LAB RECORD

NAME :Nikhil S N

USN: 1BM19CS102

SEM : 3

SECTION : B

SUBJECT : Object Oriented Java Programming(OOJ)

TEST :1 and 2

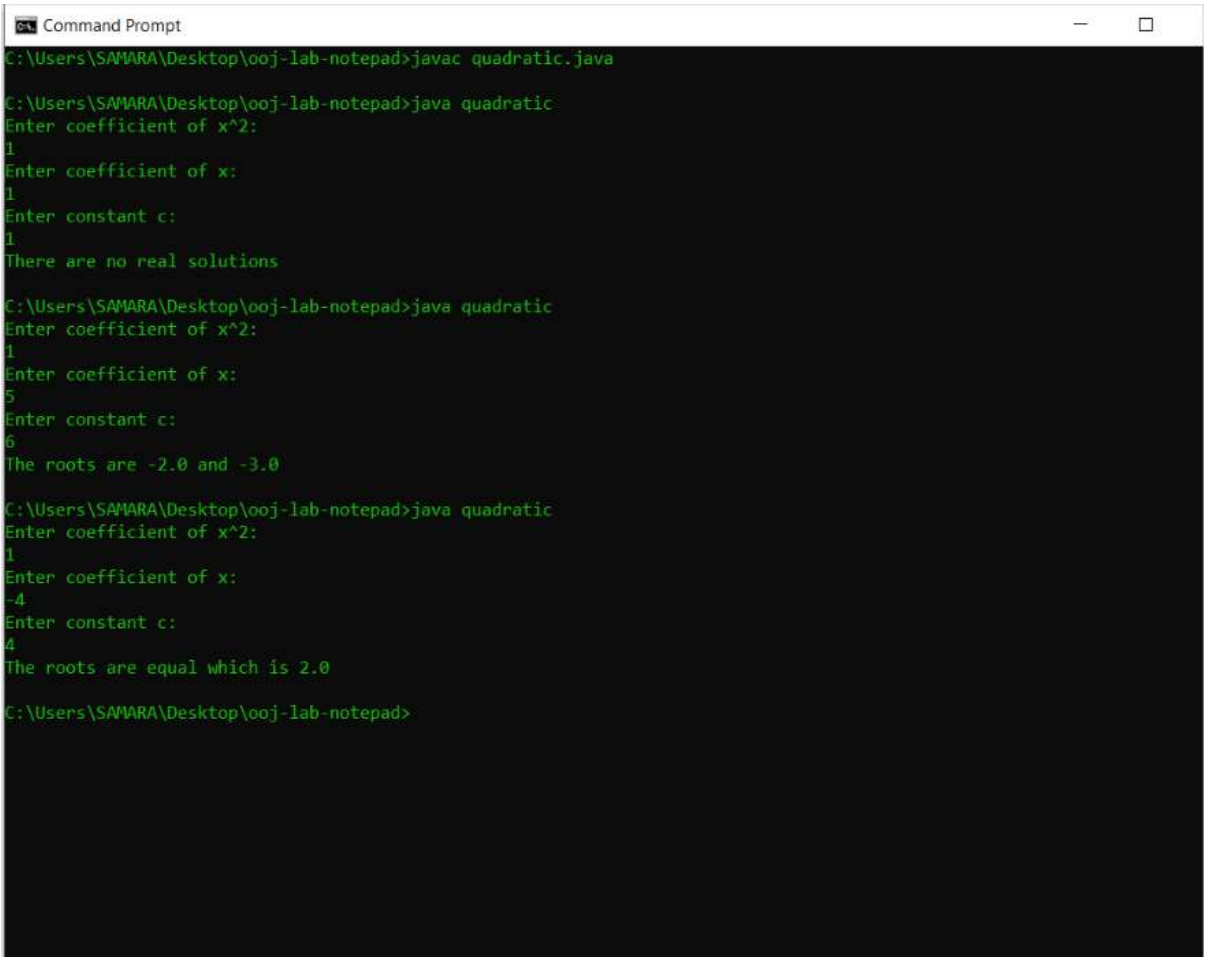
LAB 1:

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 discriminate b2-4ac is negative, display a message stating that there are no real solutions.

Lab-1

```
import java.util.*;
Public class quadratic {
public static void main (String args[]){
Scanner in = new, Scanner(System.in);
double ans, disc, real, root1=0, root2=0;
System.out.println("Enter coefficient of x2:");
double a = in.nextDouble();
System.out.println("Enter coefficient of x:");
double b = in.nextDouble();
System.out.println("Enter constant c:");
double c = in.nextDouble();
disc = (b*b) - (4.0*a*c);
if (disc < 0.0) {
   System.out.println("There are no real
                       solutions");
}
else if (disc == 0.0) {
   real = -b/(2*a);
   System.out.println("the roots are
   equal which is" + real);
}
else
```

```
{
root1 = ((-b) + sqrt) / (2.0*a);
root2 = ((-b) + sqrt)/ (2.0*a);
System.out.println("the root1 are " + root1+" 
                    and " +root2);
    }
  }
}
```



LAB 2:

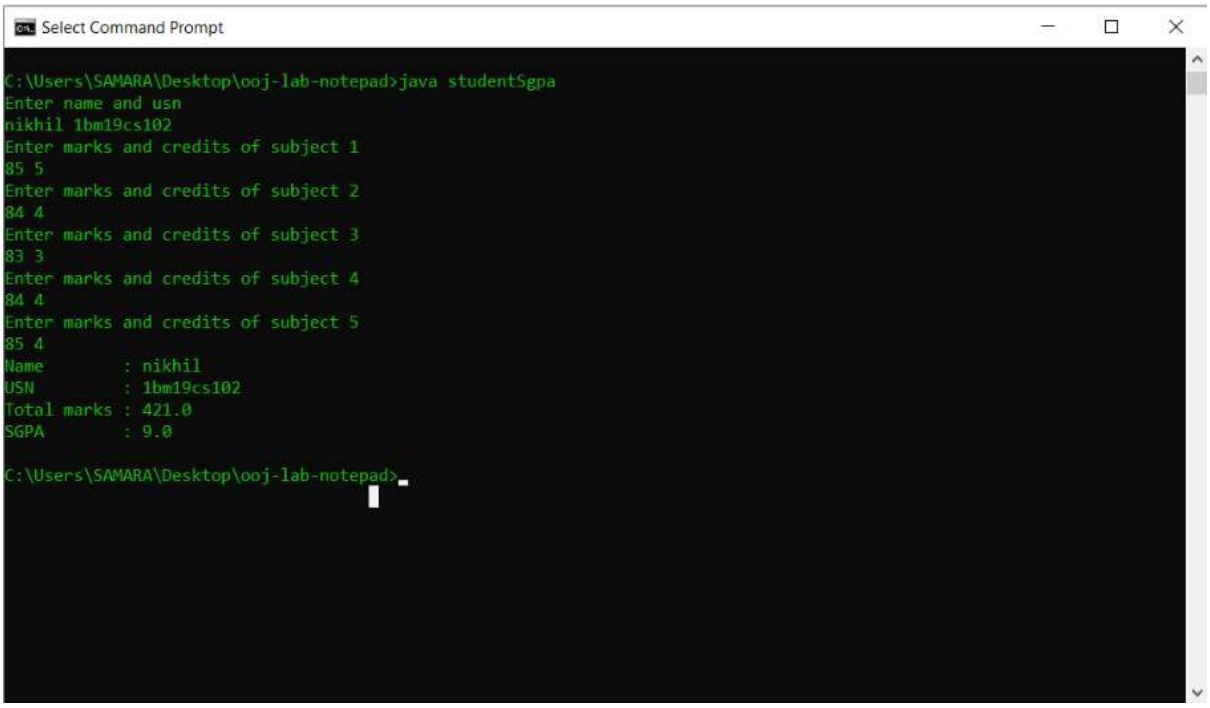
Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

Lab : 2

```
import java.util.*;
class student{
String name, usn;
double credits[] = new double[5];
double marks[] = new double[5];
double a[] = new double[5];

double total=0, tc=0, sgpa=0, tmc=0;
void getdetails(){
  Scanner in = new Scanner(System.in);
  System.out.println("Enter name and usn");
  name = in.next();
  usn = in.next();
  for(int i=0; i<5; i++){
  System.out.println("Enter marks and credits
                      of subject "+(i+1));
  marks[i] = in.nextDouble();
  credits[i] = in.nextDouble();

  }
}

void displaydetails(){

System.out.println("Name : "+name);
System.out.println("USN : "+usn);
for(int j=0; j<5; j++){
  total = marks[j];
}
```

```
System.out.println("Total marks : " +total);
}
void SgPA(){
for(int j=0;j<5;j++){
  if(marks[j]>=90)
    a[j] = 10.0;
  else if(marks[j]>=80 && marks[j]<90)
    a[j] = 9.0;
  else if(marks[j]>=70 && marks[j]<80)
    a[j] = 8.0;
  else if(marks[j]>=60 && marks[j]<70)
    a[j] = 7.0;
  else if(marks[j]>=50 && marks[j]<60)
    a[j] = 6.0;
  else
    a[j] = 0.0;
}
for(int n=0; n<5; n++){
  tc += credits[n];
  tmc += (credits[n]*a[n]);
}
sgpa = (tmc/tc);
System.out.println("SG PA : " +sgpa);
}
}
```

```
public class StudentSgpa {
    public static void main (String[] args)
    {
    S1. getdetails ();
    S1. displaydetails ();
    S1. SgpA ();
    }
}
```



