LAB RECORD

OOJ LAB

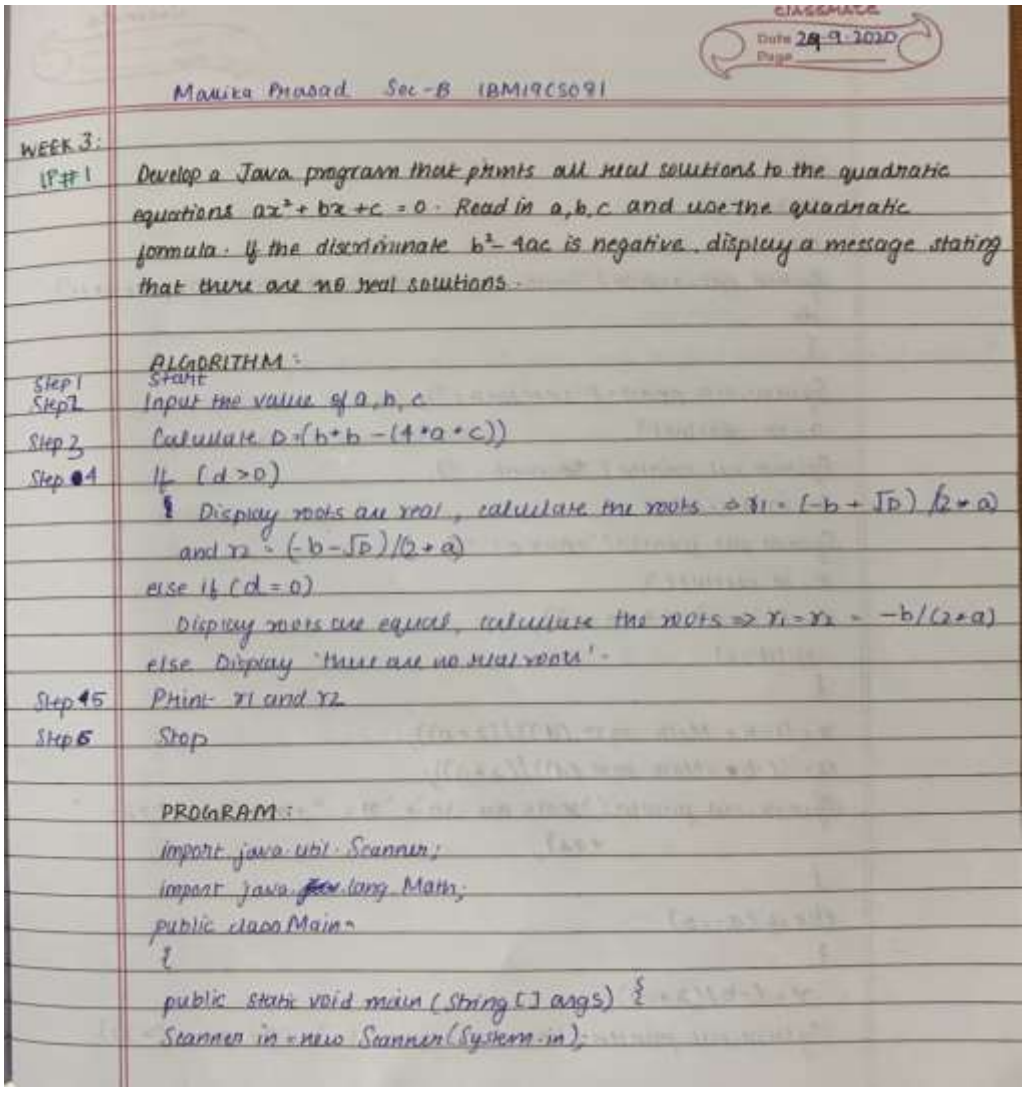
Mallika Prasad

1BM19CS081

LAB 1-

Develop a Java program that prints all the real solutions to the quadratic equations ax^2+bx+c. 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.

Observation-



Mallika Prasad Sec-B 1BM19CS091

WEEK 3:

LP#1 Develop a Java program that prints all real solutions to the quadratic equations $ax^2 + bx + c = 0$. 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.

ALGORITHM:

Step 1 Start

Step 2 Input the value of a, b, c

Step 3 Calculate D = (b*b - (4*a*c))

Step 4 If (d > 0)

Display roots are real, calculate the roots $\Rightarrow r1 = (-b + \sqrt{D})/(2*a)$ and $r2 = (-b - \sqrt{D})/(2*a)$

else if (d = 0)

Display roots are equal, calculate the roots $\Rightarrow r1 = r2 = -b/(2*a)$

else Display 'There are no real roots'.

Step 5 Print r1 and r2

Step 6 Stop

PROGRAM:

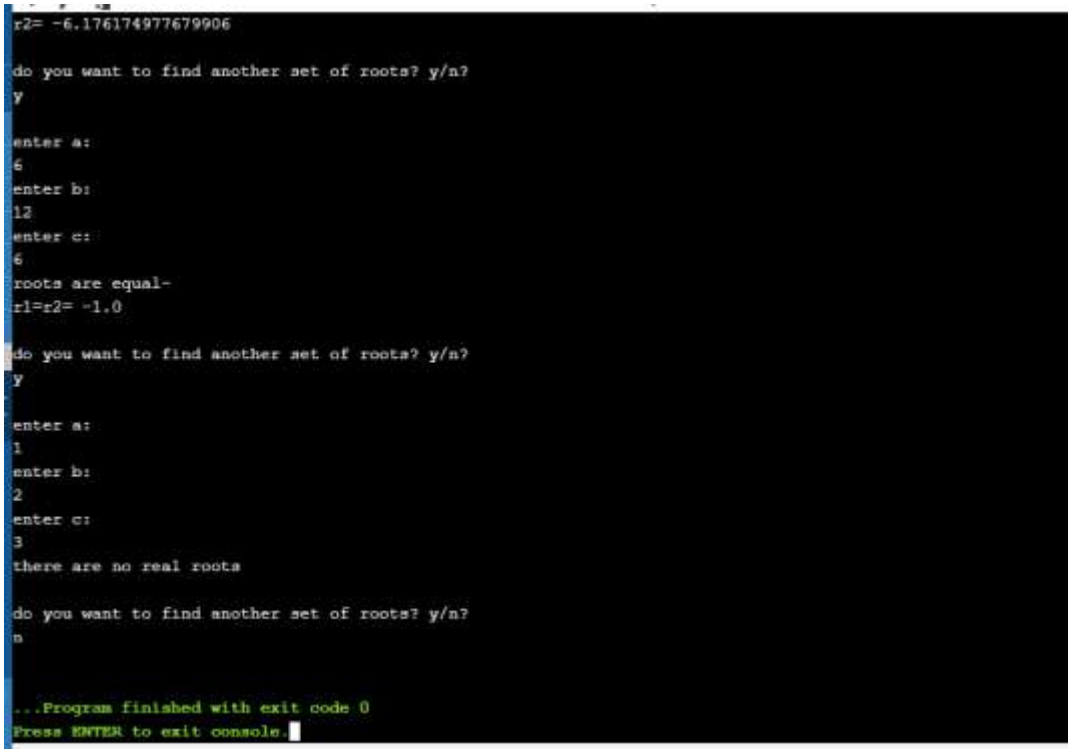
```
import java.util.Scanner;
import java.lang.Math;
public class Main
{
public static void main (String[] args) {
Scanner in = new Scanner(System.in);
```

```
int a,b,c;
double r1, r2, d;
char ch;
System.out.println("Solution of Quadratic equation -ax^2+bx+c");
do
{
System.out.println("\nenter a:");
a=in.nextInt();
System.out.println("enter b:");
b=in.nextInt();
System.out.println("enter c:");
c=in.nextInt();
d=((b*b)-(4*a*c));
if(d>0)
{
r=((-b+Math.sqrt(d))/(2*a));
r2=((-b-Math.sqrt(d))/(2*a));
System.out.println("roots are -\n"+"r=" +r+"\n"+"r2="
                   +r2);
}
else if(d==0)
{
r=(-b/(2*a));
System.out.println("roots are equal-\n"+"r1=r2="+r1);
```

```
}
else
{
System.out.println("there are no real roots");
}
System.out.println("\n"+"do you want to find another set of roots?
y\n?);
ch=in.next().charAt(0);
}
while(ch=='y');
}
}
```

