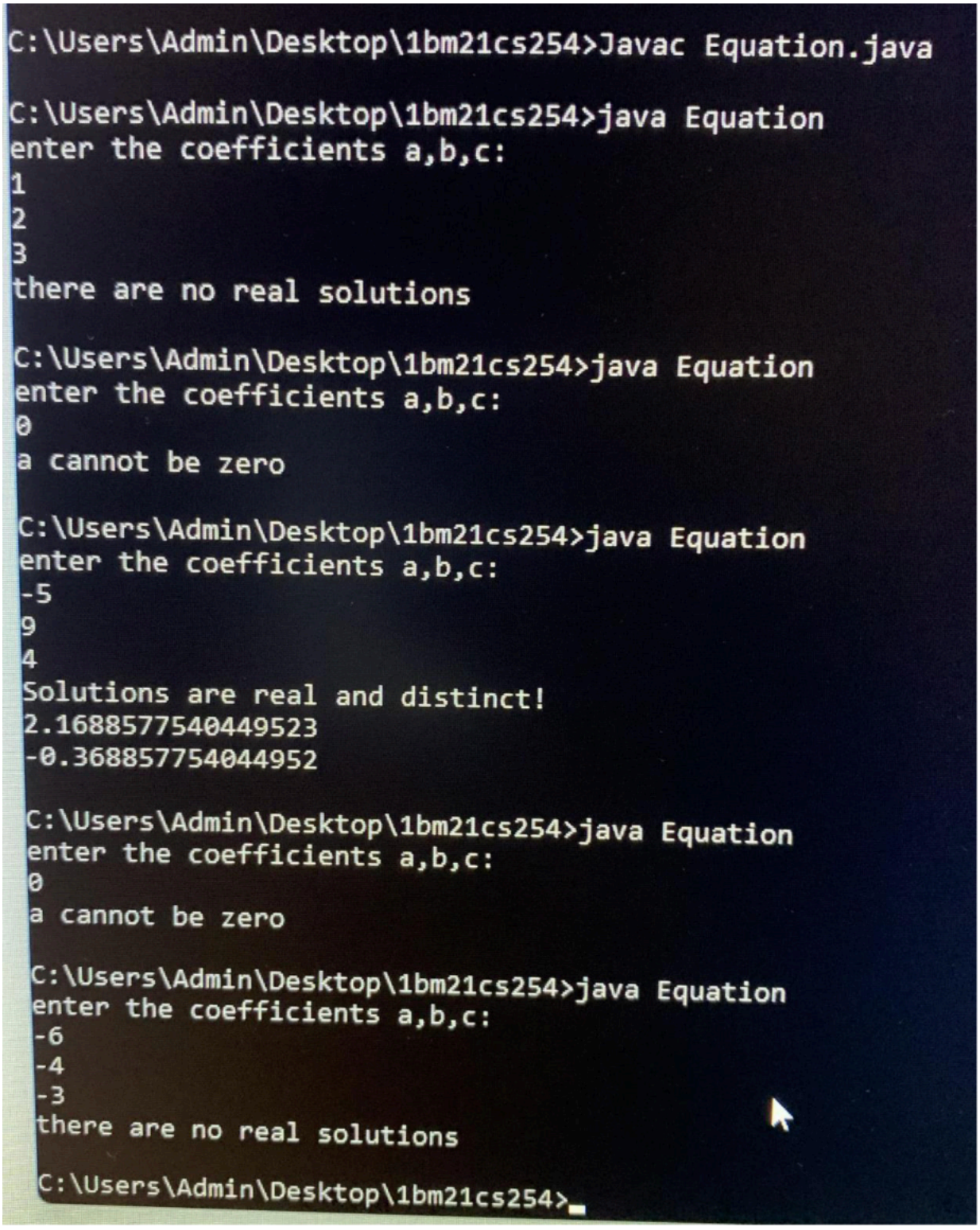
17/11/22



Program :- Quadratic equation

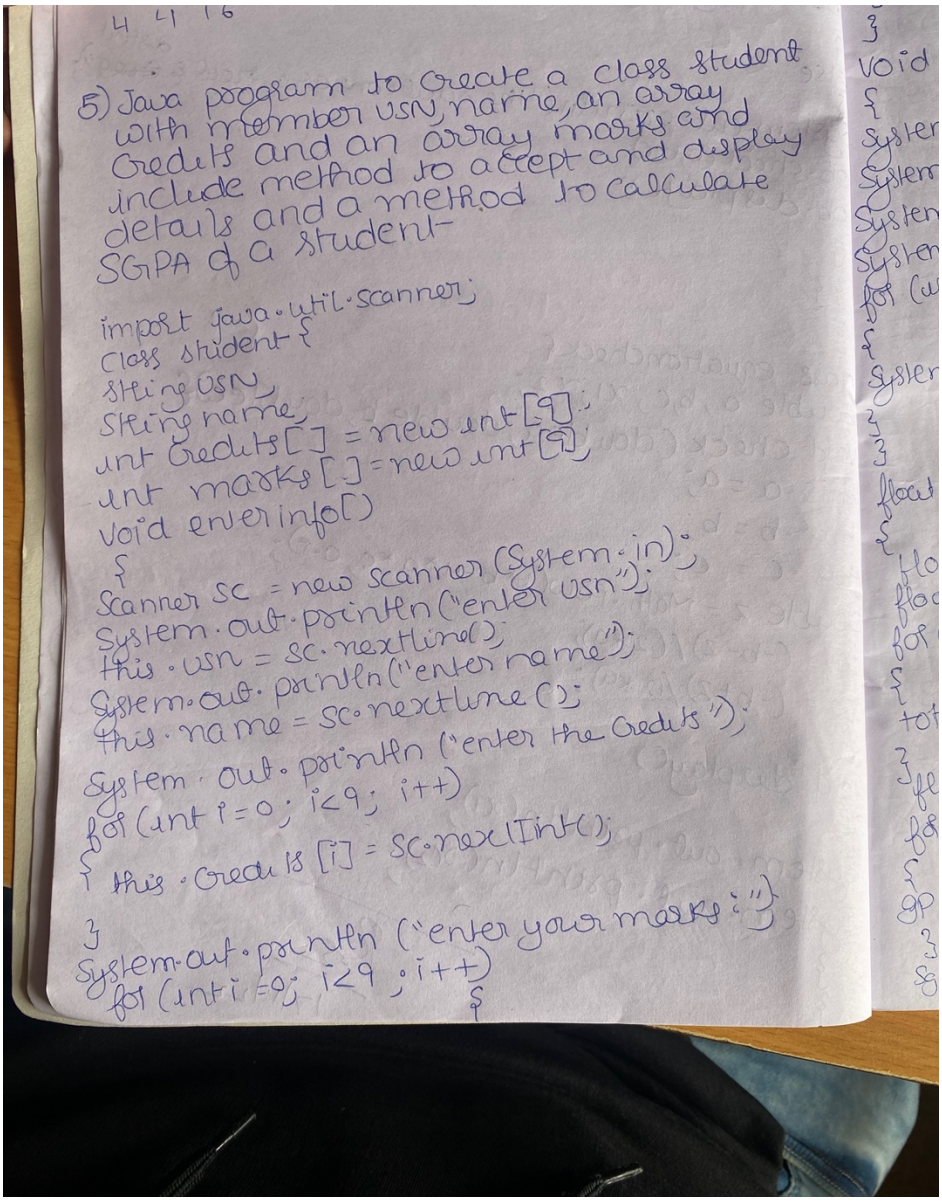
```
import java.util.Scanner;
class equation
{
Public static void main (String args[])
{
System.out.println("enter the coefficient");
Scanner sc = new Scanner (System.in);
double a = sc.nextInt();
if (a==0)
{
System.out.println ("a cannot be zero");
}
else
{
double b = sc.nextInt();
double c = sc.nextInt();
double z = b*b - 4*a*c;
equationcheck ob= new equationcheck();
if (z<0)
{
System.out.println("There are no real solutions");
}
else if(z==0)
{
```

Output:

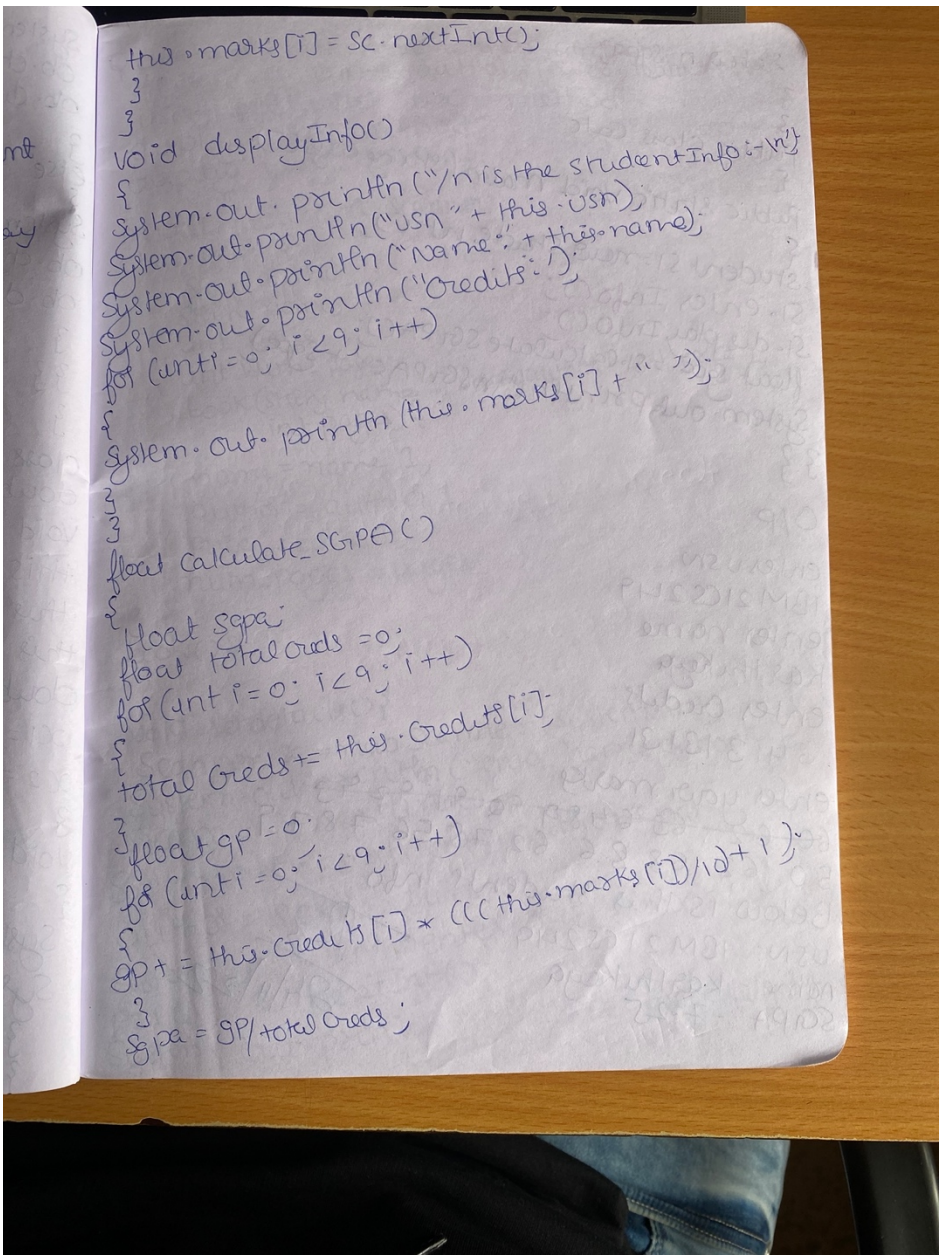


LAB PROGRAM 2:

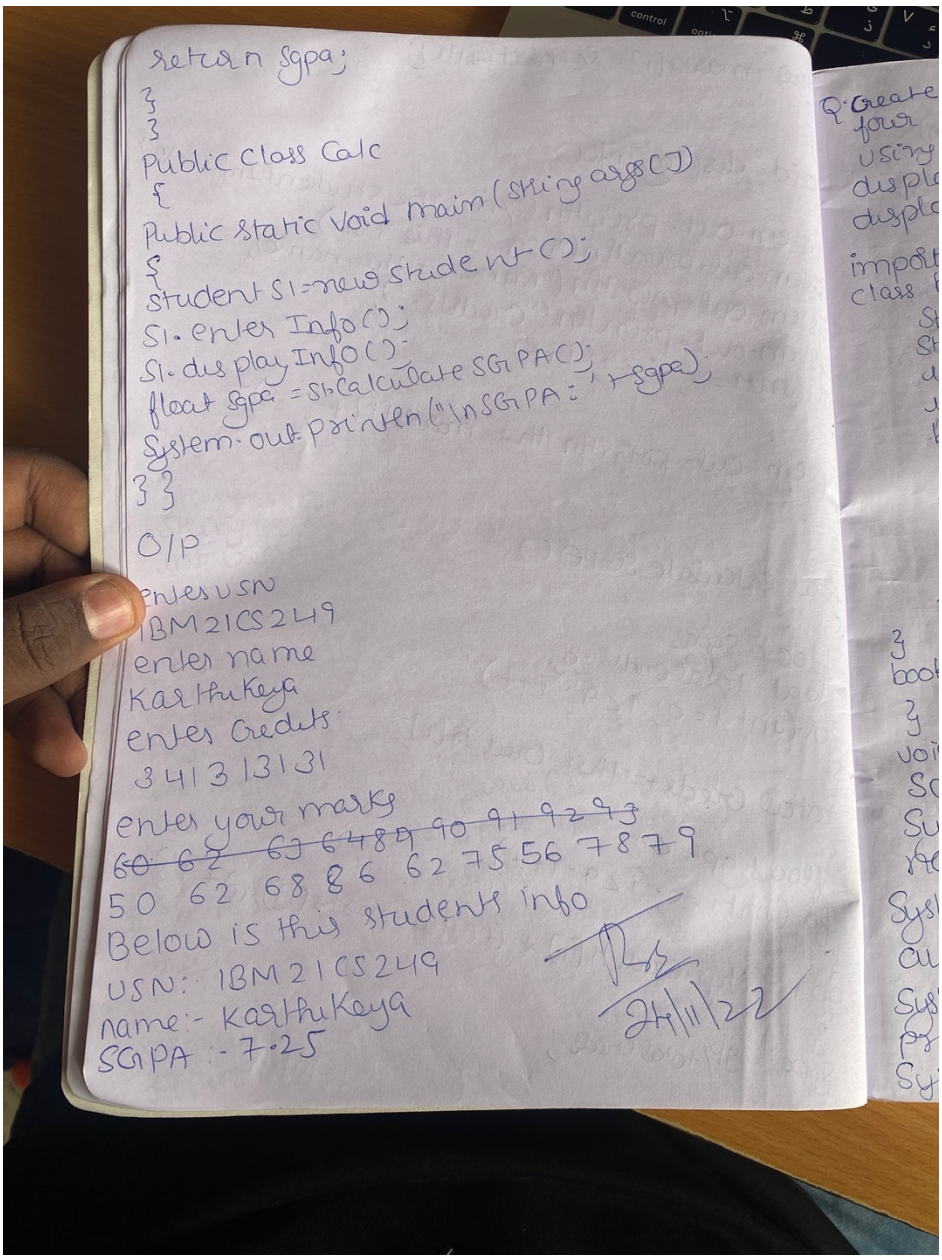
Q: Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.



```
5) Java program to create a class student
with member USN, name, an array
Credits and an array marks and
include method to accept and display
details and a method to calculate
SGPA of a student-

import java.util.scanner;
class student {
String USN;
String name;
int Credits[] = new int[9];
int marks[] = new int[9];
void enterinfo()
{
Scanner sc = new scanner(System.in);
System.out.println("enter usn");
this.usn = sc.nextline();
System.out.println("enter name");
this.name = sc.nextline();
System.out.println("enter the Credits");
for(int i=0; i<9; i++)
{
this.Credits[i] = sc.nextInt();
}
System.out.println("enter your marks:");
for(int i=0; i<9 ;i++)
{
```



```
        this.marks[i] = sc.nextInt();
    }
}
void displayInfo()
{
System.out.println("\n is the StudentInfo:-\n");
System.out.println("USN" + this.usn);
System.out.println("Name:" + this.name);
System.out.println("Credits:");
for(int i=0; i<9; i++)

System.out.println(this.marks[i] + " ");


float calculate_SGPA()

float sgpa;
float totalCreds =0;
for(int i=0; i<9; i++)

totalCreds += this.Credits[i];

float gp=0;
for(int i=0; i<9; i++)

gp += this.Credits[i] * (((this.marks[i])/10)+1);

sgpa = gp/totalCreds;
```