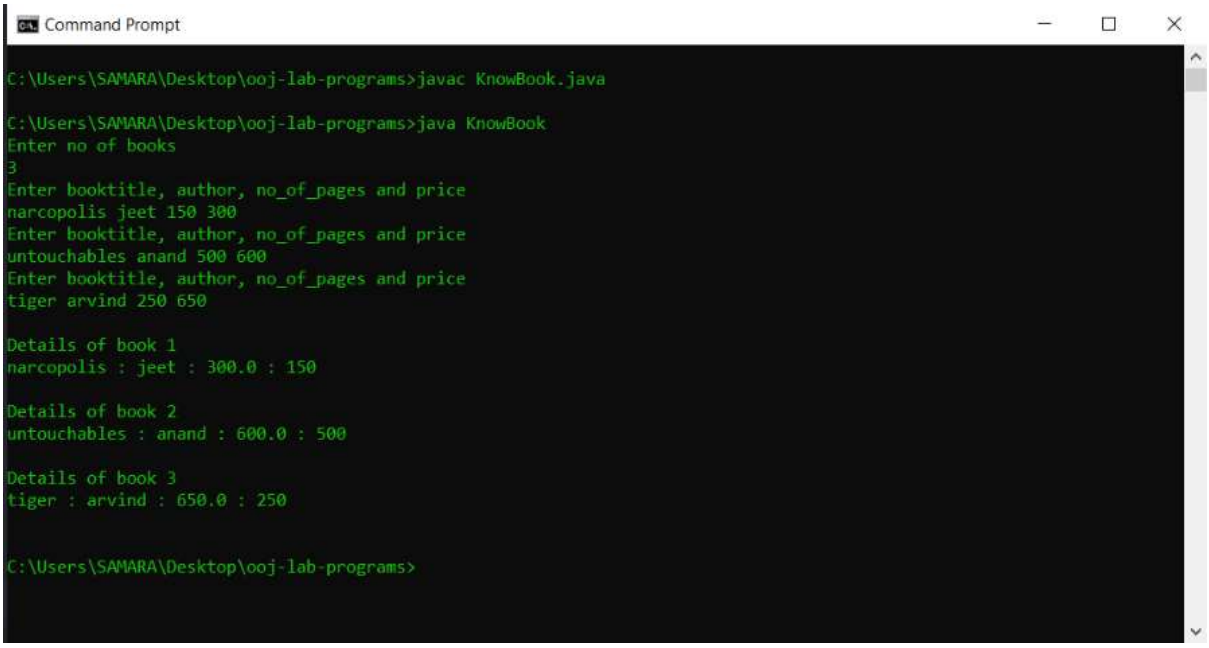
LAB 3:

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.

Lab 3:

```
import java.util.*;
class book{
String booktitle, author;
double price;
int no_of_pages;
public book(){
author = "unknown";
price = 300;
no_of_pages = 150;
}
book(String booktitle, String author, double
price, int no_of_pages){
this.booktitle = booktitle
this.author = author;
this.price = price;
this.no_of_pages = no_of_pages;
}
public String toString(){
return(booktitle+":"+author+":"+Price+":"
+no_of_pages);
}
void GetDetails(){
Scanner in = new Scanner(System.in);
System.out.println("Enter booktitle, author,
no_of_pages and price");
```

```
booktitle = in.next();
author = in.next();
no_of_pages = in.nextInt();
price = in.nextDouble();
}
void setDetails() {
System.out.println("Book Title :" + booktitle);
System.out.println("Author :" + author);
System.out.println("No of pages :" + no_of_pages);
System.out.println("price :" + price);
}
}
class knowBook {
public static void main(String[] args){
Scanner in = new Scanner(System.in);
int n;
System.out.println("Enter no of books");
n = in.nextInt();
book[] b = new book[n];
for (int i=0; i<n; i++) {
b[i] = new book();
b[i].getDetails();
}
System.out.println("Details of book" + (i+1));
System.out.println(b[i]); } }
```


Command Prompt
C:\Users\SAMARA\Desktop\ooj-lab-programs>javac KnowBook.java
C:\Users\SAMARA\Desktop\ooj-lab-programs>java KnowBook
Enter no of books
3
Enter booktitle, author, no_of_pages and price
narcopolis jeet 150 300
Enter booktitle, author, no_of_pages and price
untouchables anand 500 600
Enter booktitle, author, no_of_pages and price
tiger arvind 250 650
Details of book 1
narcopolis : jeet : 300.0 : 150
Details of book 2
untouchables : anand : 600.0 : 500
Details of book 3
tiger : arvind : 650.0 : 250
C:\Users\SAMARA\Desktop\ooj-lab-programs>

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.

Lab 11

```
abstract class Shape {
int num1, num2;
void printArea(int a, int b){
    num1 = a;
    num2 = b;
}
}
class Rectangle extends Shape {
void printArea(int num1, int num2){
System.out.println("Area of rectangle
                    = " + (num1 * num2));
}
}
class Triangle extends Shape {
    void printArea(int num1, int num2){
System.out.println("Area of Triangle =
                    " + ((double)(num1 * num2)
                    / 2));
}
}
class circle extends Shape {
    void printArea(int num1){
System.out.println("Area of circle = " +
                    (3.14 * num1 * num1));
}
}
```

```
class Area {
public static void main(String[] args){
Rectangle r = new Rectangle();
r.printArea(7,7);
Triangle t = new Triangle();
t.printArea(7,7);
Circle c = new Circle();
c.printArea(7);
}
}
```