Output-

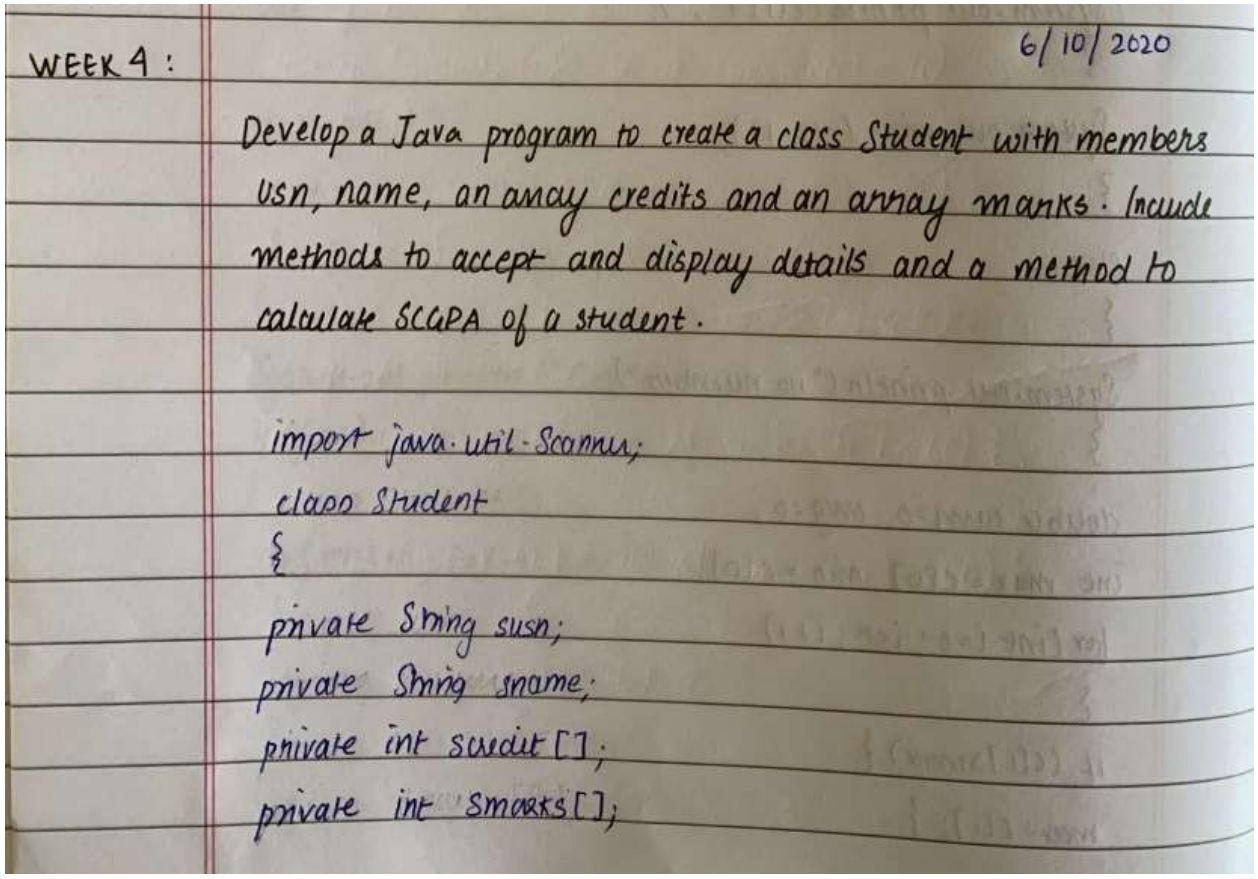

Input
Solution of Quadratic equation- ax^2+bx+c
enter a:
2
enter b:
13
enter c:
4
roots are-
r1= -0.3238250223200936
r2= -6.176174977679906
do you want to find another set of roots? y/n?
y
enter a:
6
enter b:
12
enter c:
6
roots are equal-
r1=r2= -1.0
do you want to find another set of roots? y/n?
y
enter a:
1


r2= -6.176174977679906
do you want to find another set of roots? y/n?
y
enter a:
6
enter b:
12
enter c:
6
roots are equal-
r1=r2= -1.0
do you want to find another set of roots? y/n?
y
enter a:
1
enter b:
2
enter c:
3
there are no real roots
do you want to find another set of roots? y/n?
n
...Program finished with exit code 0
Press ENTER to exit console.