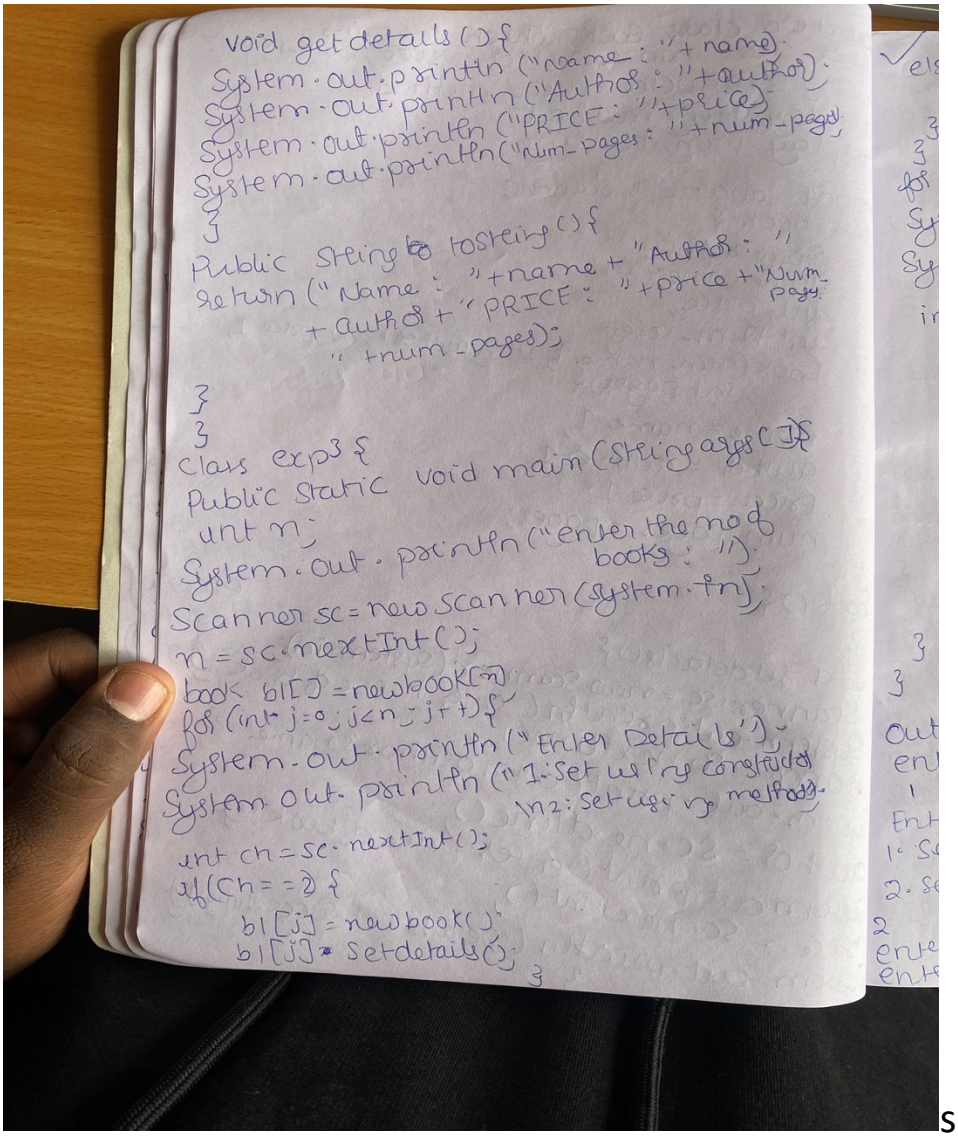

return Sgpa;
}
}
Public class Calc
{
Public static void main(String args[])
{
Student S1=new Student();
S1.enterInfo();
S1.displayInfo();
float Sgpa=S1.calculateSGPA();
System.out.println("\nSGPA:"+Sgpa);
}}
O/P
enter usn
1BM21CS249
enter name
Karthikeya
enter Credits
3 4 1 3 1 3 1 3 1
enter your marks
60 62 67 64 89 90 91 92 93
50 62 68 86 62 75 56 78 79
Below is this students info
USN:- 1BM21CS249
name:- Karthikeya
SGPA :- 7.25
24/11/22

Output:

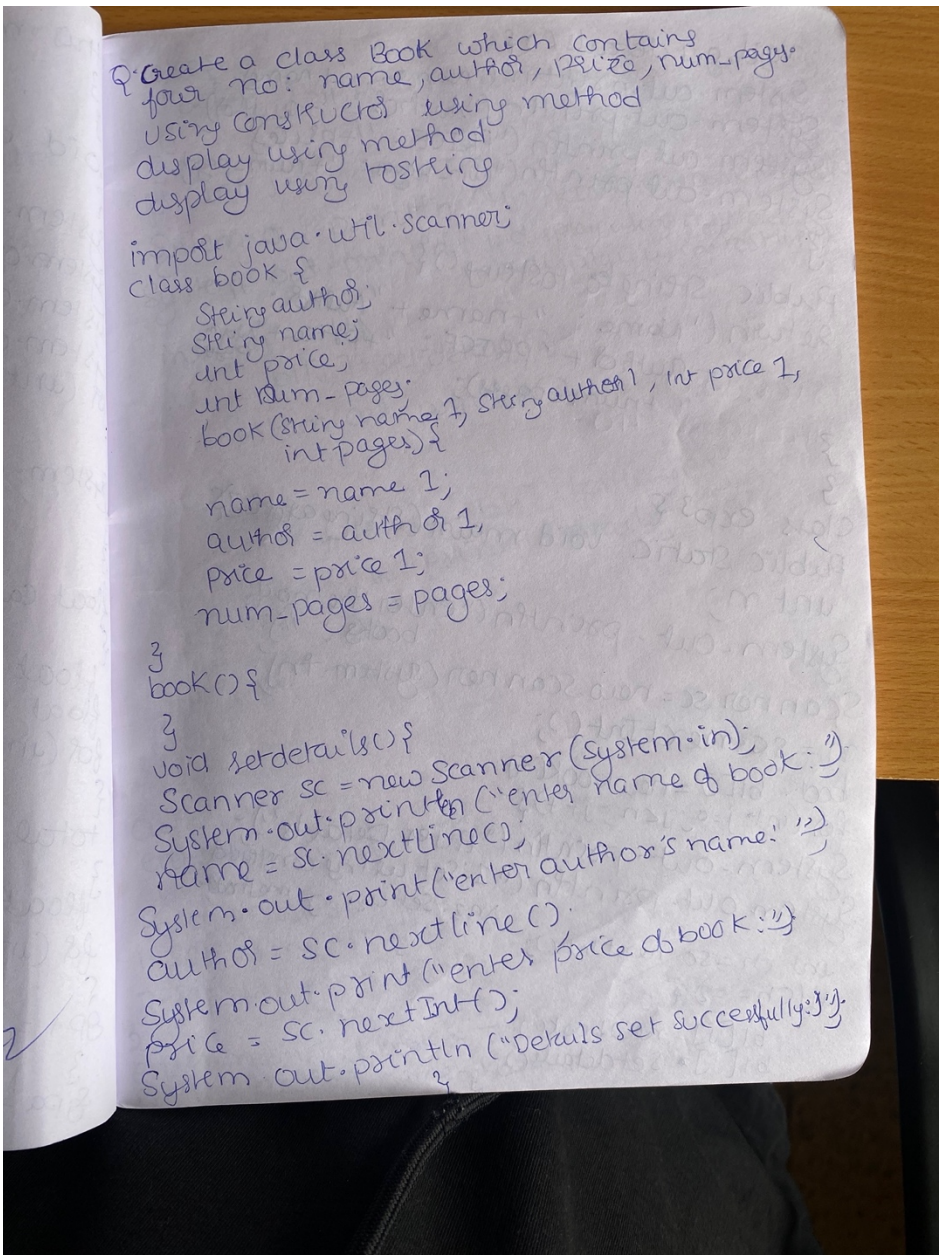
```
C:\Users\Admin\Desktop\1bm21cs254>java calc
enter the USN
1BM21CS254
Enter the Name:
L.Lawliet
Enter the credits:
3 4 1 3 1 3 1 3 1
enter your marks:
50 62 86 68 62 75 56 78 79

Below is the Student Information:-

USN: 1BM21CS254
NAME: L.Lawliet
CREDITS:
50
62
86
68
62
75
56
78
79

SGPA: 7.25
```

LAB PROGRAM 3:

Q: 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 object

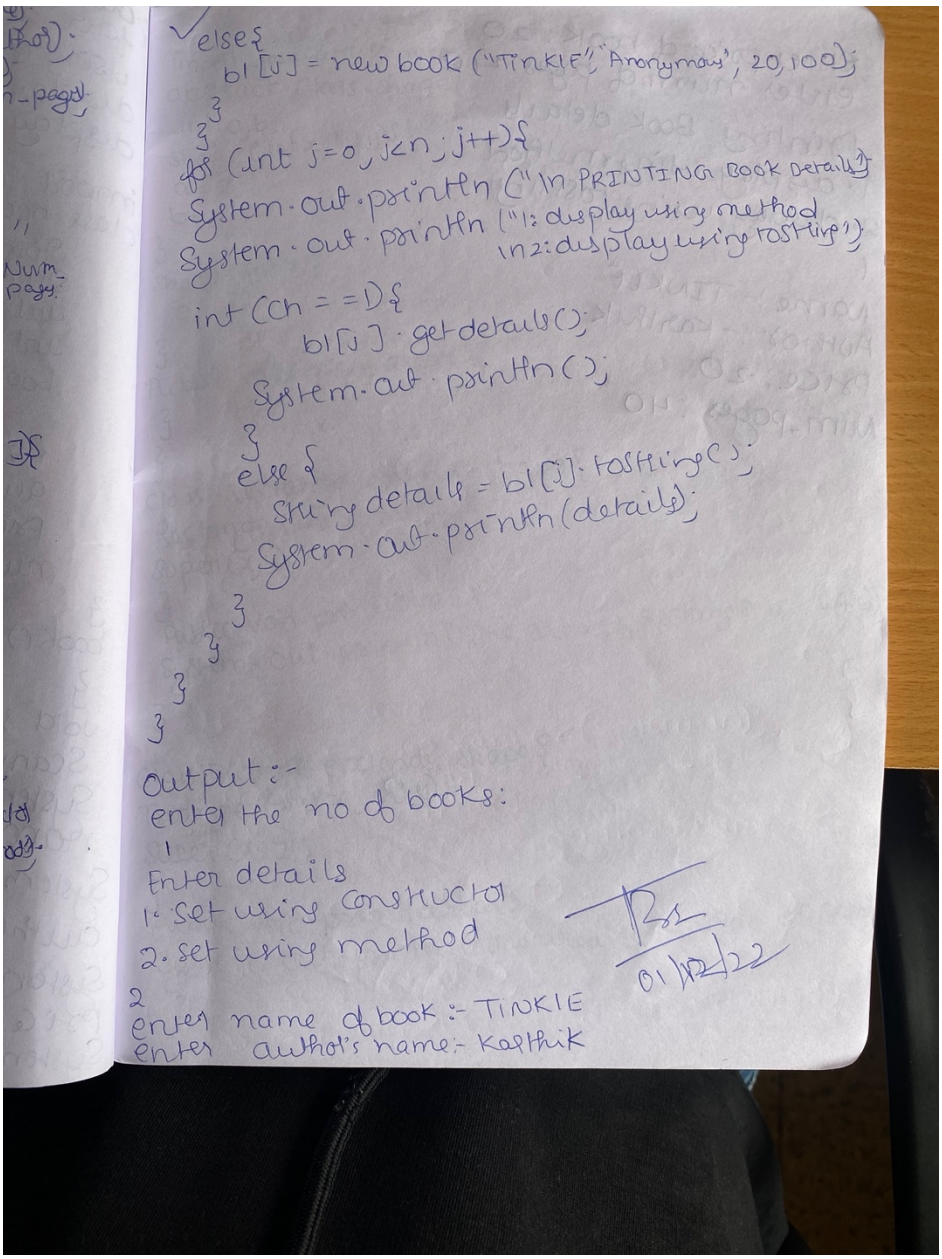


s.