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

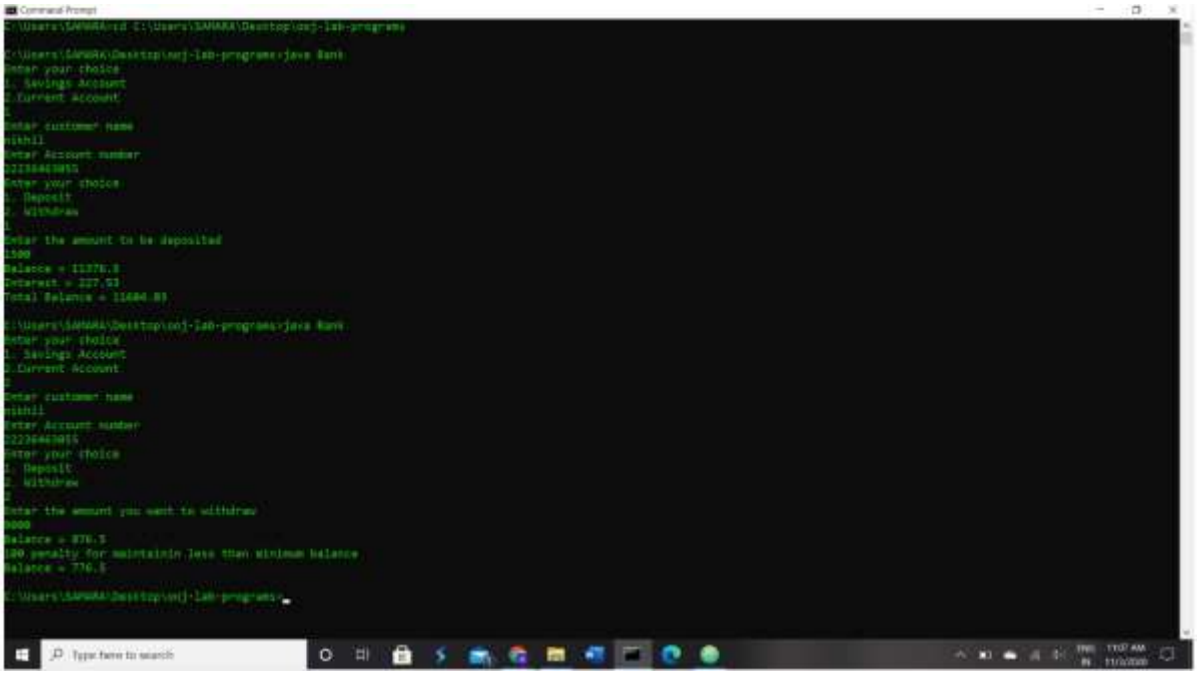
Lab 5:

```
import java.util.*;
class Account {
Scanner in = new Scanner (System.in);
String Customer_name, type_of_account;
long account_number;
double balance = 98765;
void Accept () {
System.out.println ("Enter Customer name");
Customer_name = in.nextLine ();
System.out.println ("Enter Account number");
account_number = in.nextLong ();
}
void deposit () {
int dep;
System.out.println ("Enter the amount to be
                    deposited");
dep = in.nextInt ();
balance += dep;
System.out.println ("Balance =" + balance);
}
void withdrawal () {
int withdra
System.out.println ("Enter the amount
    you want to withdraw");
withdra = in.nextInt ();
```

```
balance -= withdra';
System.out.println("Balance :" + balance);
}
}

class CurrAct extends Account {
  void penalty(){
  if (balance < 100) {
    balance -= 100;
  System.out.println("100 Penalty for maintain
  less than minimum balance");
  System.out.println("Balance =" + balance);
  }
}

class SavAct extends Account {
  void interest(){
    double i;
    i = balance * 0.02;
    balance += i;
  System.out.println("Interest =" + i);
  System.out.println("Total Balance =" +
                     balance);
  }
}
```

```
class Bank {
public static void main (String[] args)
{
Scanner in = new Scanner (System.in);
CurrAct c = new CurrAct ();
SavAct s = new SavAct ();
System.out.println ("Enter your choice\n
1. Savings Account \n 2. Current Account");
int choice = in.nextInt();
if (choice == 2)
{
c.Accept ();
System.out.println ("Enter your choice\n
1. Deposit \n 2. Withdraw");
int n = in.nextInt();
switch (n) {
case 1: {
c.deposit ();
break;
}
case 2: {
c.withdraw ();
c.finalbal ();
break;
}
```

```
default: System.out.println("Wrong
                    choice!");
    }
}
if (choice==1){
 S.accept();
 System.out.println("Enter your choice
 In 1:Deposit In 2:Withdraw");
 int n=in.nextInt();
 switch(n){
 case 1: {
  S.deposit();
  S.interest();
  break;
 }
 case 2: {
 S.withdrawal();
  break;
 }
default: System.out.println("Wrong
                    choice!");
```



Lab 6

: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.