LAB 2-

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

Observation-



WEEK 4 : 6/10/2020

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
{
private String susn;
private String sname;
private int scredit[];
private int smarks[];
```

```
void getDetails()
{
System.out.println("enter student details :");
Scanner in = new Scanner(System.in);
scredit = new int[5];
smarks = new int[5];
susn = in.next();
sname = in.next();
for (int i=0; i<5; i++){
System.out.println("credit for sub "+(i+1)+":");
scredit[i] = in.nextInt();
}
for (int i=0; i<5; i++){
System.out.println("marks for sub "+(i+1)+":");
smarks[i] = in.nextInt();
}
}
void printDetails()
{
System.out.println("USN : "+susn);
System.out.println("Name : "+sname);
for (int i=0; i<5; i++){
System.out.print("credits for sub "+(i+1)+":");
System.out.println(scredit[i]); }
```

```
for (int i=0; i<5; i++) {
System.out.print("marks for sub " + (i+1) + ":");
System.out.println(smarks[i]); }
}

void Sgpa() {
int sum=0, sum2=0;
double g=0;
double sgpa;
for (int i=0; i<5; i++) {
if (smarks[i]>=90) {
   g=10; }
else if (smarks[i]>=80 && smarks[i]<90)
 {g=9;}
else if (smarks[i]>=70 && smarks[i]<80) {
  g=8; }
else if (smarks[i]>=60 && smarks[i]<70) {
  g=7; }
else if (smarks[i]>=50 && smarks[i]<60) {
  g=6; }
else if (smarks[i]>=10 && smarks[i]<50) {
  g=1; }
else if (smarks[i]<10) {
  g=0;
}
```

```
sum+ = g * subcredit[i];
sum2+ = subcredit[i];
}
sgpa = sum/sum2;
System.out.print("SGPA of student: ");
System.out.println(sgpa);
}
}