Q. Create a class Book which contains four no: name, author, price, num_pages.
using constructor using method
display using method
display using toString

```
import java.util.Scanner;
class book {
    String author;
    String name;
    int price;
    int num_pages;
    book(String name1, String author1, int price1,
        int pages){

    name = name1;
    author = author1;
    price = price1;
    num_pages = pages;

}
book(){
}
void setdetails(){
Scanner sc = new Scanner(System.in);
System.out.println("enter name of book:");
name = sc.nextLine();
System.out.print("enter author's name:");
author = sc.nextLine();
System.out.print("enter price of book:");
price = sc.nextInt();
System.out.println("Details set successfully:");
}
```



```
else{
    bl[i] = new book ("TINKIE", "Anonymous", 20, 100);
  }
}
for (int j=0; j<n; j++){
System.out.println ("In PRINTING Book Details")
System.out.println ("1: display using method
                     in2: display using toString");
int ch = =1){
        bl[j].getdetails();
    System.out.println();
  }
  else {
    String details = bl[j].toString();
    System.out.println(details);
  }
 }
}
}
```

Output :-

enter the no of books:

1

Enter details

1. Set using Constructor

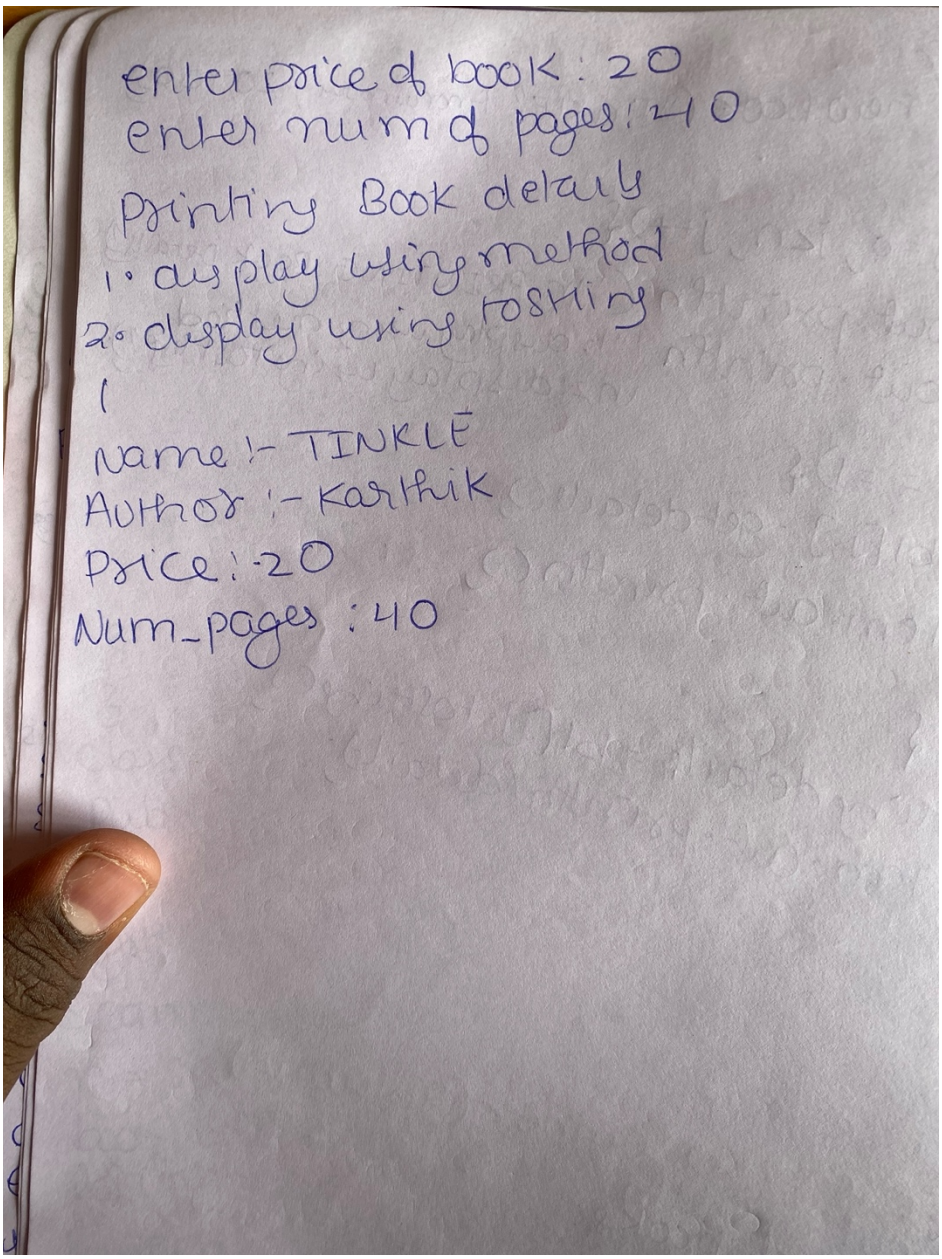
2. set using method

2

enter name of book :- TINKIE

enter author's name:- Kasthik

01/12/22

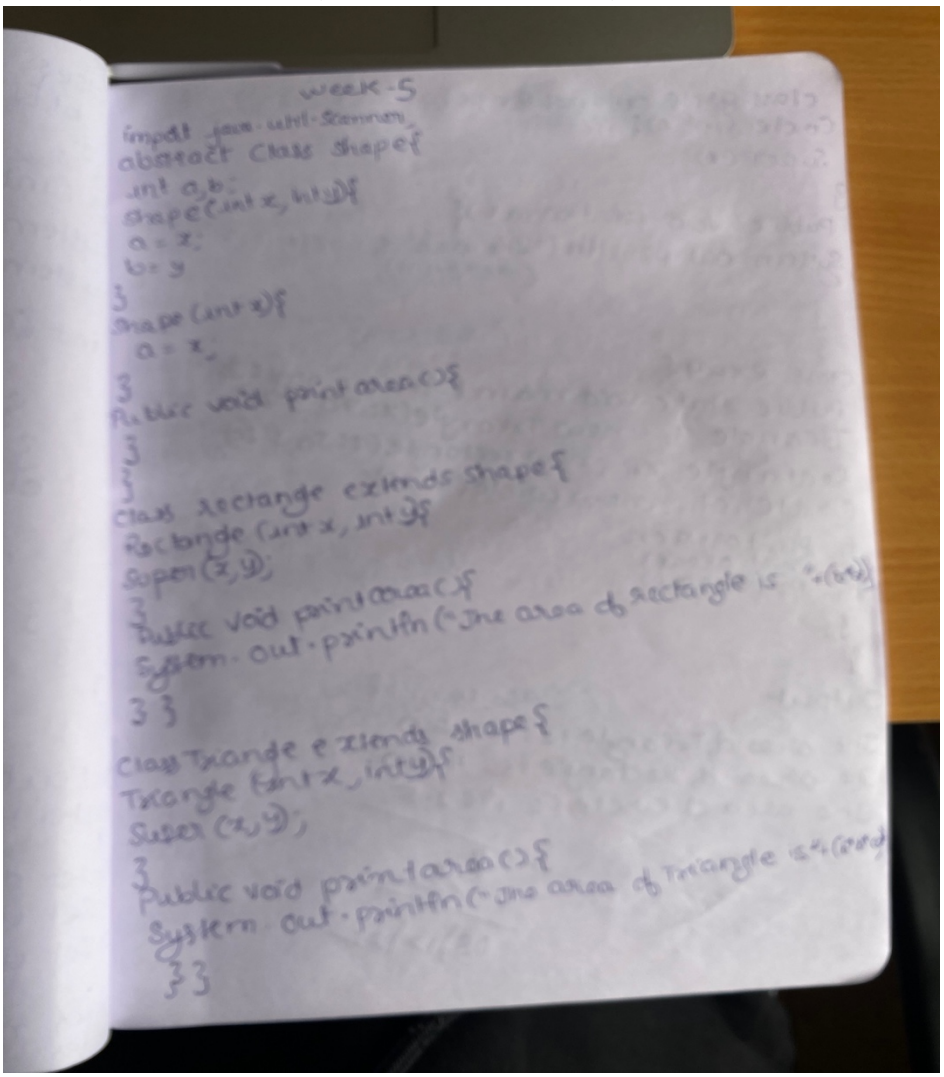

enter price of book : 20
enter num of pages : 40
Printing Book details
1. display using method
2. display using toString
1
Name :- TINKLE
Author :- Karthik
Price : 20
Num_pages : 40

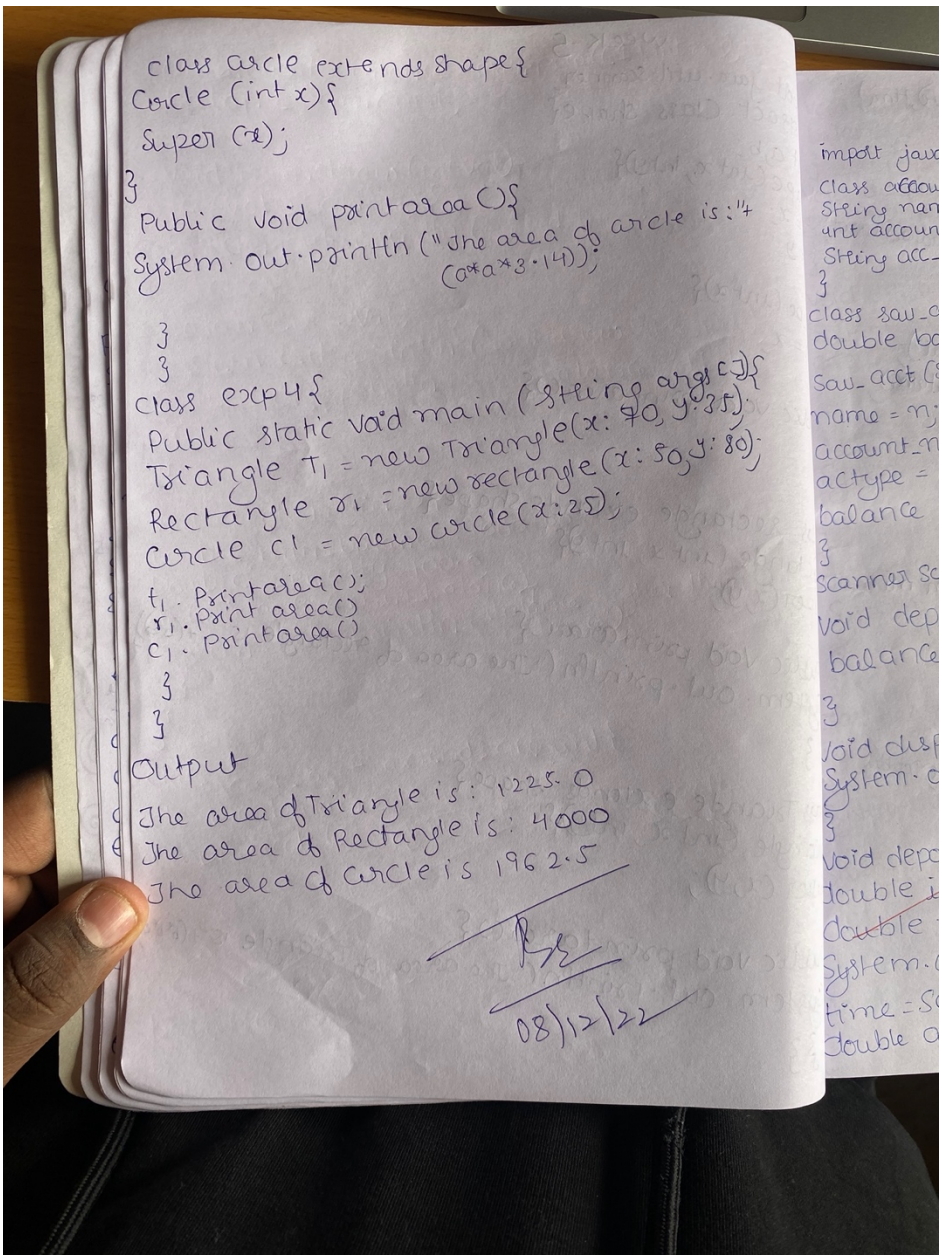
Output:

```
enter the number of books:
2
ENTER DETAILS
1:set using constructor
2:set using method
2
enter name of book: lmao
enter author's name: rfgg
enter price of book: 22
enter num of pages: 22
DETAILS SET SUCCESSFULLY :)
ENTER DETAILS
1:set using constructor
2:set using method
1

PRINTING BOOK DETAILS
1:display using method
2:display using toString
1
NAME: lmao
AUTHOR: rfgg
PRICE: 22
NUM_PAGES: 22


PRINTING BOOK DETAILS
1:display using method
2:display using toString
2
NAME: TINKLE AUTHOR: ANONYMOUS PRICE: 20 NUM_PAGES: 100
```

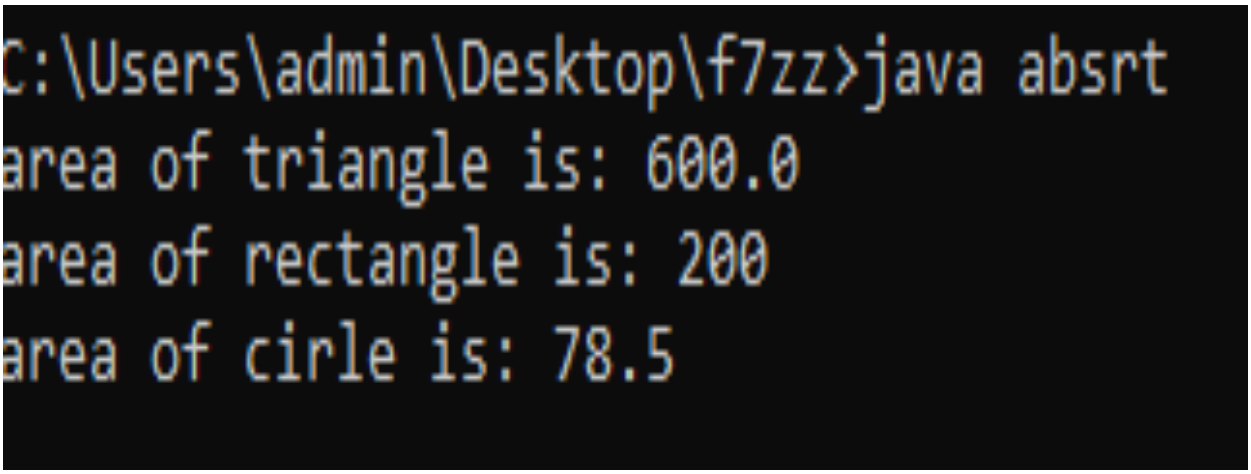
LAB PROGRAM 4:

Q: 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.




class circle extends shape{
Circle (int x){
Super (x);
}
Public void printarea (){
System.out.println ("The area of circle is:"+
(a*a*3.14));
}
}
class exp4{
Public static void main (String args[]){
Triangle T1 = new Triangle(x:70, y:35);
Rectangle r1 = new rectangle(x:50, y:80);
Circle c1 = new circle(x:25);
t1.Printarea();
r1.Print area()
c1.Printarea()
}
}
Output
The area of Triangle is: 1225.0
The area of Rectangle is: 4000
The area of circle is 1962.5
08/12/22

Output:



LAB PROGRAM 5:

Q: 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.

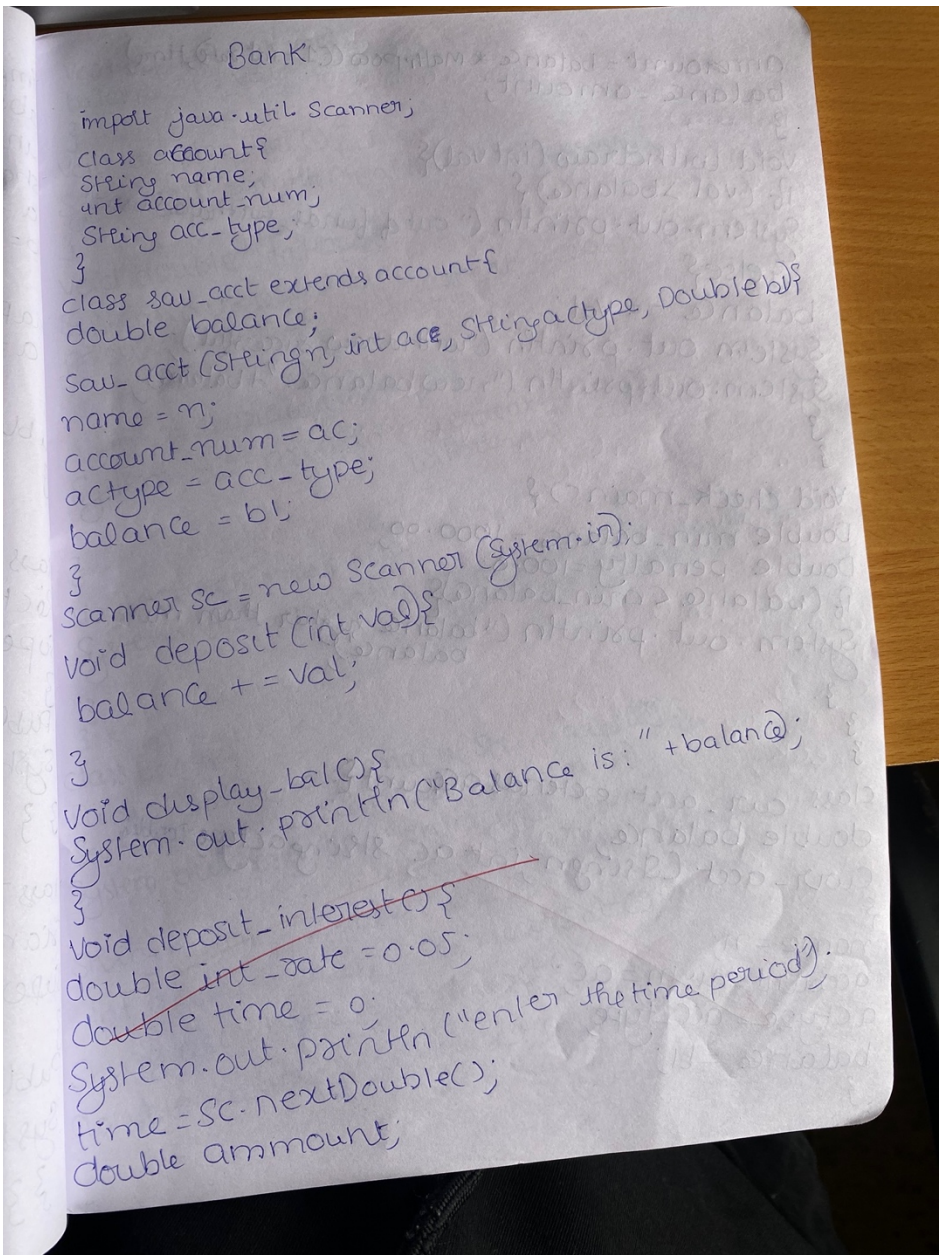
Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