Lab:- 6

Student class

```
package CIE;
import java.util.*;
public class Student {
Scanner in = new Scanner(System.in);
public String name, usn;
public int sem;
public void Accept() {
System.out.println("Enter name and usn of a
student");
name = in.nextLine();
usn = in.next();
System.out.println("Enter sem of student");
sem = in.nextInt();
}
}
```

Internals class

```
package CIE;
public class Internals extends
Student {
public int marks[] = new int[5];
public void Accept1() {
```

```
for (int i=0; i<5; i++){
System.out.println("Enter cie marks of subject
                    + (i+1));
 cie mar[i] = in.nextInt();
 }
 }
}
```

external class

```
package EE;
import CIE.*;
import java.util.*;
public class externals extends CIE.student {
Scanner in = new Scanner (System.in);
public int semar[]= new int[5];
public void acceptSee(){
for (int i=0; i<5; i++){
System.out.println("Enter EE marks of
subject " + (i+1));
see mar[i] = in.nextInt();
 }
}
}
```

Main class

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

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

{
Scanner in = new Scanner (System.in);
int n, tot=0;
System.out.println ("Enter number of Students");
n = in.nextInt ();
CIE.Internals[] I = new CIE.Internals[n];
SEE.Externals[] E = new SEE.Externals[n];
for (int i=0; i<n; i++) {
I[i] = new CIE.Internals ();
E[i] = new SEE.Externals ();
I[i].Accept();
I[i].Accept();
E[i].Accept();
}
for (int i=0; i<n; i++) {
System.out.println ("Student" + (i+1));
System.out.println ("Name: " + I[i].name +
" USN: " + I[i].usn + " SEM: " + I[i].sem);
```