public class Main
{
public static void main(String ss[])
{
Student s1 = new Student();
s1.getDetails();
s1.printDetails();
s1.Sgpa();
}
}
```

Algorithm :

Step1 : Start

Step2 : Input student details i.e. usn, name, credits and marks(of each of 5 subjects in 2 different arrays)

Step3 : Display the student details

Step4 : if smarks >=90 , g = 10

else if smarks >=80 && smarks <90, g=9

else if smarks >=70 && smarks <80, g=8

else if smarks >=60 && smarks <70, g=7

else if smarks >=50 && smarks <60, g=6

else if smarks >=40 && smarks <50, g=4

else if smarks <40, g=0

Get value of g and calculate sum of (g * credits) (sum)

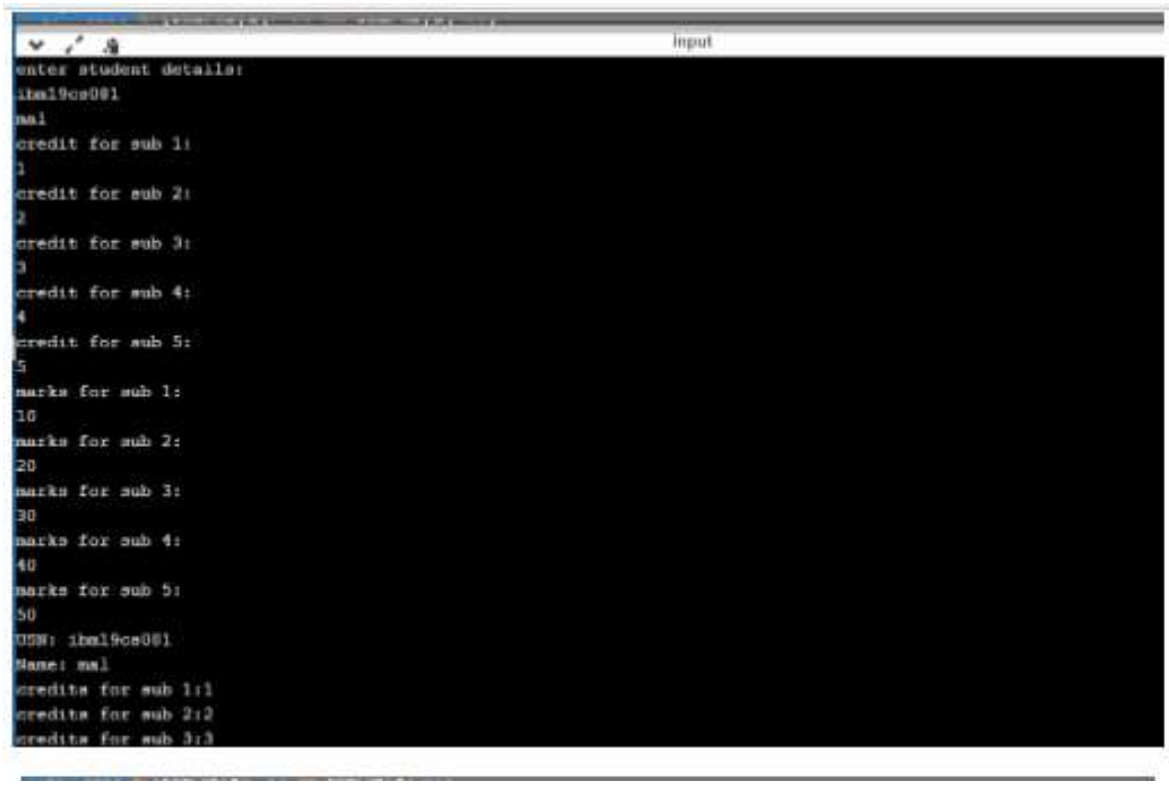
Step 5 : Calculate sgpa Get to sum of credits (sum2)

Step 6 : Calculate sgpa = sum/sum2

Step 7 : Print sgpa of student

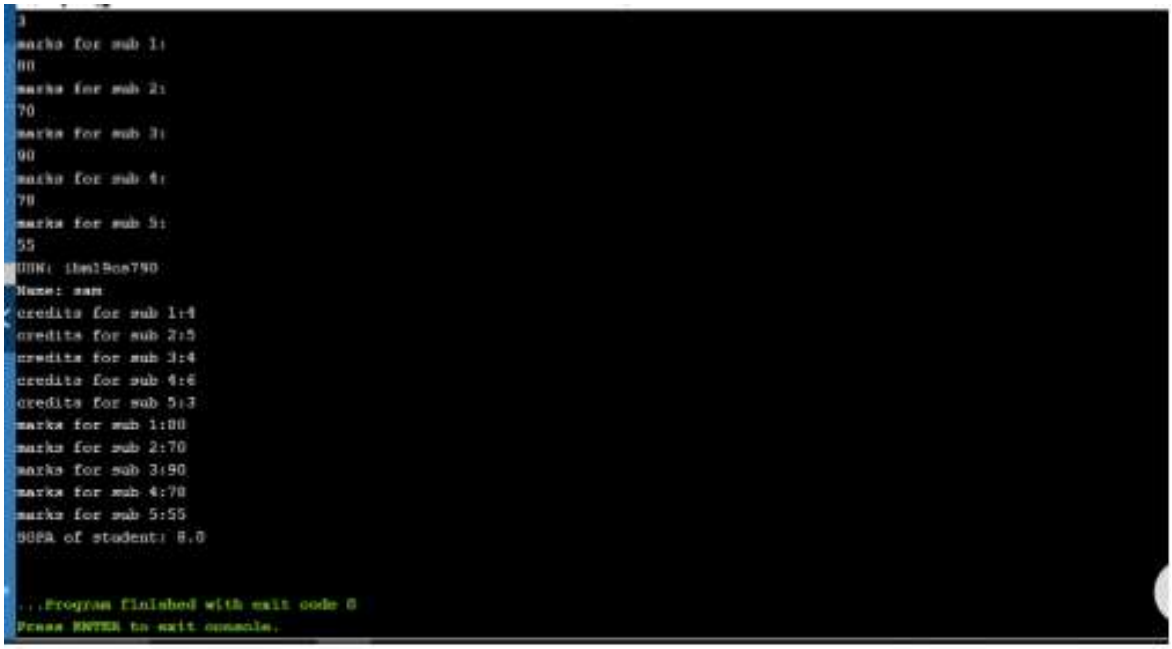
Step 8 : Stop

Output-


Input
enter student details:
ibml9cs001
mal
credit for sub 1:
1
credit for sub 2:
2
credit for sub 3:
3
credit for sub 4:
4
credit for sub 5:
5
marks for sub 1:
10
marks for sub 2:
20
marks for sub 3:
30
marks for sub 4:
40
marks for sub 5:
50
USN: ibml9cs001
Name: mal
credits for sub 1:1
credits for sub 2:2
credits for sub 3:3


3
marks for sub 1:
10
marks for sub 2:
20
marks for sub 3:
30
marks for sub 4:
40
marks for sub 5:
50
USN: ibml9cs001
Name: mal
credits for sub 1:1
credits for sub 2:2
credits for sub 3:3
credits for sub 4:4
credits for sub 5:5
marks for sub 1:10
marks for sub 2:20
marks for sub 3:30
marks for sub 4:40
marks for sub 5:50
SGPA of student: 3.0
...Program finished with exit code 0
Press ENTER to exit console.

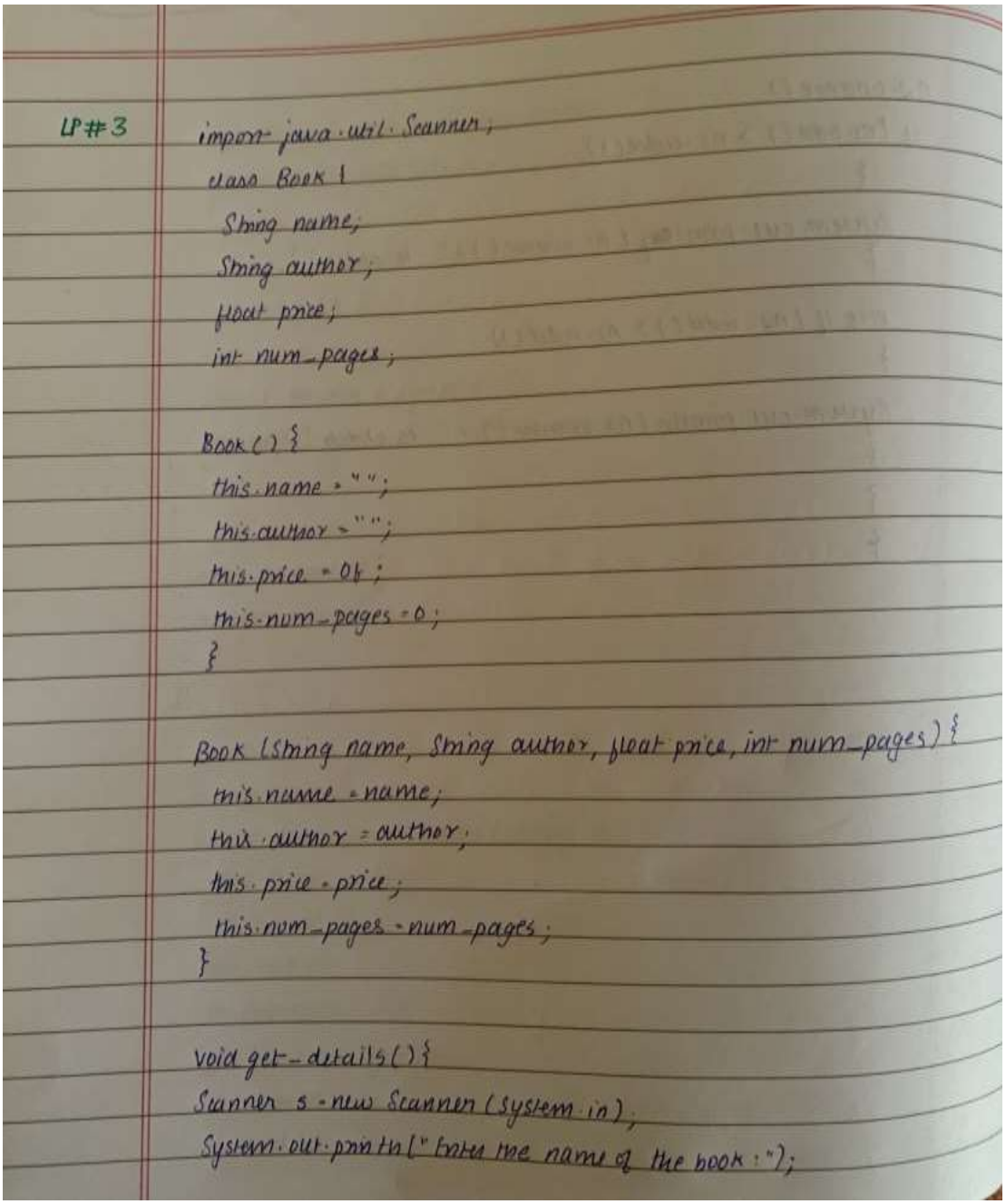

input
enter student details:
1bm19cs790
sam
credit for sub 1:
4
credit for sub 2:
5
credit for sub 3:
4
credit for sub 4:
6
credit for sub 5:
3
marks for sub 1:
80
marks for sub 2:
70
marks for sub 3:
90
marks for sub 4:
70
marks for sub 5:
55
USN: 1bm19cs790
Name: sam
credits for sub 1:4
credits for sub 2:5
credits for sub 3:4


3
marks for sub 1:
80
marks for sub 2:
70
marks for sub 3:
90
marks for sub 4:
70
marks for sub 5:
55
USN: 1bm19cs790
Name: sam
credits for sub 1:4
credits for sub 2:5
credits for sub 3:4
credits for sub 4:6
credits for sub 5:3
marks for sub 1:80
marks for sub 2:70
marks for sub 3:90
marks for sub 4:70
marks for sub 5:55
SGPA of student: 8.0
...Program finished with exit code 0
Press ENTER to exit console.

LAB 3-

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

Observation-

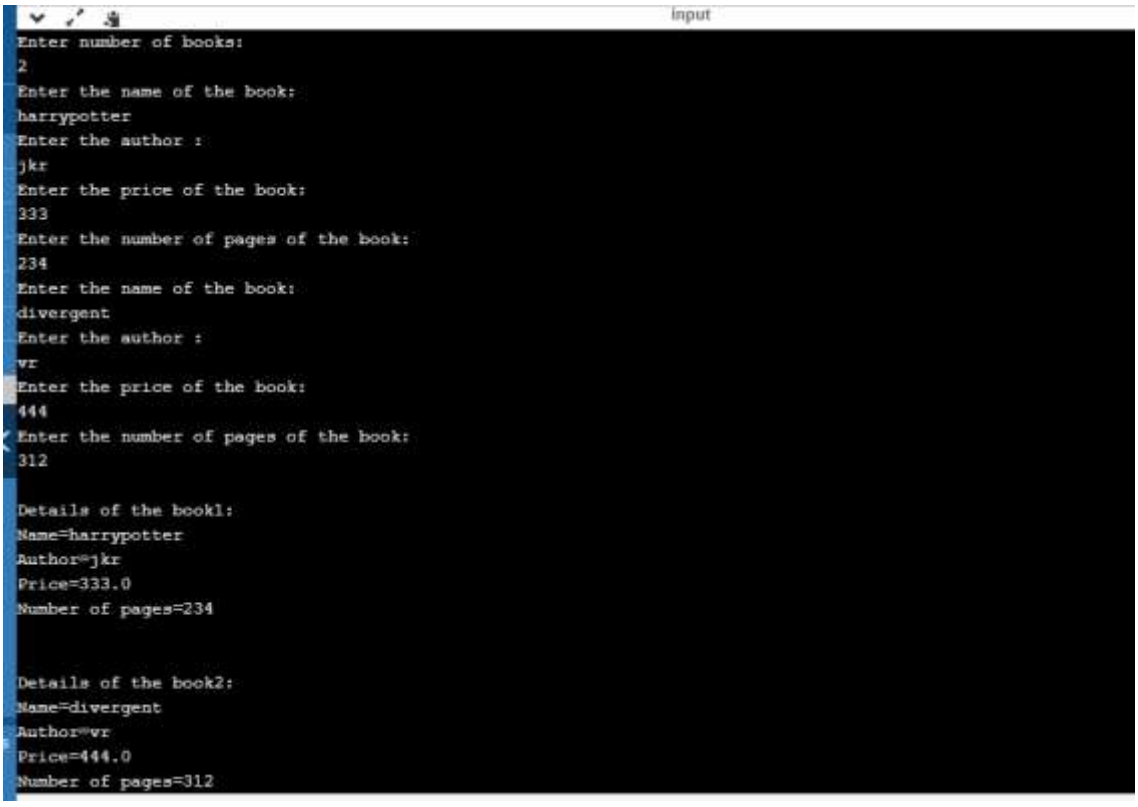


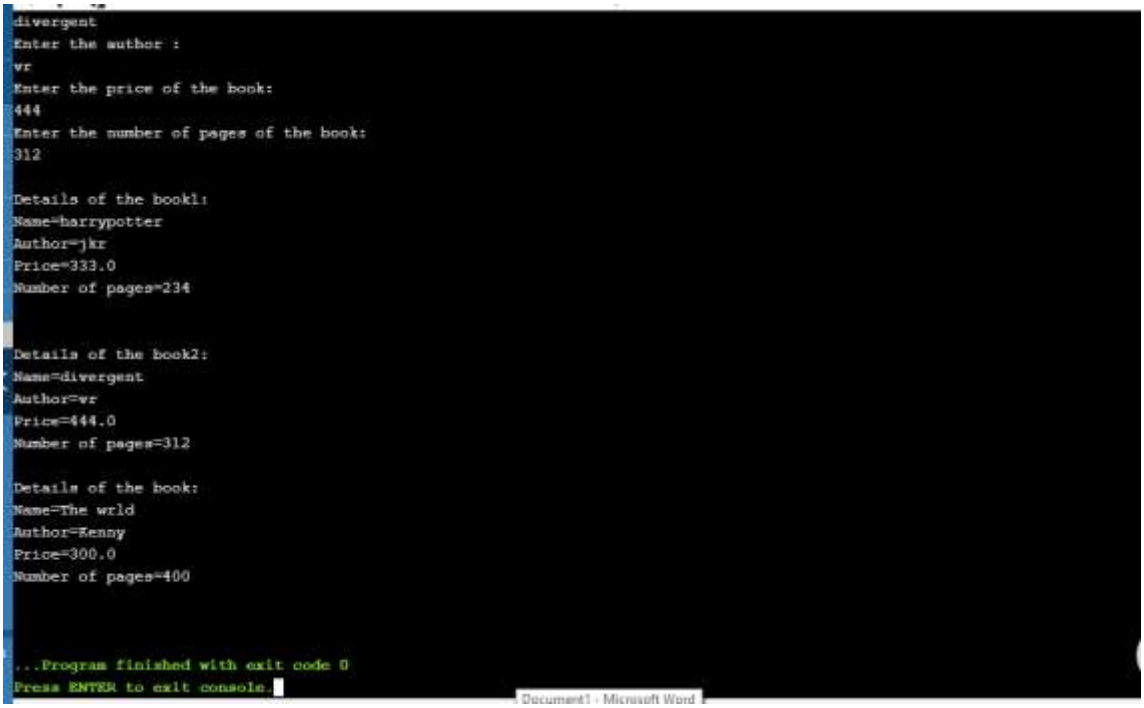
```
LP#3
import java.util.Scanner;
class Book {
 String name;
 String author;
 float price;
 int num_pages;