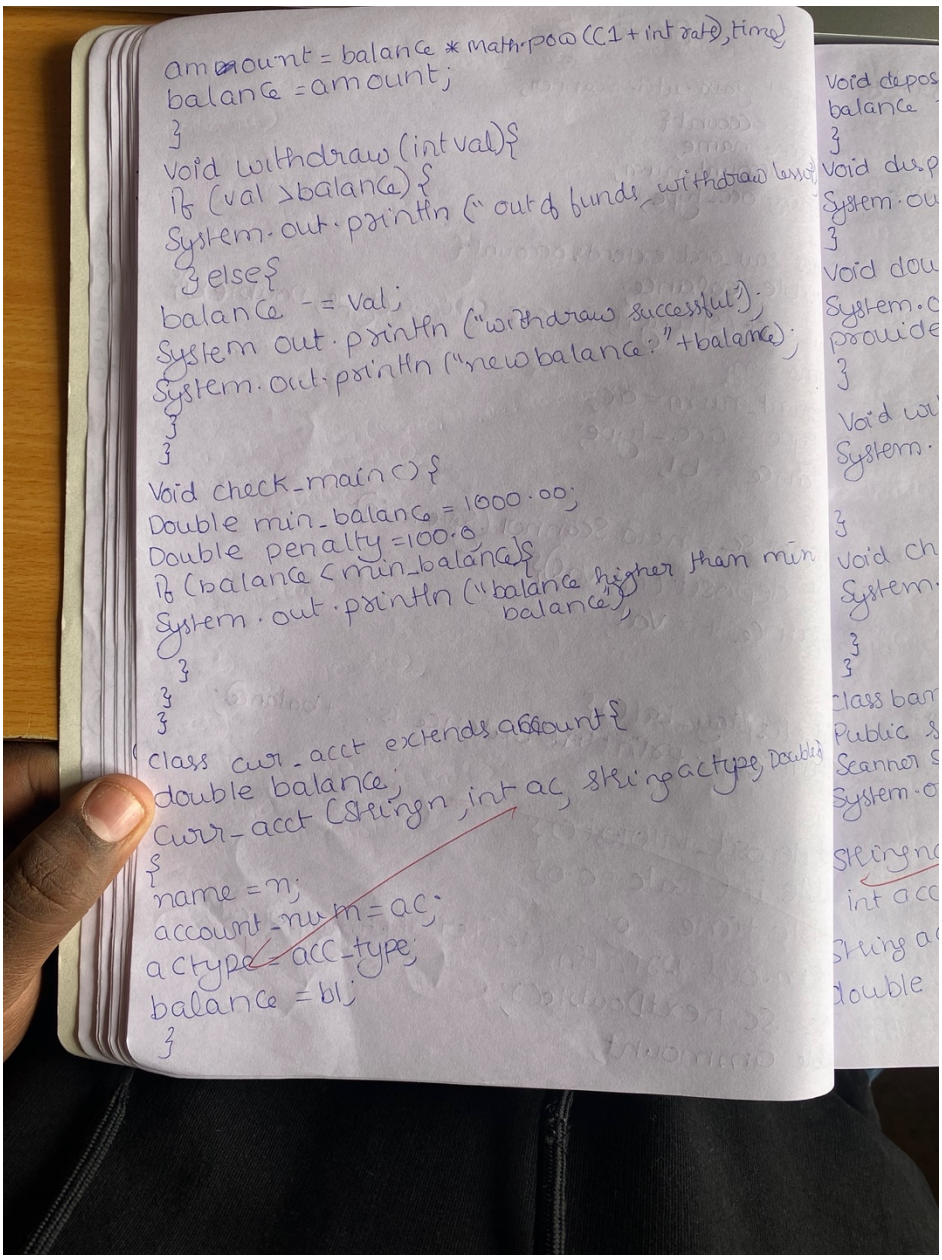
c) Compute and deposit interest

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

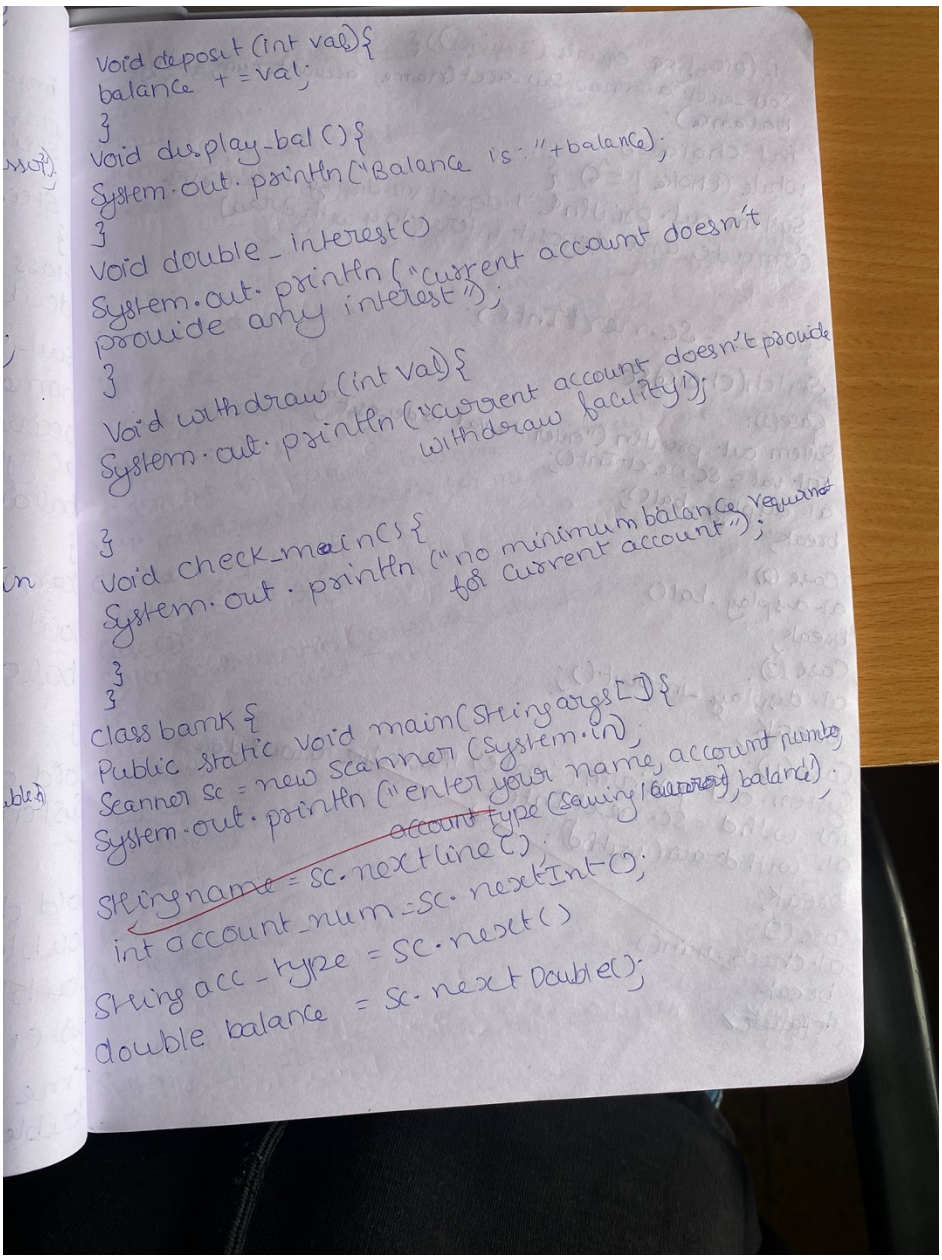


Bank

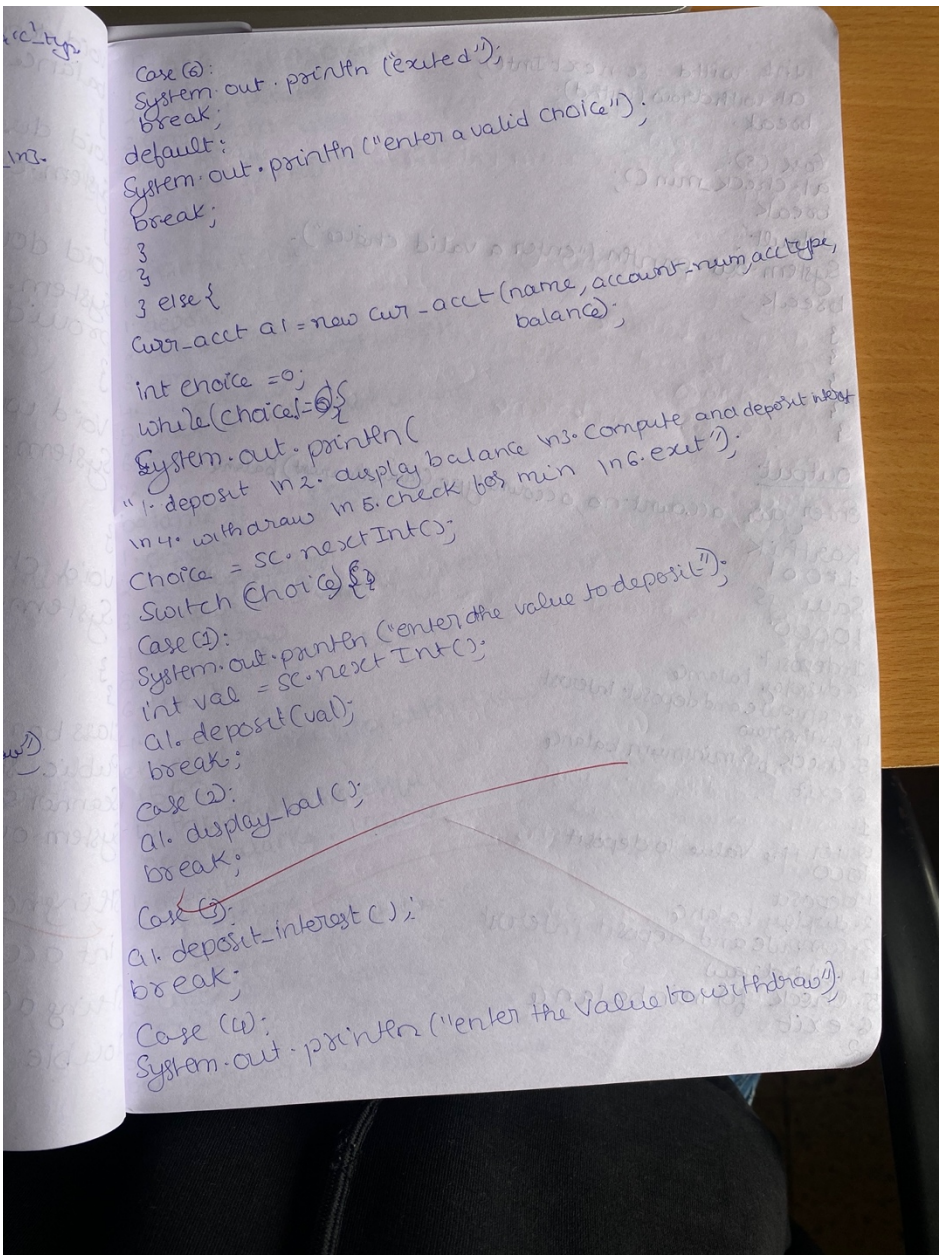
```
import java.util.Scanner;
class account{
String name;
int account_num;
String acc_type;
}
class sav_acct extends account{
double balance;
sav_acct(String n, int ac, String actype, Double bl){
name = n;
account_num = ac;
actype = acc_type;
balance = bl;
}
Scanner sc = new Scanner(System.in);
void deposit(int val){
balance += val;
}
void display_bal(){
System.out.println("Balance is: " +balance);
}
void deposit_interest(){
double int_rate = 0.05;
double time = 0;
System.out.println("enter the time period");
time = sc.nextDouble();
double ammount;
```



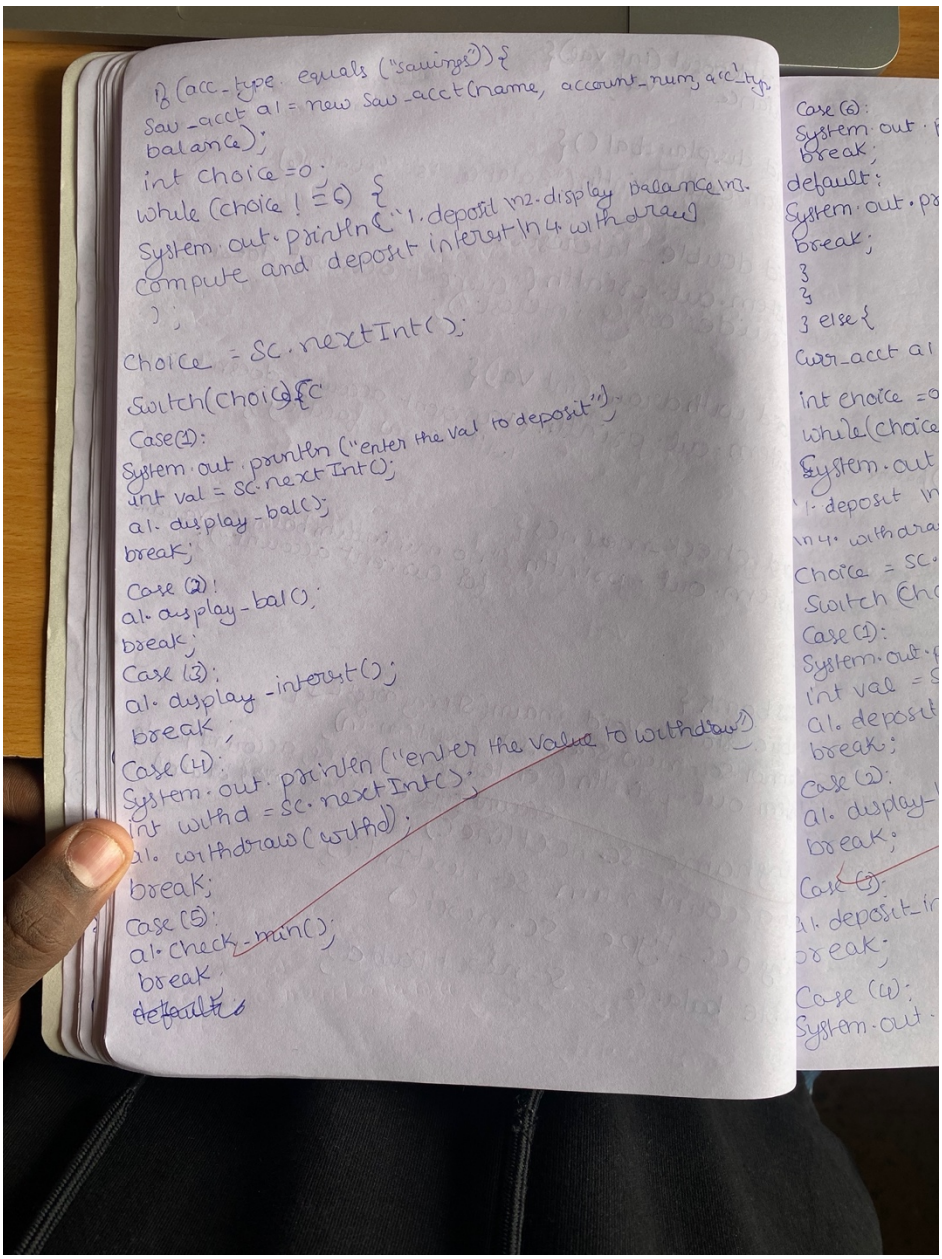
```
amount = balance * Math.pow((1 + int rate), time)
balance = amount;
}
Void withdraw (int val){
if (val > balance){
System.out.println (" out of funds, withdraw lesser");
}else{
balance -= val;
System out.println ("withdraw successful");
System.out.println ("new balance:" + balance);
}
}

Void check_main(){
Double min_balance = 1000.00;
Double penalty = 100.0;
If (balance < min_balance){
System.out.println ("balance higher than min
                     balance");
}
}
}

class cur_acct extends account{
double balance;
Curr_acct (String n, int ac, String actype, Double
{
name = n;
account_num = ac;
actype = acc_type;
balance = bl;
}
```