```
System.out.println("Total marks");
for (int j=0; i<5; j++){
tot = I[i].cmarks[i]+(E[i].emarks[i]/2);
System.out.println("Subject"+(j+1)+" = "+tot);
    }
  }
}
}.
```

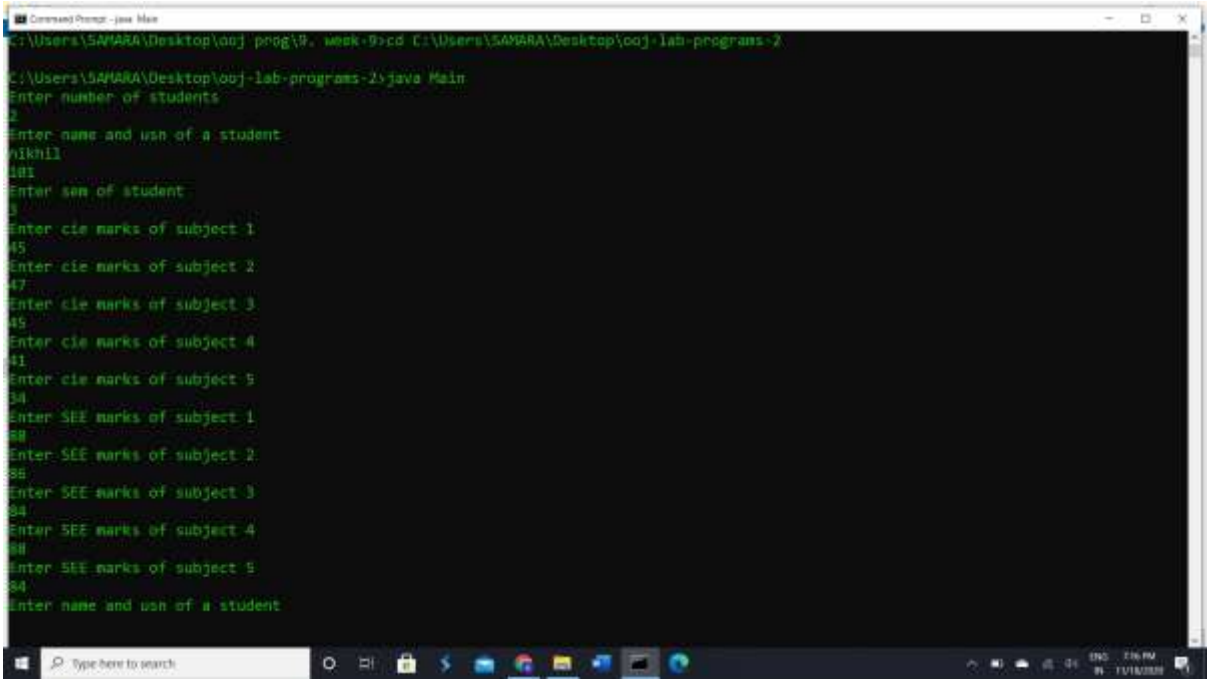
Procedure

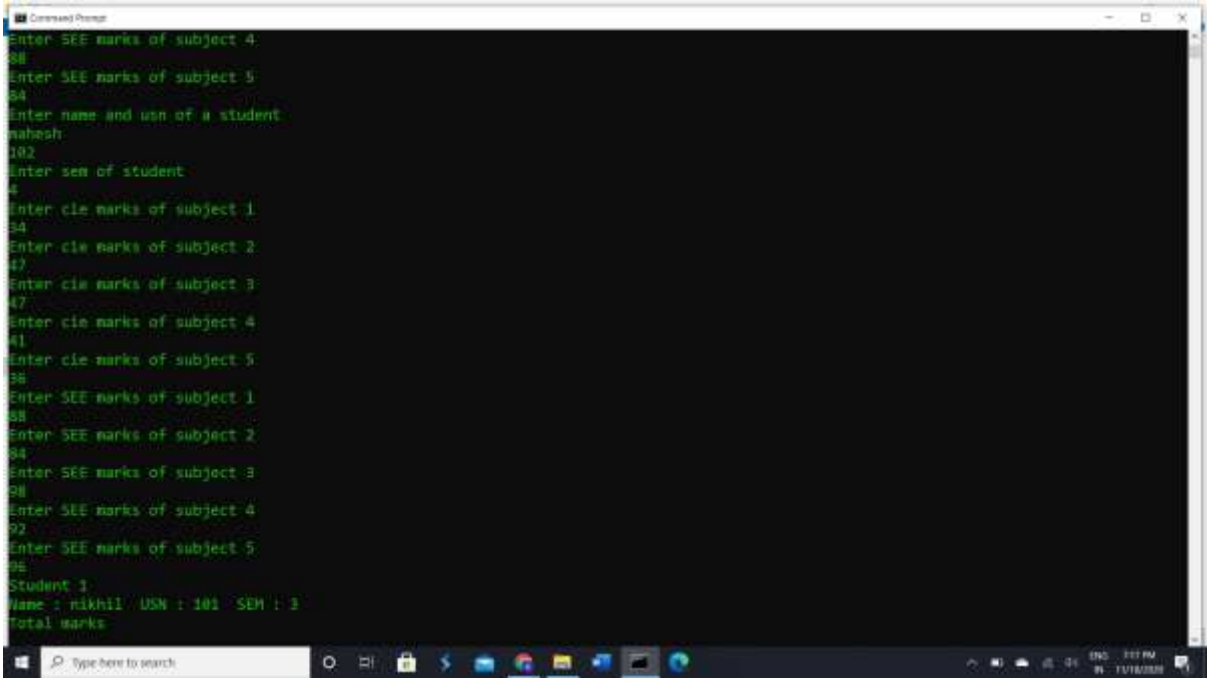
(i) Execute student file and store class file in CIE folder.

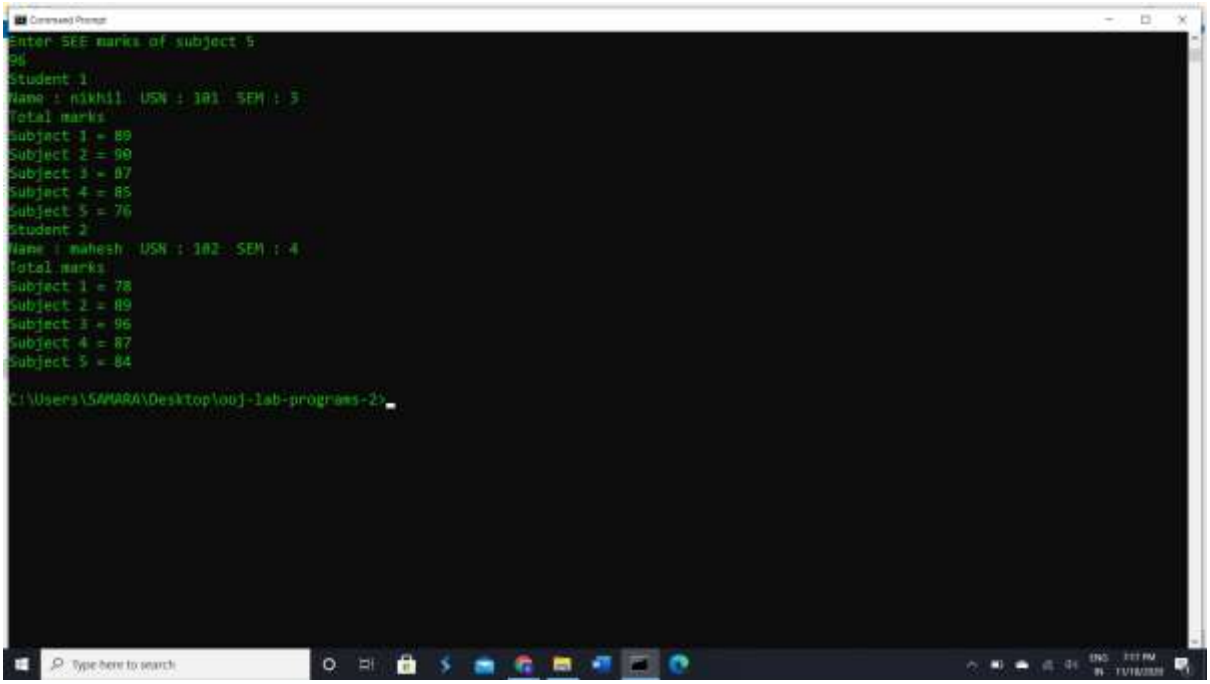
(ii) execute Internals file and store internals class file in CIE folder.

(iii) execute externals file and store externals class file in SEE folder.

(iv) Execute main java file.


C:\Users\SAMARA\Desktop\ooj prog\9. week-9>cd C:\Users\SAMARA\Desktop\ooj-lab-programs-2
C:\Users\SAMARA\Desktop\ooj-lab-programs-2>java Main
Enter number of students
2
Enter name and usn of a student
nikhil
101
Enter sem of student
3
Enter cie marks of subject 1
45
Enter cie marks of subject 2
47
Enter cie marks of subject 3
45
Enter cie marks of subject 4
41
Enter cie marks of subject 5
38
Enter SEE marks of subject 1
88
Enter SEE marks of subject 2
86
Enter SEE marks of subject 3
84
Enter SEE marks of subject 4
88
Enter SEE marks of subject 5
84
Enter name and usn of a student


Enter SEE marks of subject 4
88
Enter SEE marks of subject 5
84
Enter name and usn of a student
vahesh
102
Enter sem of student
4
Enter cie marks of subject 1
34
Enter cie marks of subject 2
47
Enter cie marks of subject 3
47
Enter cie marks of subject 4
41
Enter cie marks of subject 5
38
Enter SEE marks of subject 1
88
Enter SEE marks of subject 2
84
Enter SEE marks of subject 3
98
Enter SEE marks of subject 4
92
Enter SEE marks of subject 5
96
Student 1
Name : nikhil USN : 101 SEM : 3
Total marks

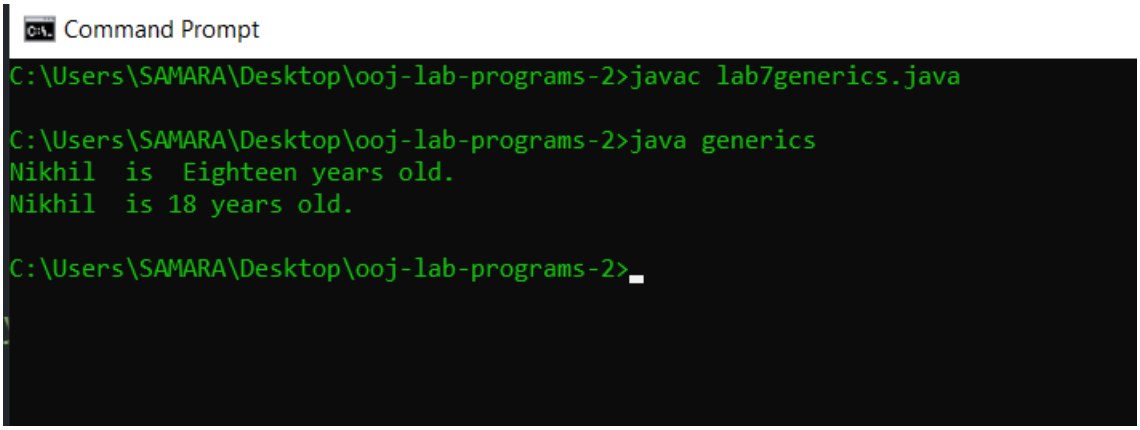


lab7.

Write a program to demonstrate generics with multiple object parameters.

Lab-7