 Book() {
 this.name = "";
 this.author = "";
 this.price = 0f;
 this.num_pages = 0;
 }

 Book (String name, String author, float price, int num_pages) {
 this.name = name;
 this.author = author;
 this.price = price;
 this.num_pages = num_pages;
 }

 void get_details() {
 Scanner s = new Scanner (System.in);
 System.out.println(" Enter the name of the book : ");
```

```
name = s.nextLine();
System.out.println("Enter the author :");
author = s.nextLine();
System.out.println("Enter price of the book: ");
price = s.nextFloat();
System.out.println("Enter number of pages of the book :");
num_pages = s.nextInt();
}

void set_details(String n, String a, float p, int np){
 this.name = n;
 this.author = a;
 this.price = p;
 this.num_pages = np;
}

 public String toString(){
 return("Name = "+name+"\n Author = "+author+"\nPrice = "+
 price+"\n Number of pages = "+num_pages+"\n");
 }
}

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

```
int n;
Scanner s = new Scanner (System.in);
System.out.println("Enter number of books:");
 n = s.nextInt();
Book []b = new Book[n];
for (int i=0; i<n ; i++) {
  b[i] = new Book();
  b[i].get_details();
  }
for (int i=0; i<n; i++) {
  System.out.println ("Details of the book" + (i+1) + ":");
  System.out.println (b[i]);
  }
  Book book1 = new Book();
  book1.set_details("The ~~wrong~~ world", "Kenny", 300, 400);
  System.out.println("Details of the book:");
   System.out.println (book1);
 }
}
```