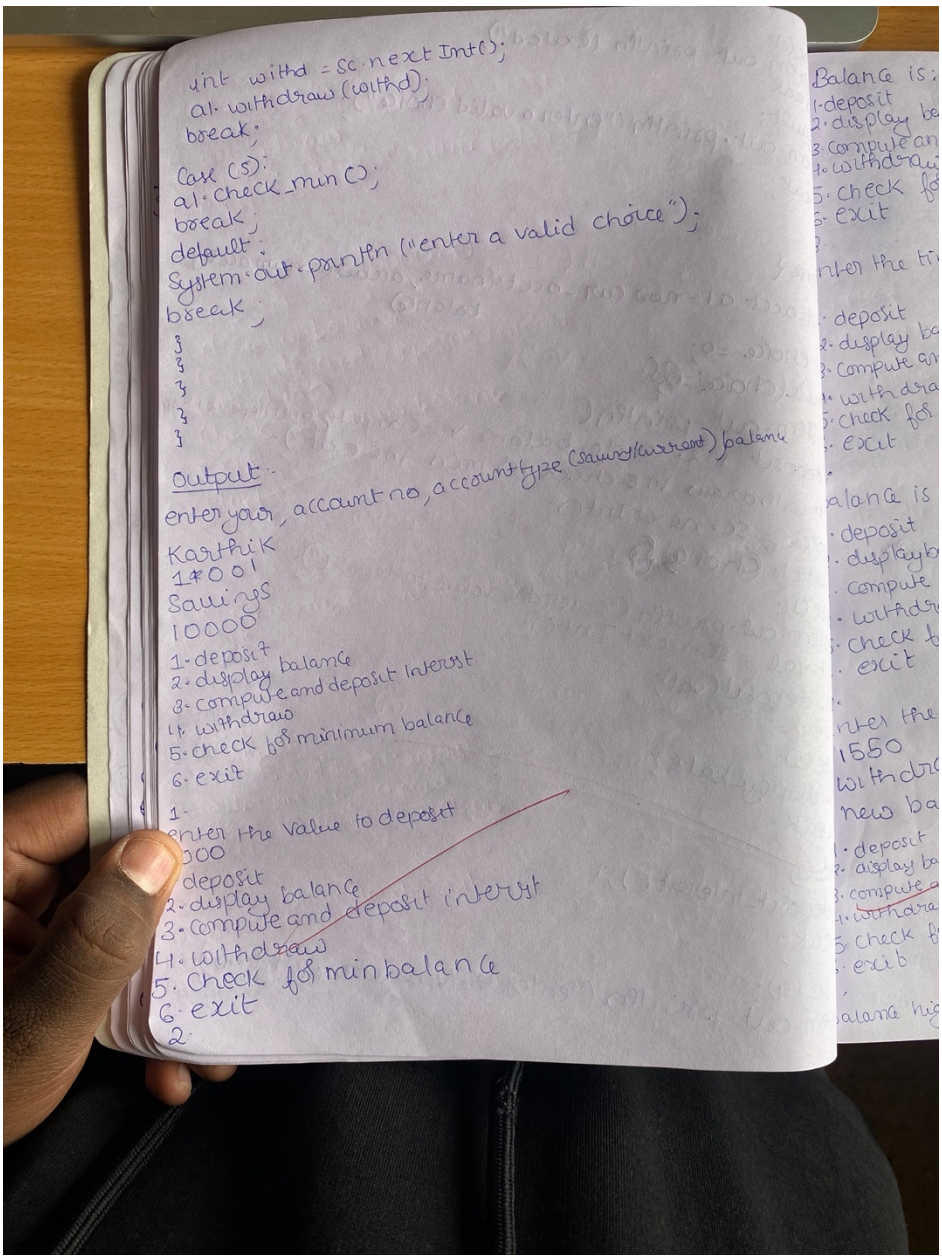


```
Void deposit (int val){
balance += val;
}
Void display_bal(){
System.out.println("Balance is: "+balance);
}
Void double_interest()
System.out.println("current account doesn't
provide any interest");
}
Void withdraw (int val){
System.out.println("current account doesn't provide
                    withdraw facility");
}
Void check_main(){
System.out.println("no minimum balance required
                    for current account");
}
}
Class bank{
Public static void main(String args[]){
Scanner sc = new Scanner (System.in);
System.out.println("enter your name, account number
                    account type (Saving/Current), balance");
String name = sc.nextline();
int account_num = sc.nextInt();
String acc_type = sc.next()
double balance = sc.nextDouble();
```



```
Case (6):
System.out.println ("exited");
break;
default:
System.out.println ("enter a valid choice");
break;
}
}
} else {
Cur_acct a1 = new cur-acct (name, account_num, acc type,
                            balance);
int choice = 0;
while (choice != 6) {
System.out.println (
"1. deposit in 2. display balance in 3. Compute and deposit interest
in 4. withdraw in 5. check bal, min in 6. exit");
choice = sc.nextInt();
switch (choice) {
Case (1):
System.out.println ("enter the value to deposit");
int val = sc.nextInt();
a1.deposit(val);
break;
Case (2):
a1.display_bal();
break;
Case (3):
a1.deposit_interest();
break;
Case (4):
System.out.println ("enter the value to withdraw");
```

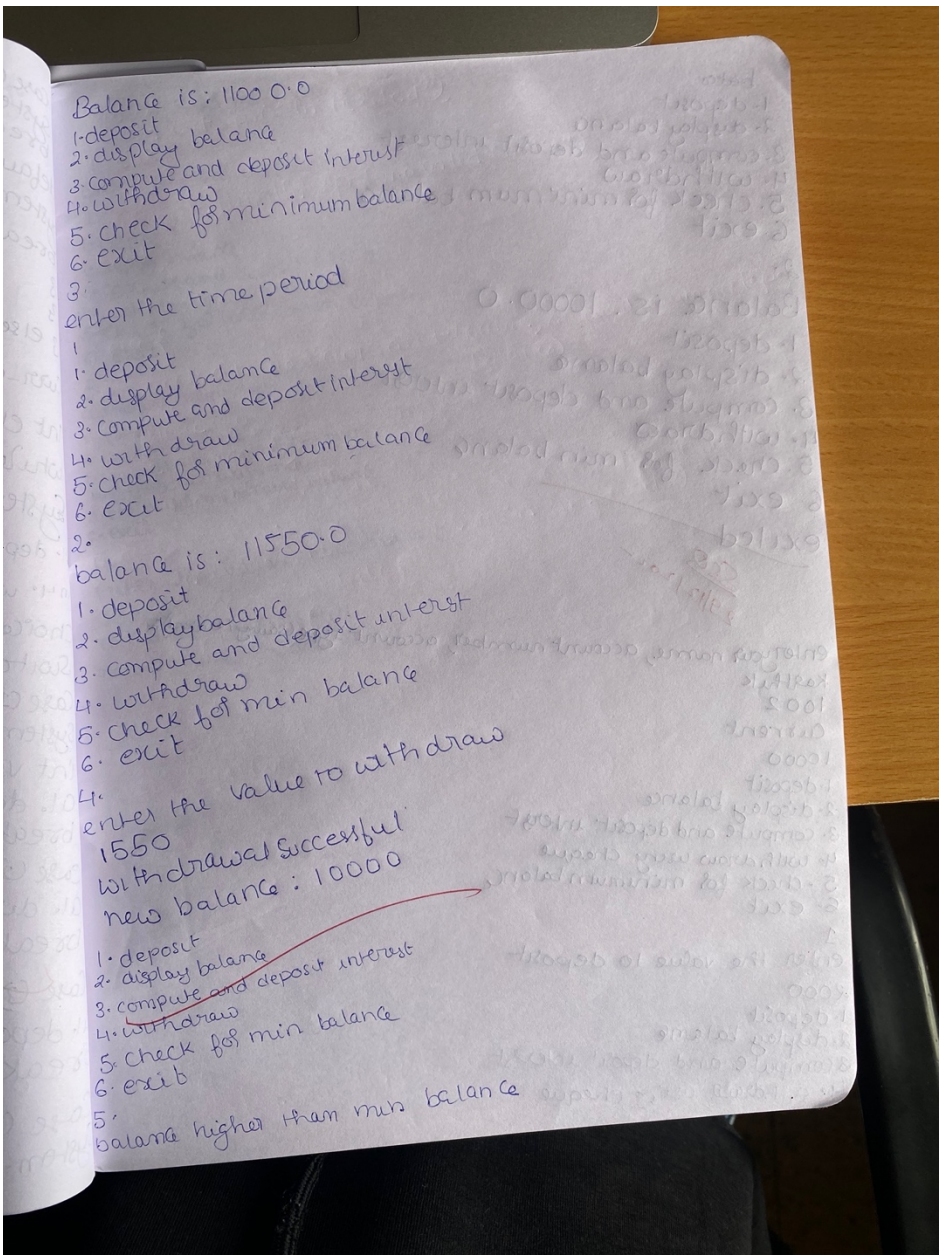

if(acc_type.equals("savings")){
Sav_acct a1 = new Sav_acct(name, account_num, acc_typ
balance);
int choice=0;
while(choice != 6) {
System.out.println(" 1.deposit 2.display balance 3.
compute and deposit interest 4 withdraw
");
choice = sc.nextInt();
switch(choice){
case(1):
System.out.println("enter the val to deposit");
int val = sc.nextInt();
a1.display_bal();
break;
case(2):
a1.display_bal();
break;
case(3):
a1.display_interest();
break;
case(4):
System.out.println("enter the value to withdraw");
int withd = sc.nextInt();
a1.withdraw(withd);
break;
case(5):
a1.check_tran();
break;
default:
case(6):
default:
} else{
int choice =
while(choice
Choice = sc.
Switch (ch
Case(1):
int val =
a1.deposit
break;
case(2):
a1.display_
break;
a1.deposit_in
break;
case(4):