```
class container<T1, T2> {
    T1 a;
    T2 b;
void display(T1 a, T2 b){
    this.a = a;
    this.b = b;
System.out.println(this.a + " is " + this.b + "
                    years old.");

    }
}
class generics {
public static void main(String[] args) {
container<String, String> c1 = new container();
Container<String, Integer> c2 = new Container();
c1.display("Nikhil", "Eighteen");
c2.display("Nikhil", 18);

    }
}
```



lab8.

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

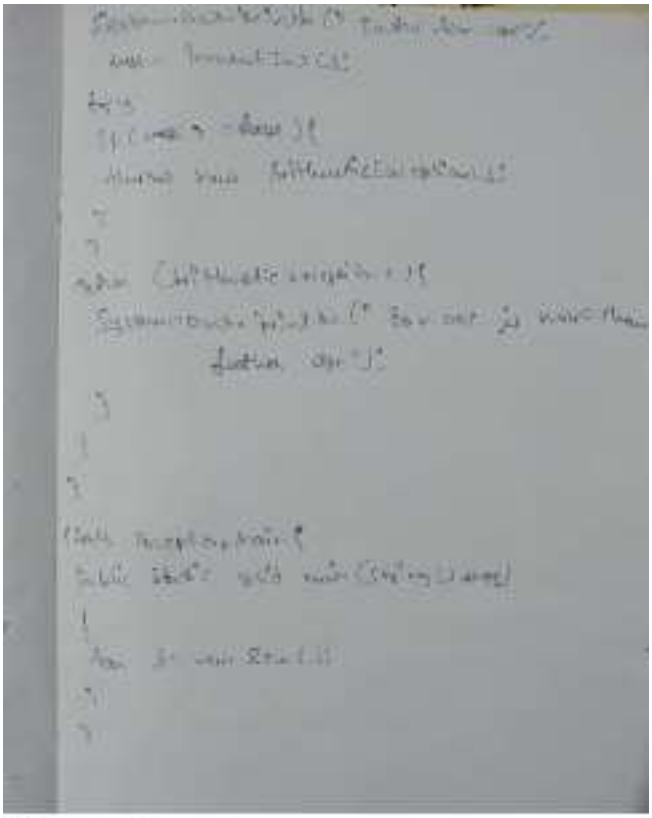
called “Father” and derived class called “Son” which extends the base class. In Father class,

implement a constructor which takes the age and throws the exception Wrong Age( ) when the input

age=father’s age.

Lab -8