Output-

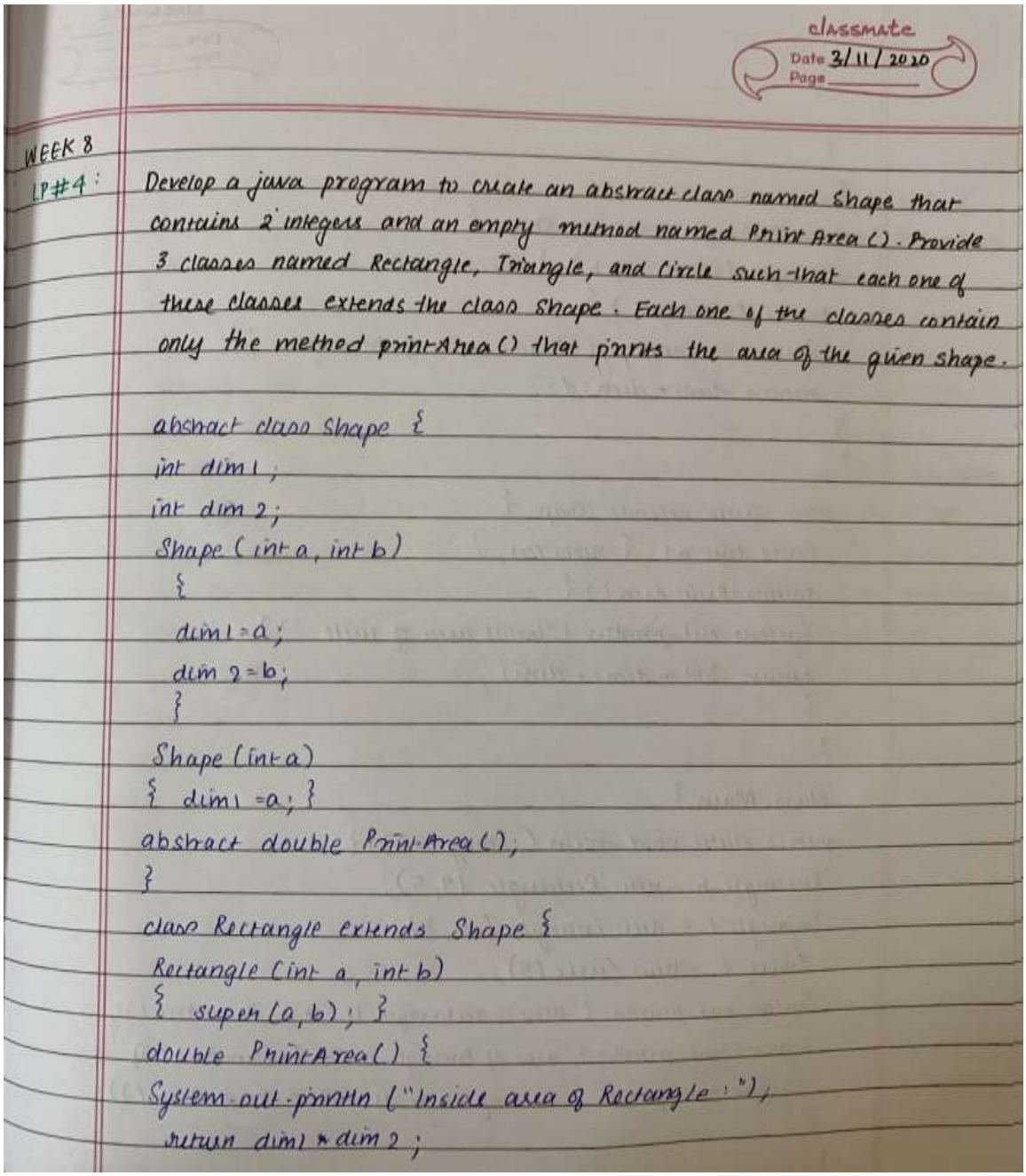

Input
Enter number of books:
2
Enter the name of the book:
harrypotter
Enter the author :
jkr
Enter the price of the book:
333
Enter the number of pages of the book:
234
Enter the name of the book:
divergent
Enter the author :
vr
Enter the price of the book:
444
Enter the number of pages of the book:
312
Details of the book1:
Name=harrypotter
Author=jkr
Price=333.0
Number of pages=234
Details of the book2:
Name=divergent
Author=vr
Price=444.0
Number of pages=312


divergent
Enter the author :
vr
Enter the price of the book:
444
Enter the number of pages of the book:
312
Details of the book1:
Name=harrypotter
Author=jkr
Price=333.0
Number of pages=234
Details of the book2:
Name=divergent
Author=vr
Price=444.0
Number of pages=312
Details of the book:
Name=The wrld
Author=Kenny
Price=300.0
Number of pages=400
...Program finished with exit code 0
Press ENTER to exit console.

LAB 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.