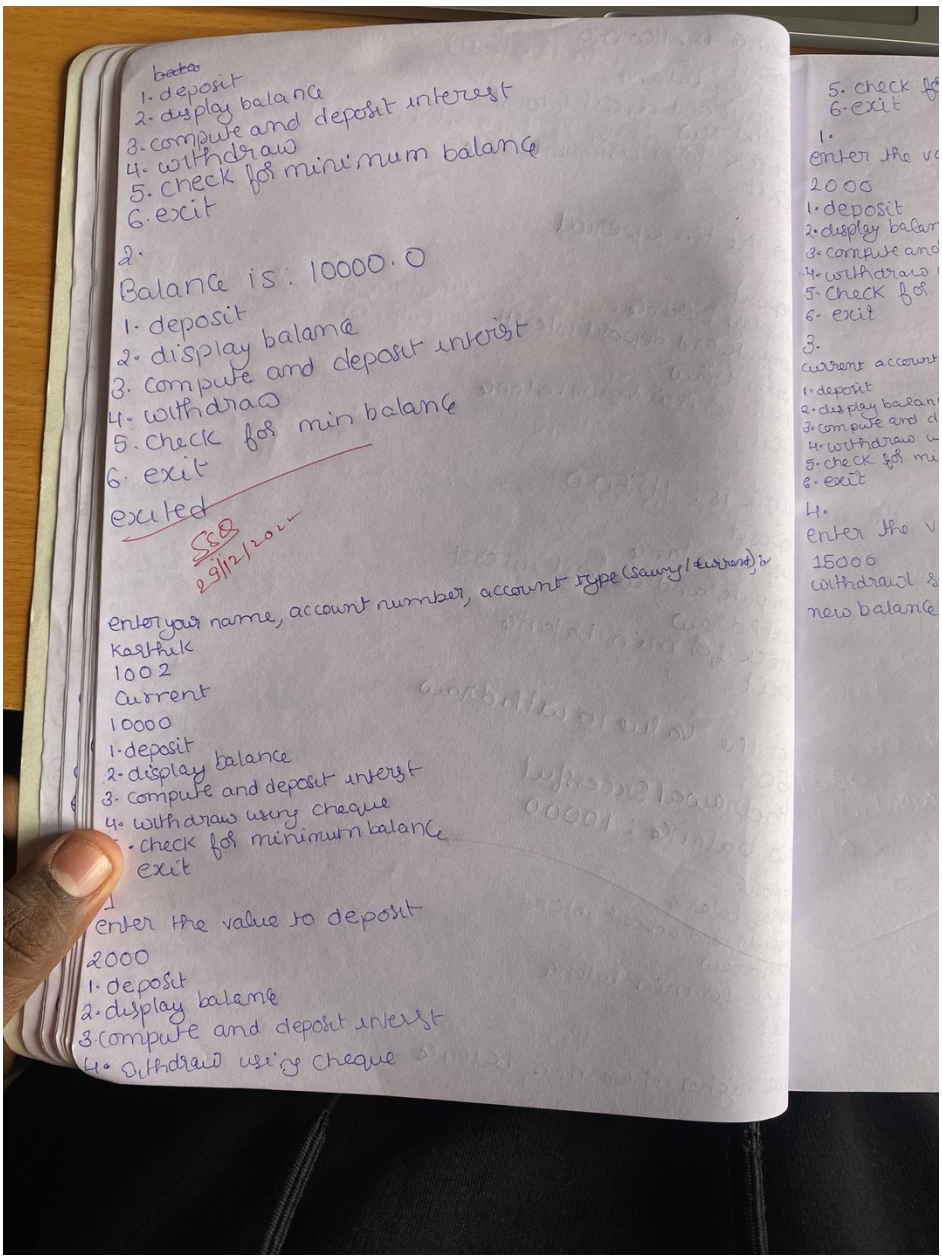


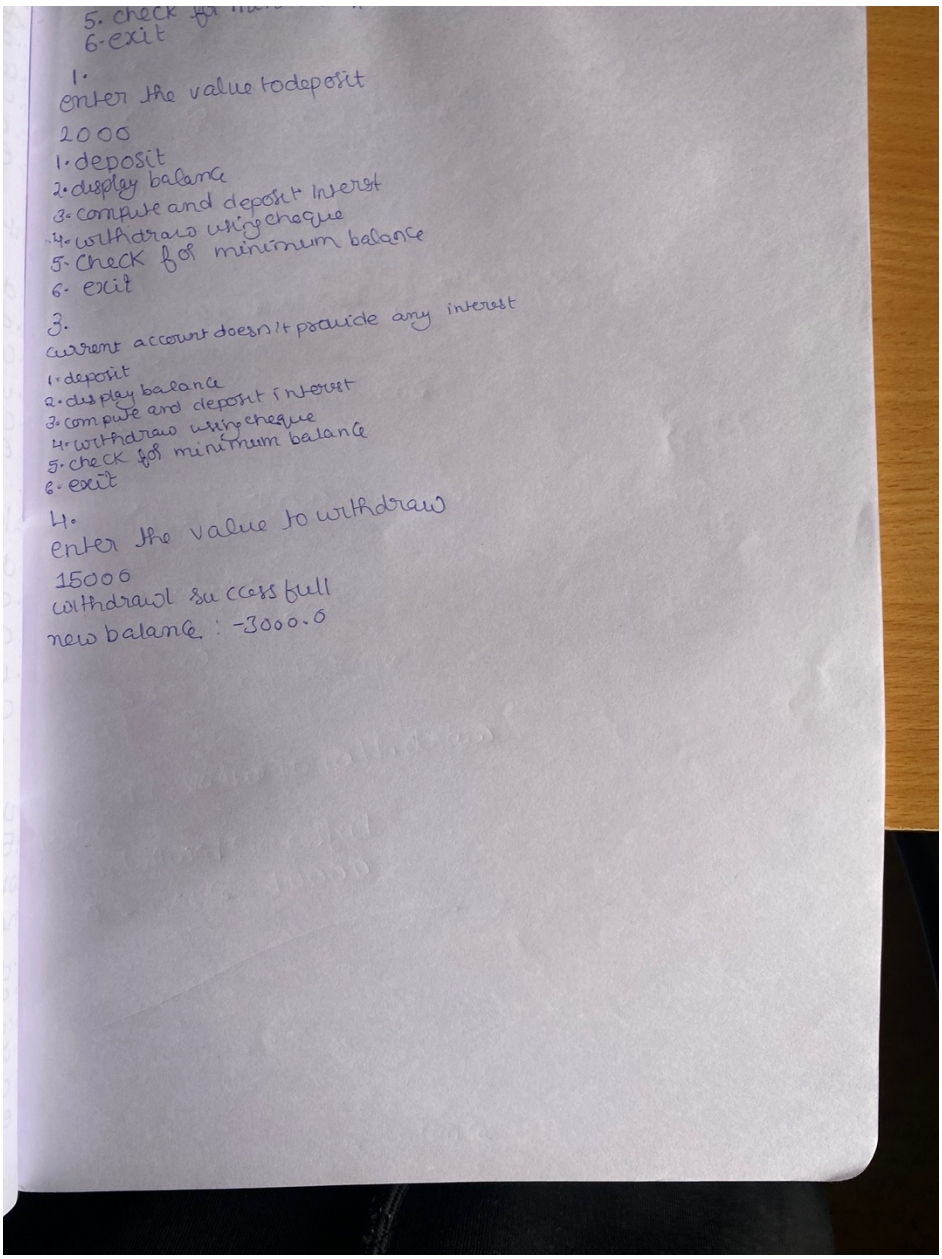
```
int withd = sc.nextInt();
a1.withdraw(withd);
break;
Case (5):
a1.check_min();
break;
default:
System.out.println("enter a valid choice");
break;
}
}
}
}
}
```

Output:

```
enter your, account no, account type (saving/current) balance
Karthik
1#001
Savings
10000
1. deposit
2. display balance
3. compute and deposit interest
4. withdraw
5. check for minimum balance
6. exit
1.
enter the value to deposit
000
deposit
2. display balance
3. compute and deposit interest
4. withdraw
5. Check for minbalance
6. exit
2.
```


Balance is: 1100.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
3.
enter the time period
1
1. deposit
2. display balance
3. Compute and deposit interest
4. withdraw
5. check for minimum balance
6. exit
2.
balance is: 11550.0
1. deposit
2. displaybalance
3. compute and deposit interest
4. withdraw
5. check for min balance
6. exit
4.
enter the value to withdraw
1550
withdrawal successful
new balance: 10000
1. deposit
2. display balance
3. compute and deposit interest
4. withdraw
5. check for min balance
6. exit
5.
balance higher than min balance


1. deposit
2. display balance
3. compute and deposit interest
4. withdraw
5. check for minimum balance
6. exit
2.
Balance is: 10000.0
1. deposit
2. display balance
3. compute and deposit interest
4. withdraw
5. check for min balance
6. exit
exited
SSB
29/12/2021
enter your name, account number, account type (saving / current) in
Kasthuk
1002
current
10000
1. deposit
2. display balance
3. compute and deposit interst
4. withdraw using cheque
5. check for minimum balance
exit
1
enter the value to deposit
2000
1. deposit
2. display balance
3. compute and deposit interst
4. withdraw using cheque
5. check
6. exit
1.
2000
1. deposit
3.
1. deposit
4.
enter the
15000
new balance



Output for savings:

```
enter your name, account number, aacount type(savings/current), balance
zayd
10000001
savings
10000
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
1
enter the value to deposit
1000
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
2
Balance is: 11000.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
3
enter the time period
2
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
2
Balance is: 12127.5
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
4
enter the value to withdraw
1000
withdrawal successful
new balance: 11127.5
1.deposit
```

```
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
4
enter the value to withdraw
1000
withdrawal successful
new balance: 11127.5
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
2
Balance is: 11127.5
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
5
balance higher than minimum balance
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw
5.check for minimum balance
6.exit
6
exited
```

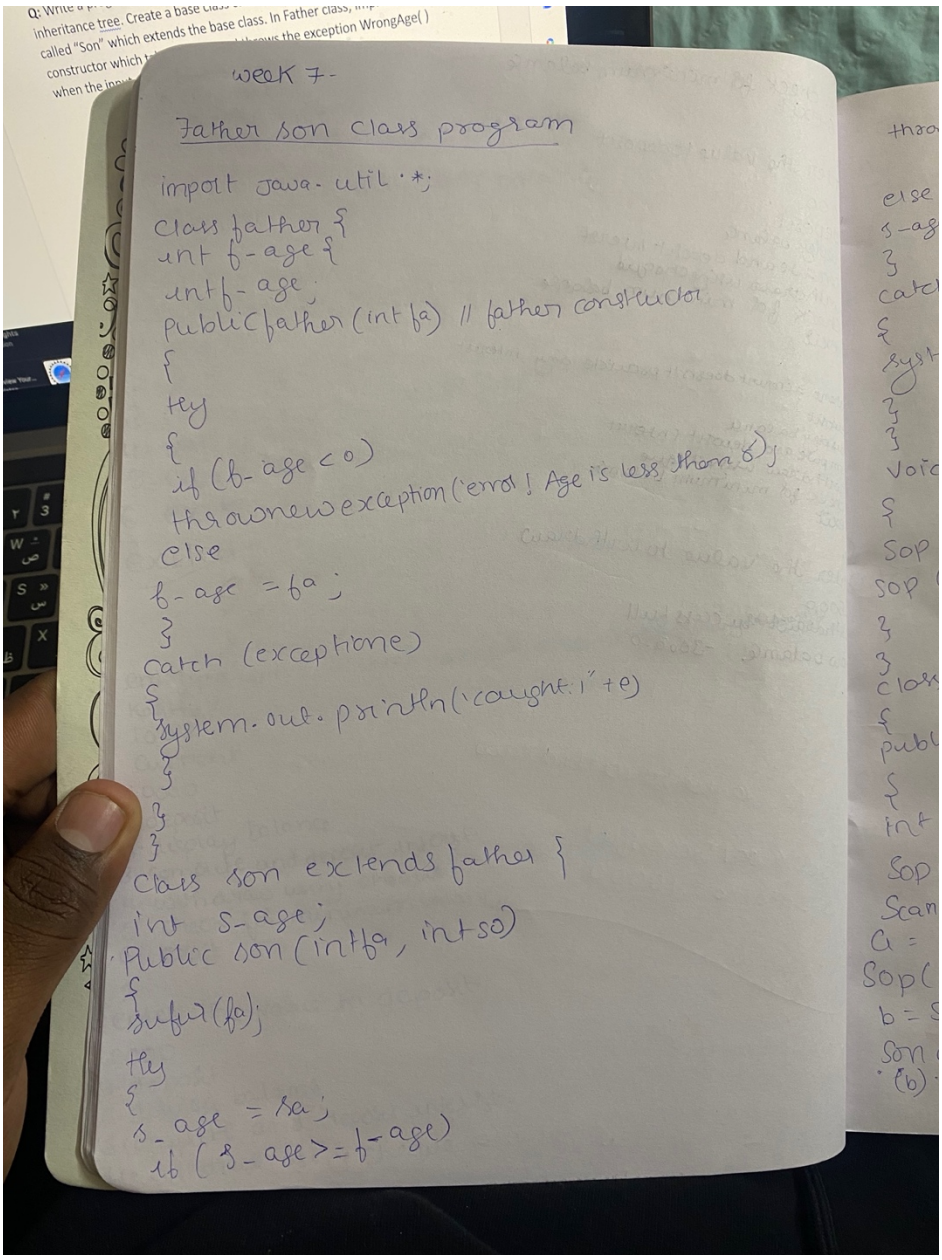
Output for current:

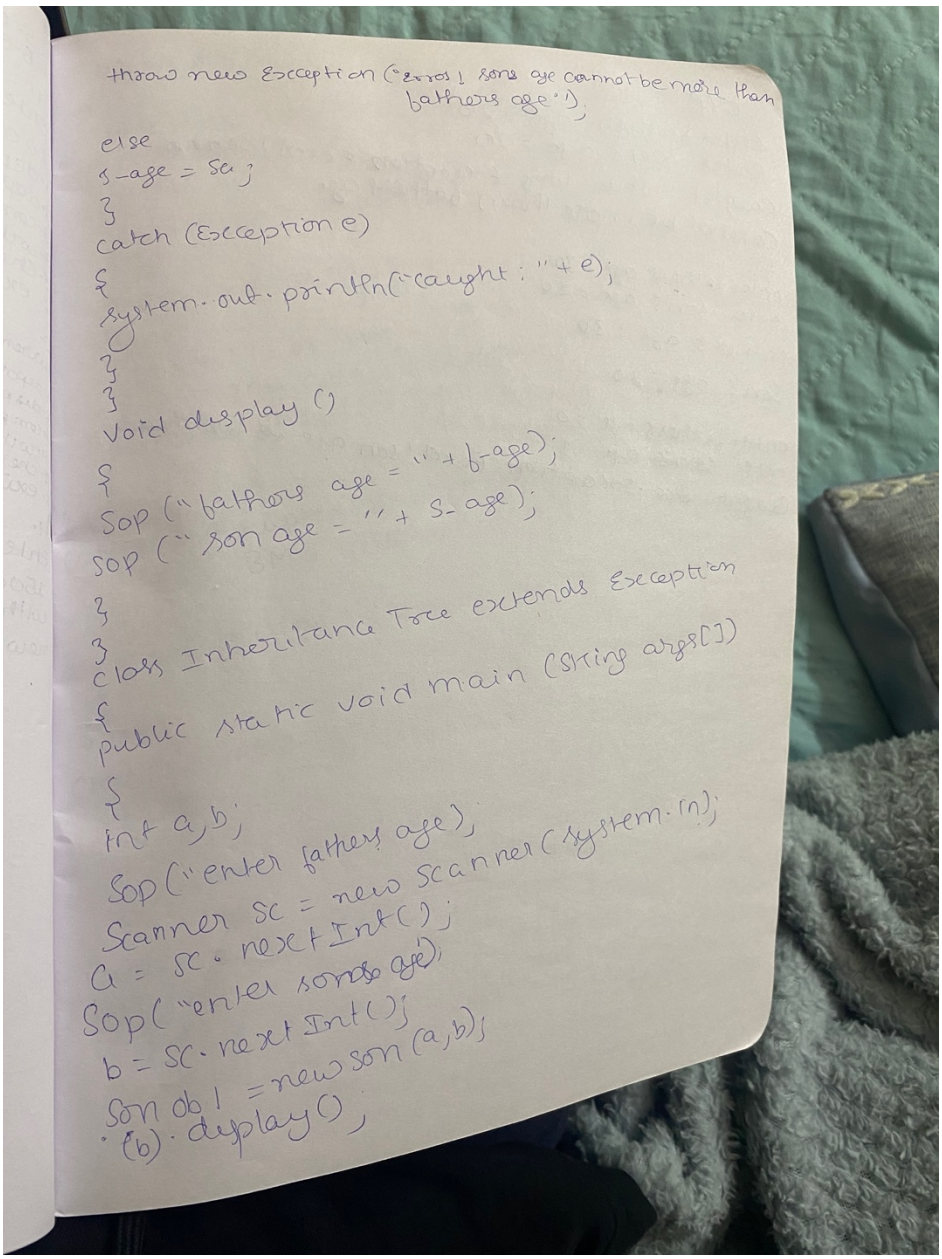
```
enter your name, account number, aacount type(savings/current), balance
zayd
10000000
current
10000
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
1
enter the value to deposit
10
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
2
Balance is: 10010.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
3
Current account doesnt provide any interest
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
4
enter the value to withdraw
200000
withdrawal successful
new balance: -189990.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
5
```

```
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
5
balance lesser than minimum balance, penalty imposed
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
2
Balance is: -190090.0
1.deposit
2.display balance
3.compute and deposit interest
4.withdraw using cheque
5.check for minimum balance
6.exit
6
exited
PS C:\Users\Admin\Desktop\zayd>
```

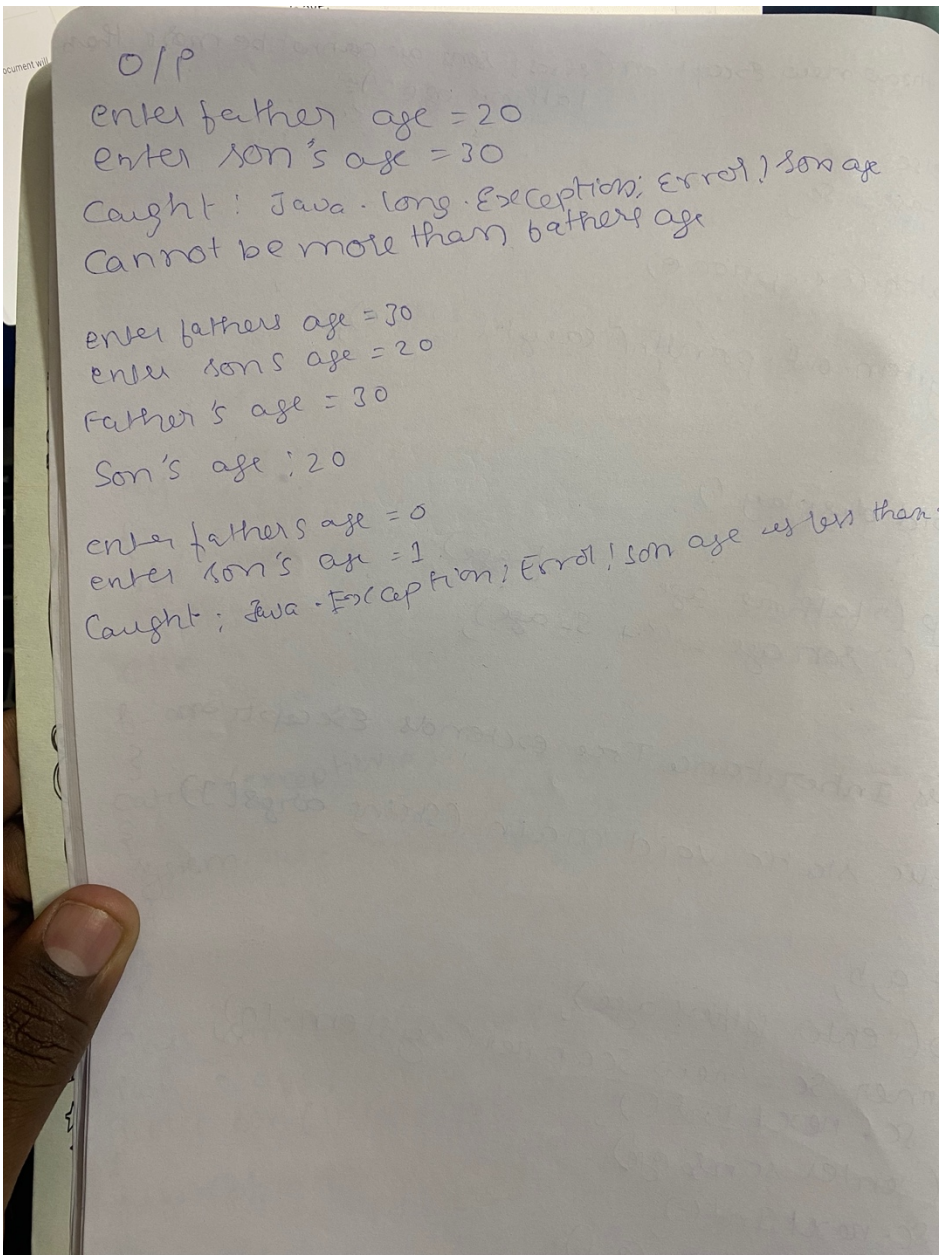
LAB PROGRAM 7:

Q: 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.