Observation-



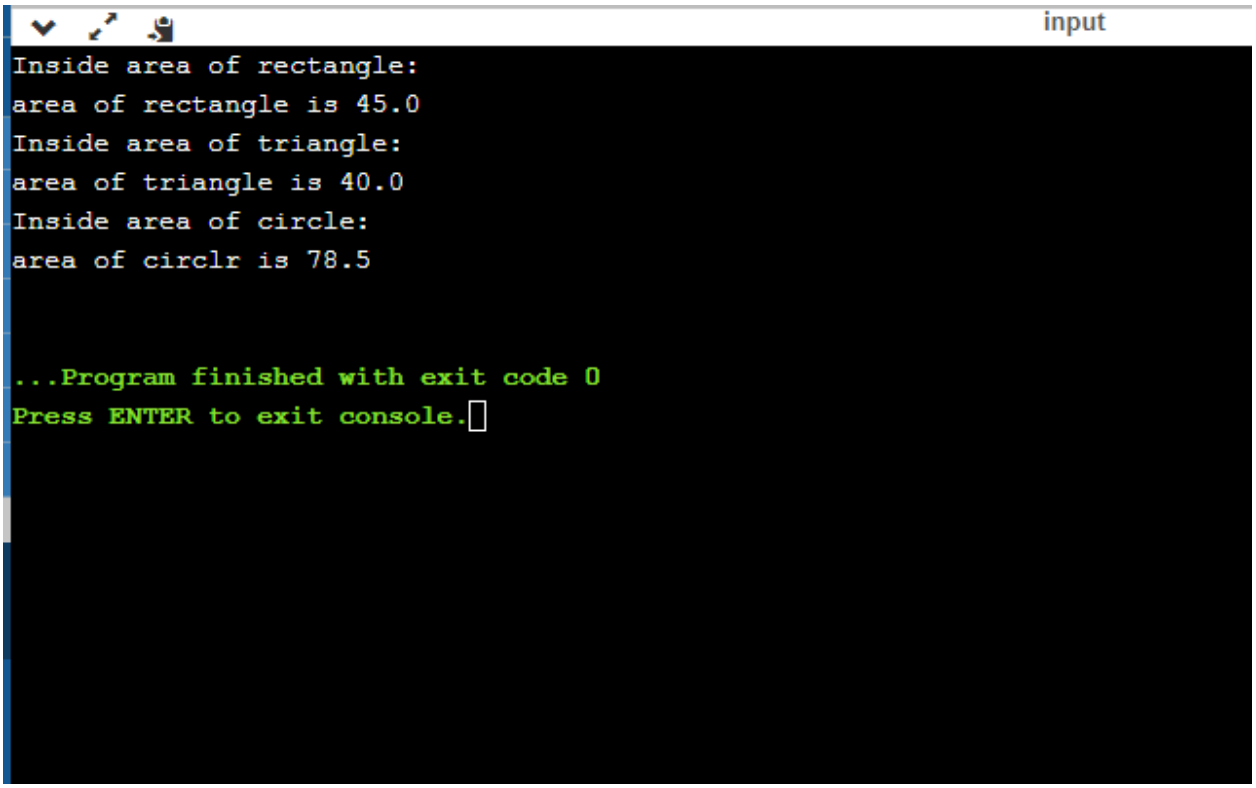
classmate
Date 3/11/2020
Page

WEEK 8
LP#4: Develop a java program to create an abstract class named Shape that contains 2 integers and an empty method named PrintArea(). Provide 3 classes named Rectangle, Triangle, and Circle such that each one of these 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 {
int dim1;
int dim2;
Shape (int a, int b)
 {
 dim1=a;
 dim2=b;
 }
Shape (int a)
{ dim1=a; }
abstract double PrintArea();
}
class Rectangle extends Shape {
Rectangle (int a, int b)
{ super(a,b); }
double PrintArea() {
System.out.println("Inside area of Rectangle:");
return dim1*dim2;
```

```
}
}
class Triangle extends Shape {
 Triangle (int a, int b) { super (a, b); }
 double PrintArea () {
  System.out.println ("Inside area of triangle: ");
  return dim1 * dim2 / 2;
 }
}

 class Circle extends Shape {
  Circle (int a) { super (a); }
  double PrintArea () {
   System.out.println ("Inside area of circle : ");
   return 3.14 * dim1 * dim1;
  }
 }

 class Main {
 public static void main ( String ss[] ) {
  Rectangle r = new Rectangle (9, 5);
  Triangle t = new Triangle (10, 8);
  Circle c = new Circle (5);
  System.out.println ("area of rectangle is " + r.PrintArea ());
  System.out.println ("area of triangle is " + t.PrintArea ());
  System.out.println ("area of circle is " + c.PrintArea ());
 }
}
```

Output-



LAB 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 Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: • Accept deposit from customer and update the balance. • Display the balance. • Compute and deposit interest • Permit withdrawal and update the balance • Check for the minimum balance, impose penalty if necessary and update the balance

Observation-

classmate
Date 11/3/2020
Page

LP#5: Develop a program to create a class Bank that maintains 2 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 facilities. The current account provides checkque book facility but no interest. Current account holders should also maintain a minimum of balance and if the balance falls below this level, a service charge is imposed.
Create a class Account that stores customer name, account number & type of account. From this drive 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 • Accept deposit from customer & update the balance.
• Display the balance • Compute and deposit interest • Permit withdrawal and update the balance • Check for minimum balance, impose penalty if necessary and update the balance

```
import java.util.Scanner
class Account {
String name, type ;
int accno ;
float bal = 0 ; float pen = 100 ;

Account () {
Scanner ss = new Scanner (System.in);
```

```
System.out.println("Enter name, account type number and type
of account:");
name = ss.next();
accno = ss.nextInt();
type = ss.next();
 }
}

class cur_acc extends Account {
float dep, wit;

void deposit() {
Scanner ss = new Scanner(System.in);
 System.out.println("Enter amount to be deposited:");
 dep = ss.nextInt();
 bal = bal + dep;
System.out.println("updated balance after deposit:"+bal);
}


void withdraw() {
Scanner ss = new Scanner(System.in);
System.out.println("enter amount to be withdrawn:");
wit = ss.nextInt();
bal = bal - wit;
System.out.println("Updated balance after withdrawal:"+bal);
```

```
}
void penalty() {
if (bal < 500)
{
bal = bal - pen;
System.out.println ("Updated balance after imposing penalty : "+bal);
}
else
System.out.println ("No penalty imposed, balance : "+bal);
}
}

class sav_acc extends Account {
float dep, wit, r, t, n, ci;

void deposite () {
Scanner ss = new Scanner (System.in);
System.out.println ("enter amount to be deposited.");
dep = ss.nextInt();
System.out.println ("enter rate %, time in years and number
of times interest is compounded per year :");
r = ss.nextInt();
t = ss.nextInt();
n = ss.nextInt();
```

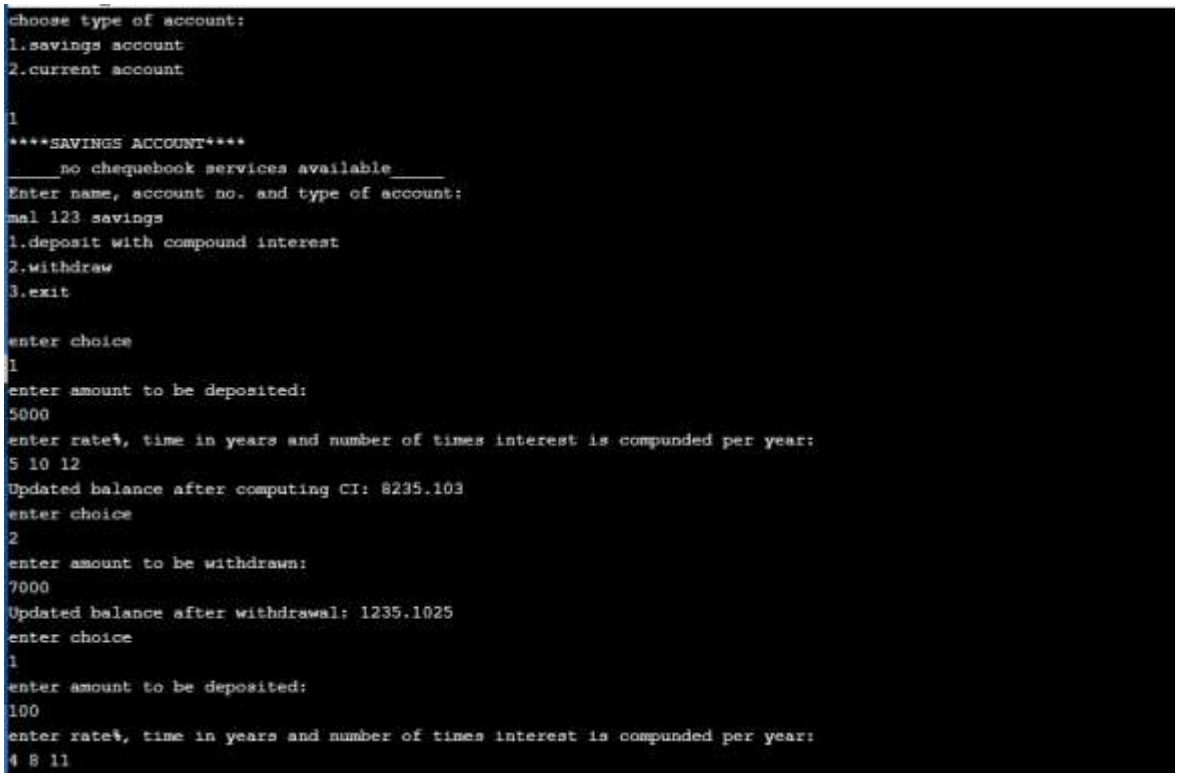
```
ci = dep * ((float) Math.pow ((1+ (r/100 * n))), (n*t)));
bal = bal + ci;
System.out.println ("Updated balance after computing ci: "+bal);
}

void withdraw(){
Scanner ss = new Scanner (System.in);
System.out.println ("enter amount to be withdrawon: ");
wit = ss.nextInt();
bal = bal - wit;
System.out.println ("Updated balance after withdrawal: " +bal);
}
}

class Bank {
public static void main (String sss[]) {
Scanner ss = new Scanner (System.in);
int opt; int ch;
System.out.println ("choose type of account: ");
System.out.println ("1. savings account \n 2. current account \n");
opt = ss.nextInt();
if (opt == 1) {
System.out.println ("**** SAVINGS ACCOUNT ****");
System.out.println ("____ no chequebook facilities available");
sav_acct s = new sav_acct();
```

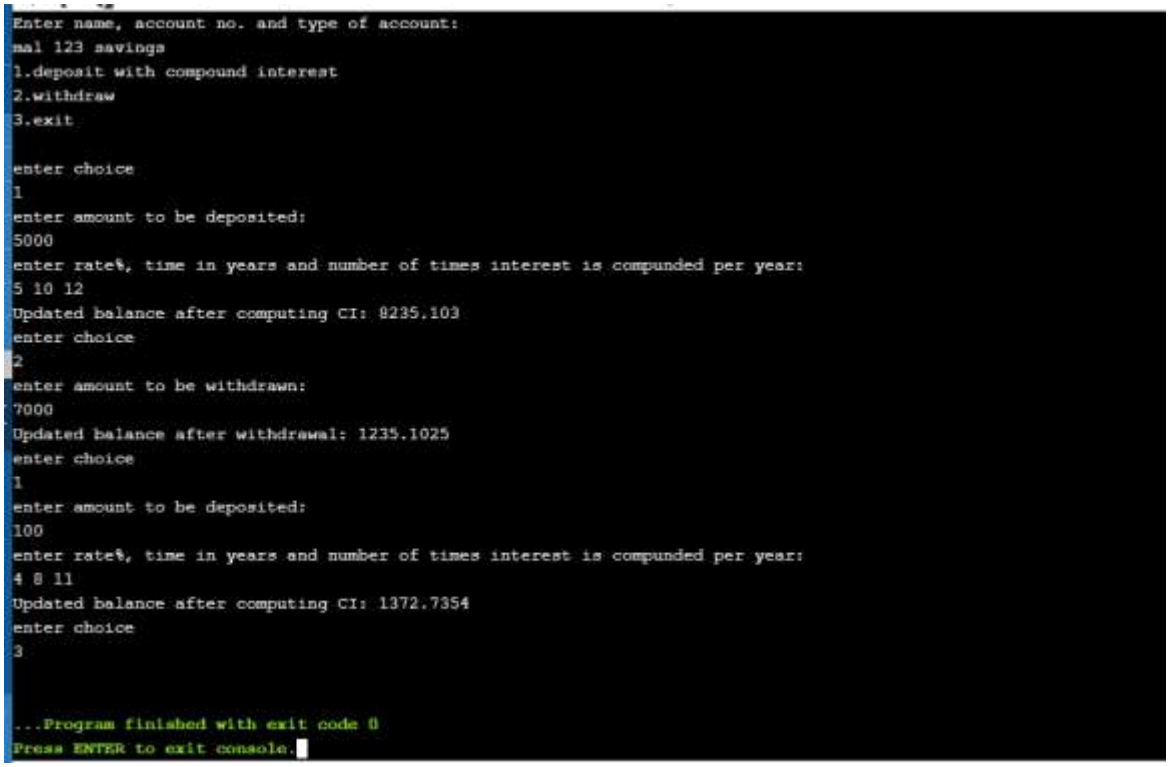
```
System.out.println("1. deposit with compound interest \n 2. with-
draw \n 3. exit \n");
do
{
System.out.println("enter choice ");
ch = ss.nextInt();
switch(ch)
{
case 1 : s.depositc(); break
case 2 : s.withdrawc(); break;
case 3: break;
}
} while(ch != 3);
}
if(opt==2){
System.out.println("**** CURRENT ACCOUNT ****");
System.out.println("___ chequebook service available ___");
cum_acct c = new cum_acct();
System.out.println("1. deposit \n 2. withdraw \n 3. check minim-
um balance / penalty \n 4. exit \n");
do
{
System.out.println("enter choice ");
ch = ss.nextInt();
```

```
switch (ch)
{
case 1 : c.deposit (); break;
case 2 : c.withdraw (); break;
case 3 : c.penalty (); break;
}
} (while ch! =4 );
}
}
}
```

Output-

Savings:





Current:

```
choose type of account:
1.savings account
2.current account

2
****CURRENT ACCOUNT****
_____chequebook services available_____
Enter name, account no. and type of account:
mal 1234 current
1.deposit
2.withdraw
3.check minimum balance/penalty
4.exit

enter choice
1
Enter amount to be deposited:
2000
updated balance after deposit: 2000.0
enter choice
2
enter amount to be withdrawn:
1400
Updated balance after withdrawal: 600.0
enter choice
3
No penalty imposed, balance:600.0
enter choice
2
enter amount to be withdrawn:
200
```

```
Updated balance after withdrawal: 600.0
enter choice
3
No penalty imposed, balance:600.0
enter choice
2
enter amount to be withdrawn:
200
Updated balance after withdrawal: 400.0
enter choice
3
Updated balace after imposing penalty: 300.0
enter choice
1
Enter amount to be deposited:
500
updated balance after deposit: 800.0
enter choice
2
enter amount to be withdrawn:
100
Updated balance after withdrawal: 700.0
enter choice
3
No penalty imposed, balance:700.0
enter choice
4


...Program finished with exit code 0
Press ENTER to exit console.
```