Q: Write a
inheritance tree. Create a base
called "Son" which extends the base class. In Father class,
constructor which
when the
throws the exception WrongAge( )
week 7-
Father son class program
import java.util.*;
class father {
int f_age{
int f_age;
Public father (int fa) // father constructor
{
try
{
if (f_age < 0)
throw new exception ('error! Age is less than 0);
else
f_age = fa;
}
catch (exception e)
{
System.out.println('caught:' + e)
}
}
}
Class son extends father {
int s_age;
Public son (int fa, int so)
{
super(fa);
try
{
s_age = sa;
if (s_age >= f_age)
throw
else
catch
{
Sys
}
}
void
{
Sop
sop
}
}
class
{
publ
{
int
Sop
Scan
a =
Sop(
b =
Son
(b)



```
throw new Exception ("Error! Sons age cannot be more than
                      fathers age");
else
s_age = sc;
}
catch (Exception e)
{
System.out.println("caught : " + e);
}
}
Void display ()
{
Sop ("fathers age = " + f_age);
Sop ("son age = " + s_age);
}
class Inheritance Tree extends Exception
{
public static void main (String args[])
{
int a,b;
Sop("enter fathers age");
Scanner sc = new Scanner(System.in);
a = sc.nextInt();
Sop("enter sons age");
b = sc.nextInt();
Son ob1 = new son(a,b);
(b).display();
```

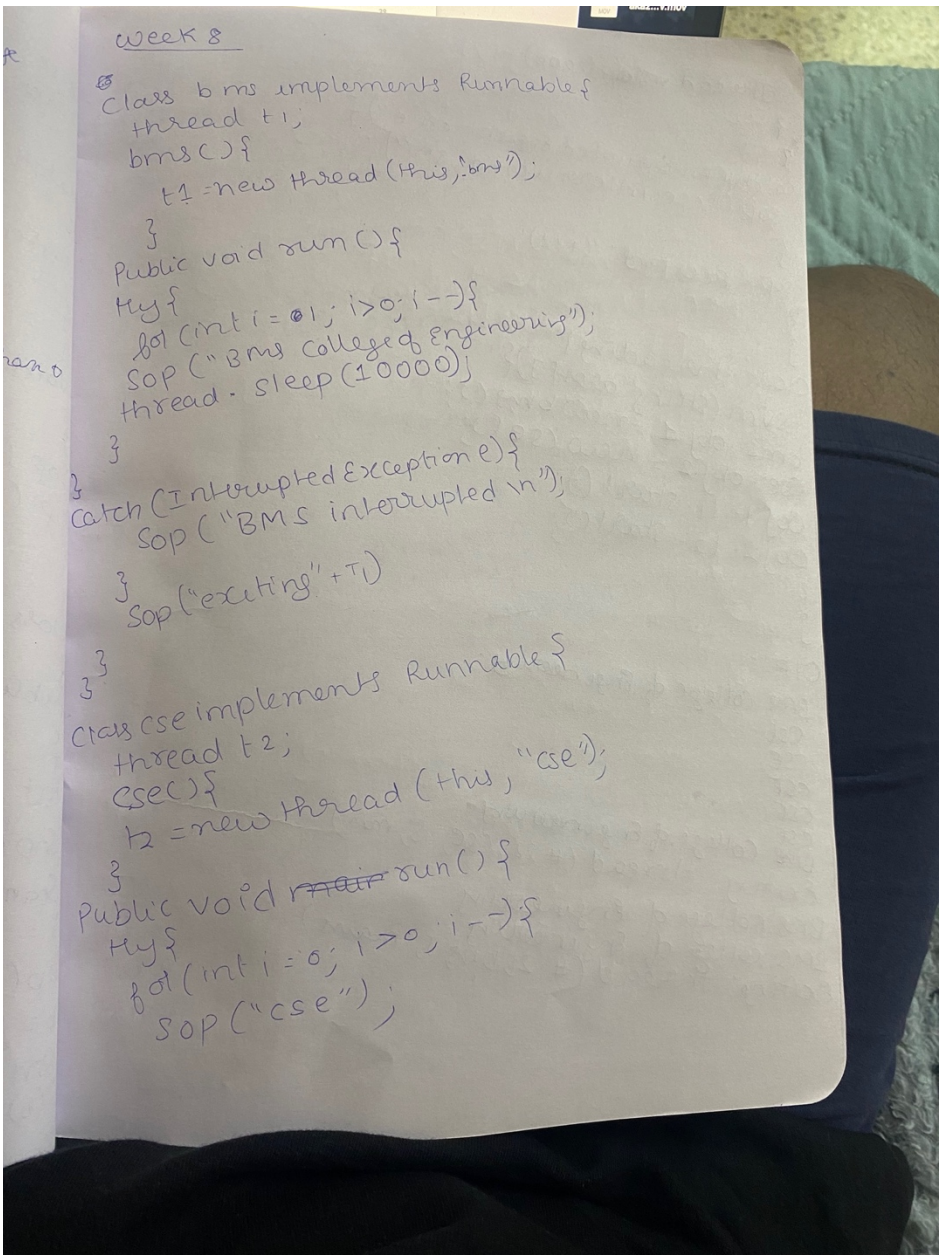

O/P
enter father age = 20
enter son's age = 30
Caught : Java.lang.Exception: Error! Son age
Cannot be more than fathers age
enter fathers age = 30
enter sons age = 20
Father's age = 30
Son's age : 20
enter fathers age = 0
enter son's age = 1
Caught : Java.Exception: Error! son age is less than

Output:

```
C:\Users\Admin\Desktop\f7zz>javac InheritanceTree.java

C:\Users\Admin\Desktop\f7zz>java InheritanceTree
Enter the father's age
25
Enter the son's age
23
Father's age = 25
Son's age = 23

C:\Users\Admin\Desktop\f7zz>java InheritanceTree
Enter the father's age
25
Enter the son's age
26
Caught : java.lang.Exception: Error! Son's age cannot be more than the Father's age
Father's age = 25
Son's age = 26

C:\Users\Admin\Desktop\f7zz>java InheritanceTree
Enter the father's age
0
Enter the son's age
0
Caught : java.lang.Exception: Error! Son's age cannot be more than the Father's age
Father's age = 0
Son's age = 0

C:\Users\Admin\Desktop\f7zz>java InheritanceTree
Enter the father's age
0
Enter the son's age
-1
Caught : java.lang.Exception: Error! Son's age is less than 0
Father's age = 0
Son's age = -1

C:\Users\Admin\Desktop\f7zz>
```

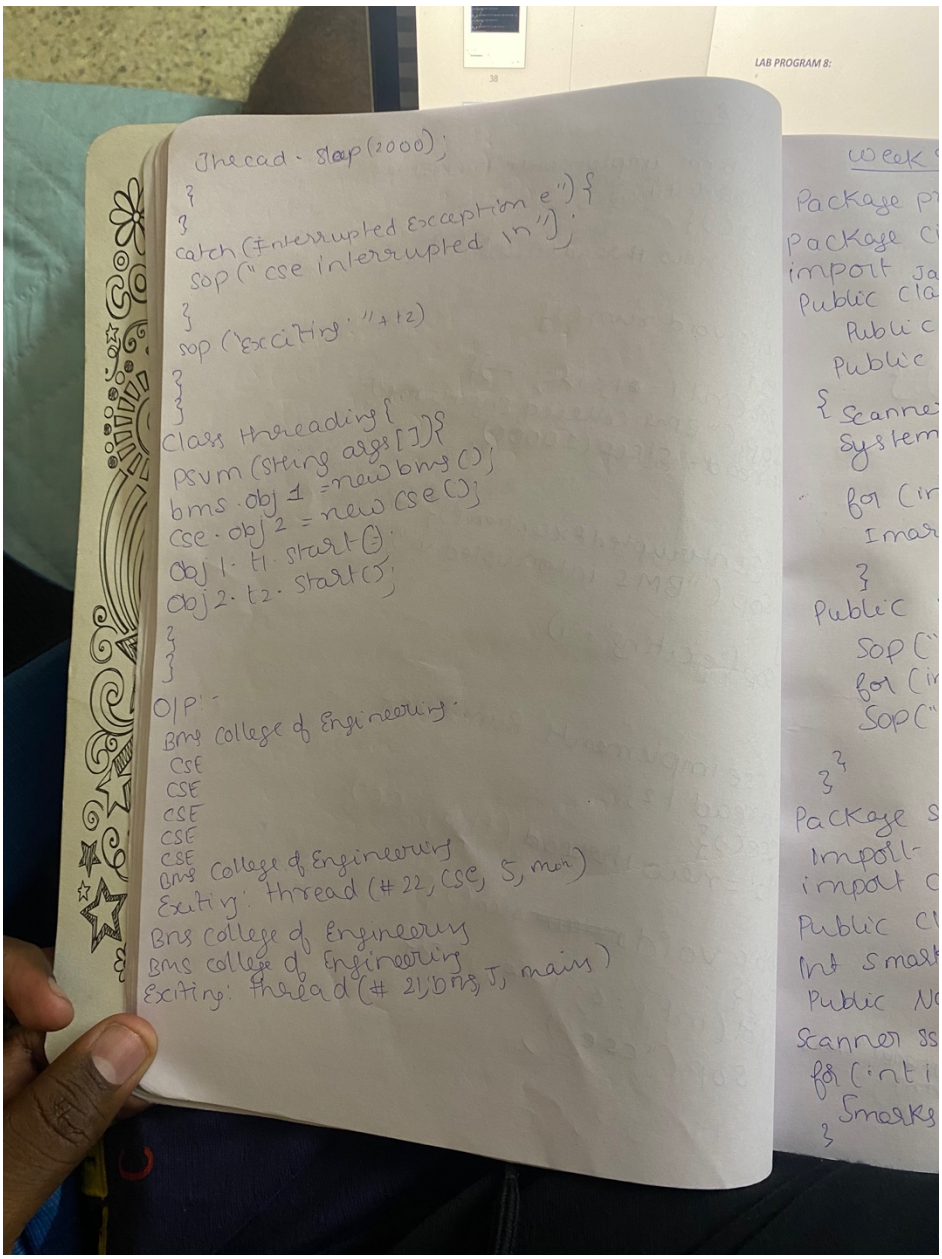
LAB PROGRAM 8:

Q: 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.

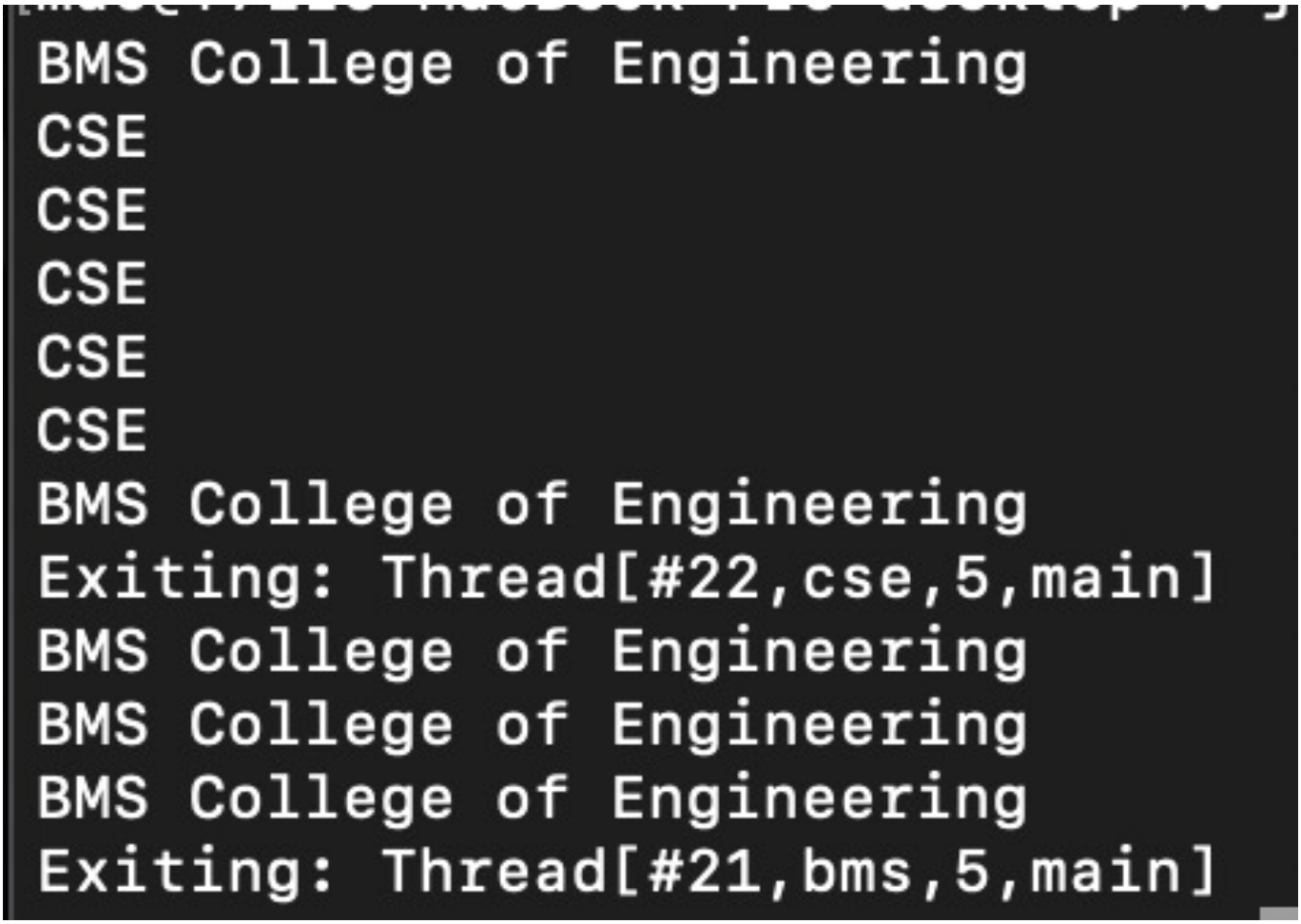


Week 8

```
Class bms implements Runnable{
  thread t1;
  bms(){
    t1 = new thread(this,"bms");
  }
  Public void run(){
  try{
    for(int i=10; i>0; i--){
    Sop("Bms College of Engineering");
    thread.sleep(10000);
  }
}
catch(InterruptedException e){
    Sop("BMS interrupted \n");
  }
  Sop("exiting"+T1)
}}
Class cse implements Runnable{
  thread t2;
  cse(){
    t2 = new thread(this, "cse");
  }
  Public void run(){
  try{
    for(int i=0; i>0; i--){
    Sop("cse");
```

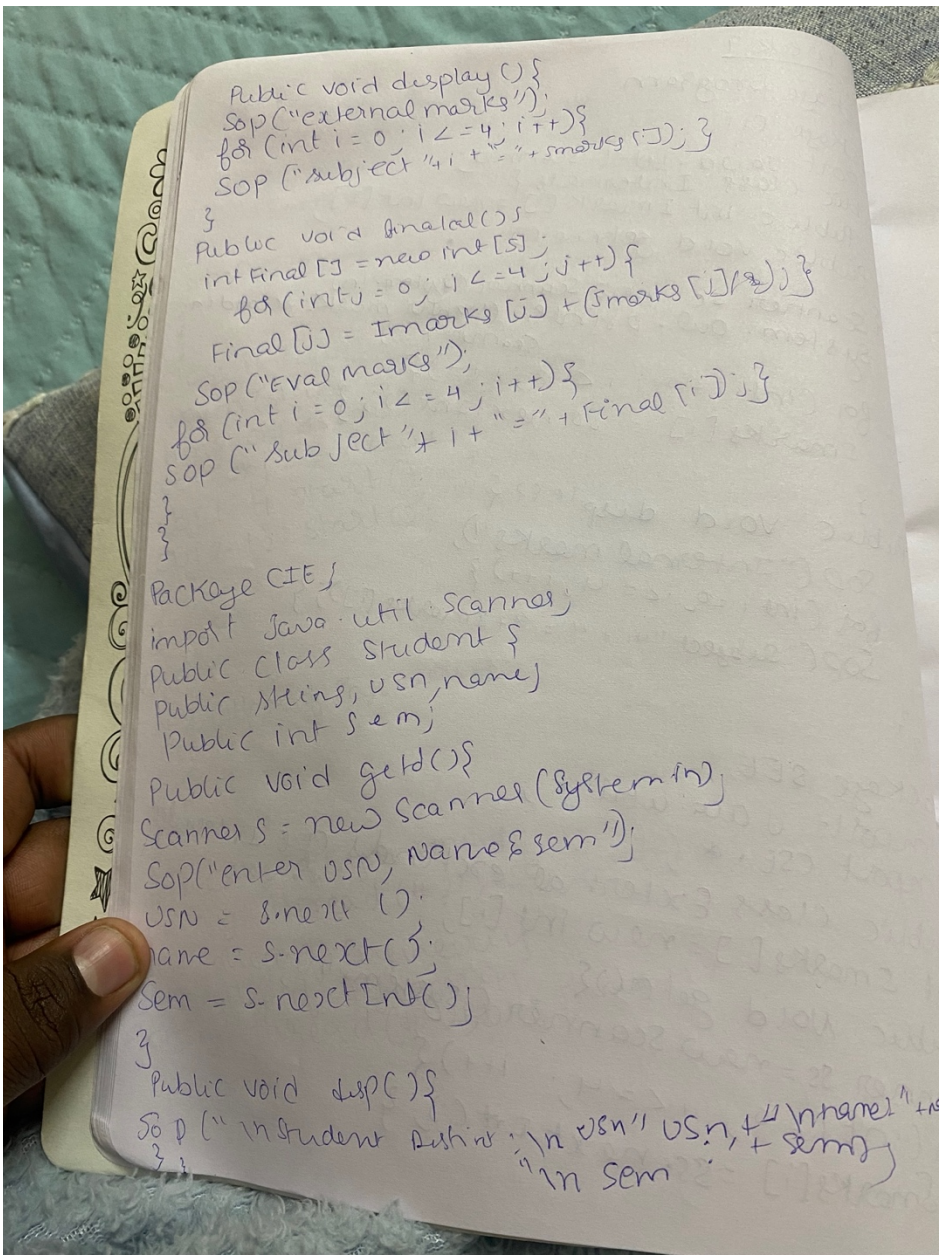

LAB PROGRAM 8:
Thread.sleep(2000);
}
}
catch (Interrupted Exception e) {
sop("cse interrupted in");
}
sop ('Exciting: "+t2)
}
}
Class threading {
psvm (String args[])}
bms.obj1 = new bms();
cse.obj2 = new cse();
obj1.t1.start();
obj2.t2.start();
}
}
O/P:
Bms College of Engineering
CSE
CSE
CSE
CSE
CSE
Bms College of Engineering
Exiting: thread (#22, CSE, 5, main)
Bms College of Engineering
Bms College of Engineering
Exiting: thread (#21, BMS, 5, main)

Output:

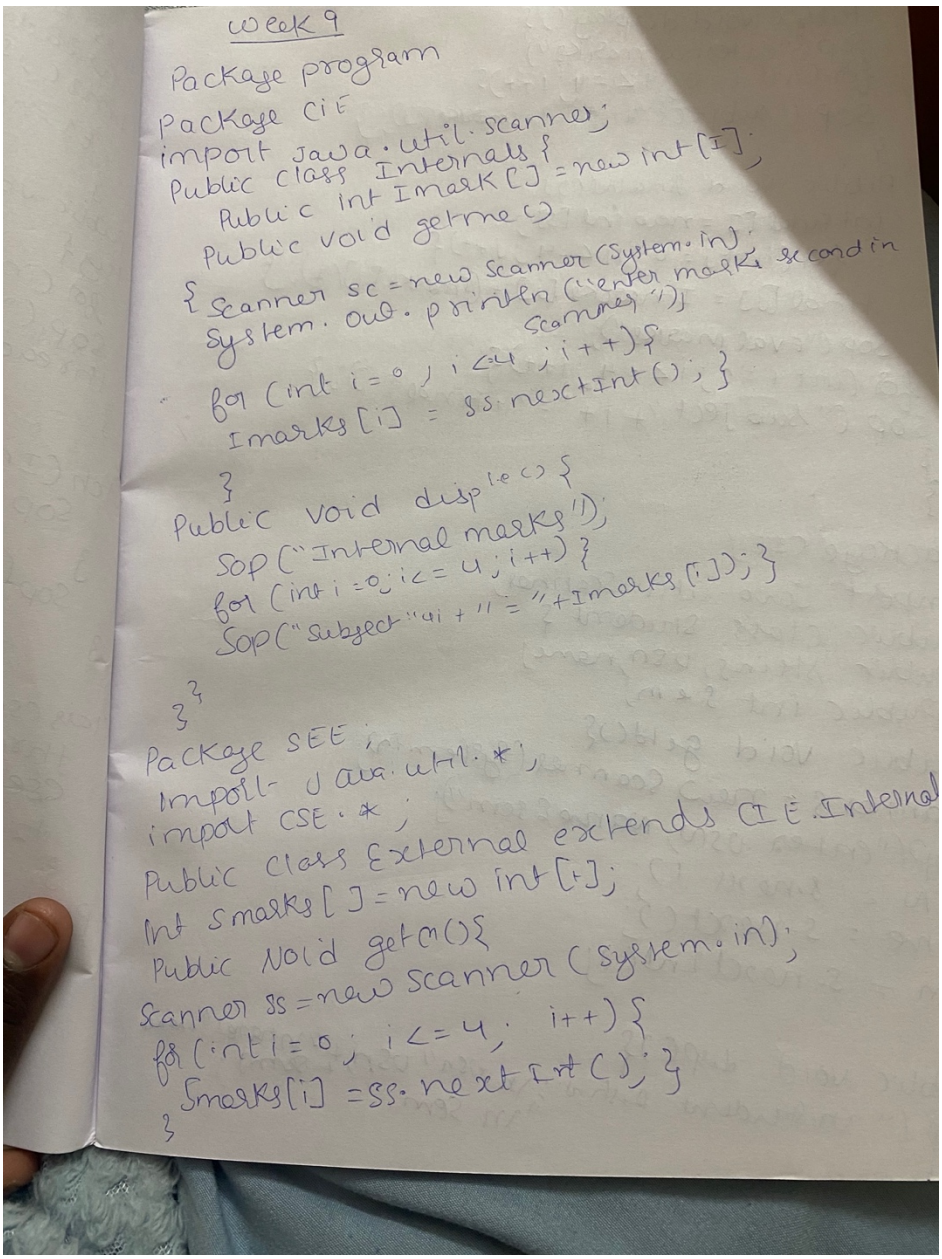


LAB PROGRAM 9:

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



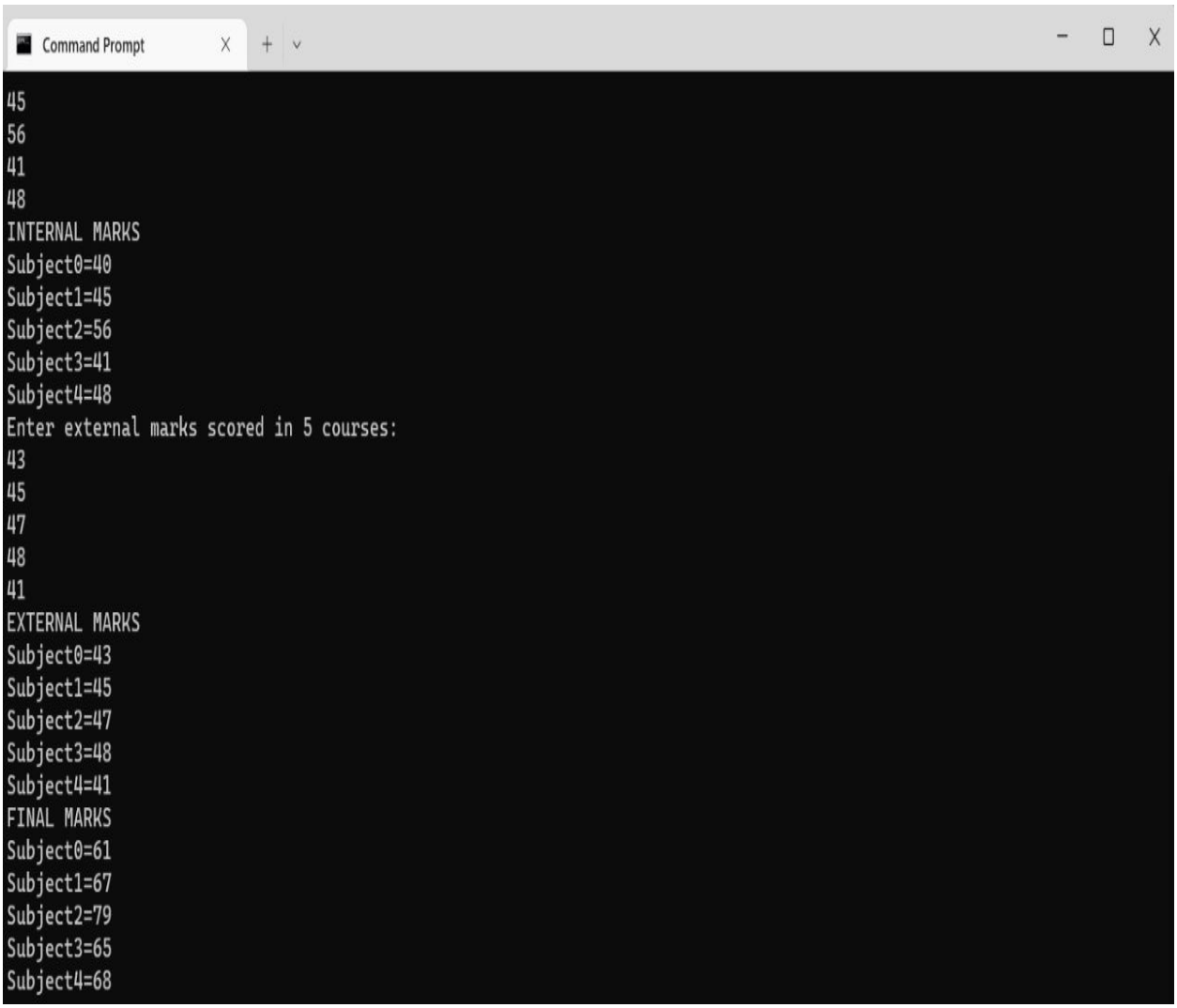
```
Public void display(){
Sop("external marks");
for(int i=0; i<=4; i++){
Sop("subject"+i+"="+smarks[i]);}
}
Public void finalal(){
int Final[] = new int[5];
for(int j=0; j<=4; j++){
Final[j] = Imarks[j] + (Smarks[j]/2);}
Sop("Eval marks");
for(int i=0; i<=4; i++){
Sop("subject"+i+"="+Final[i]);}
}
}

Package CIE;
import java.util.Scanner;
Public class Student{
Public String usn, name;
Public int sem;
Public void getd(){
Scanner s = new Scanner(System.in);
Sop("enter USN, Name & sem");
usn = s.next();
name = s.next();
Sem = s.nextInt();
}
Public void disp(){
Sop("\n Student Details: \n USN" usn+"\n name"+name
"\n sem : "+sem);
}}
```



```
Week 9
Package program
Package CIE
import java.util.scanner;
Public class Internals {
   Public int Imark[] = new int[5];
   Public void getme()
{ Scanner sc = new Scanner (System.in);
System.out.println("enter marks seccond in
                        Scanner");
   for (int i = 0 ; i <=4 ; i++){
    Imarks[i] = ss.nextInt(); }
   }
Public void display() {
   Sop("Internal marks");
   for (int i =0; i<= 4; i++) }
   Sop("subject"+i+1+"="+Imarks[i]); }
}
Package SEE;
import java.util.*;
import CSE.*;
Public class External extends CIE.Internal
Int Smarks[] = new int[5];
Public Void get()
Scanner ss = new Scanner (System.in);
for (int i = 0; i <= 4; i++) {
  Smarks[i] = ss.nextInt(); }
}
```

Output:

```
Enter USN, NAME & SEM
1BM21CS254
Z
1

Student Details:
USN:1BM21CS254
NAME:Z
SEM:1
Enter marks scored in 5 courses:
40
45
56
41
48
INTERNAL MARKS
Subject0=40
Subject1=45
Subject2=56
Subject3=41
Subject4=48
Enter external marks scored in 5 courses:
```


Command Prompt
45
56
41
48
INTERNAL MARKS
Subject0=40
Subject1=45
Subject2=56
Subject3=41
Subject4=48
Enter external marks scored in 5 courses:
43
45
47
48
41
EXTERNAL MARKS
Subject0=43
Subject1=45
Subject2=47
Subject3=48
Subject4=41
FINAL MARKS
Subject0=61
Subject1=67
Subject2=79
Subject3=65
Subject4=68