```
import java.util.*;
class father {
Scanner in = new Scanner(System.in);
int fage;
father() throws ArithmeticException {
System.out.println("Enter fathers age");
fage = in.nextInt();

try {
if (fage < 0) {
throw new ArithmeticException();

}
}
catch (ArithmeticException e) {
System.out.println("age of the father is
        less than 0");

}
}
}
class son extends father {
int sage;
son() {
```



```
Command Prompt
Microsoft Windows [Version 10.0.17763.615]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Nitish kumar M>cd documents

C:\Users\Nitish kumar M\Documents>cd java

C:\Users\Nitish kumar M\Documents\java>javac expmain.java

C:\Users\Nitish kumar M\Documents\java>java expmain
ENTER FATHER'S AGE
23
ENTER SON'S AGE
45
AGE OF SON=45 IS ENTERED INCORRECTLY

C:\Users\Nitish kumar M\Documents\java>java expmain
ENTER FATHER'S AGE
24
ENTER SON'S AGE
12
THE AGES ARE ENTERED CORECTLY
FATHER'S AGE=24 SON'S AGE=12

C:\Users\Nitish kumar M\Documents\java>
```

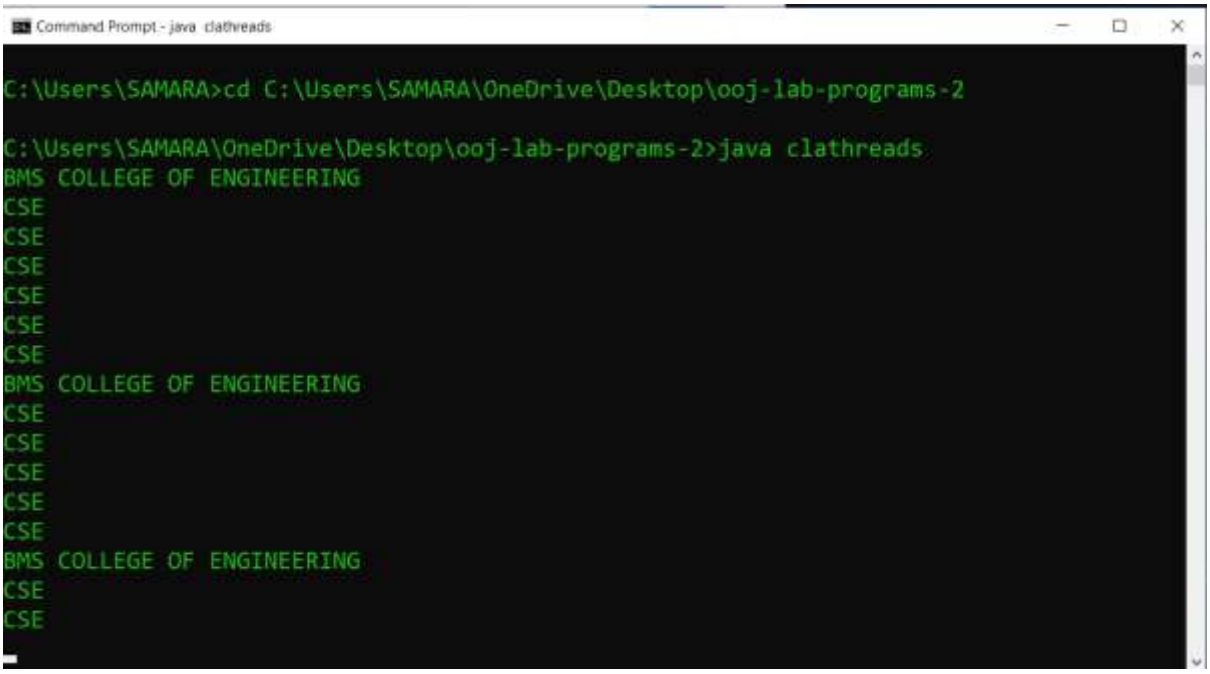
Lab 9

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.

Lab 9:

```
class cse extends Thread {
public void run(){
while (true) {
System.out.println("CSE");
try {
Thread.sleep(2000);
}catch (Exception e) {}

}
}
}
class clathread {
public static void main(String[] args) {
cse c= new cse();
c.start();
while (true) {
System.out.println("RMS COLLEGE OF
ENGINEERING");

try {
Thread.sleep(10000);
}catch (Exception e) {}

}
}
}
```



Lab 10:

Write a program that creates a user interface to perform integer divisions. The user enters two

numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the

Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program

would throw a NumberFormatException. If Num2 were Zero, the program would throw an

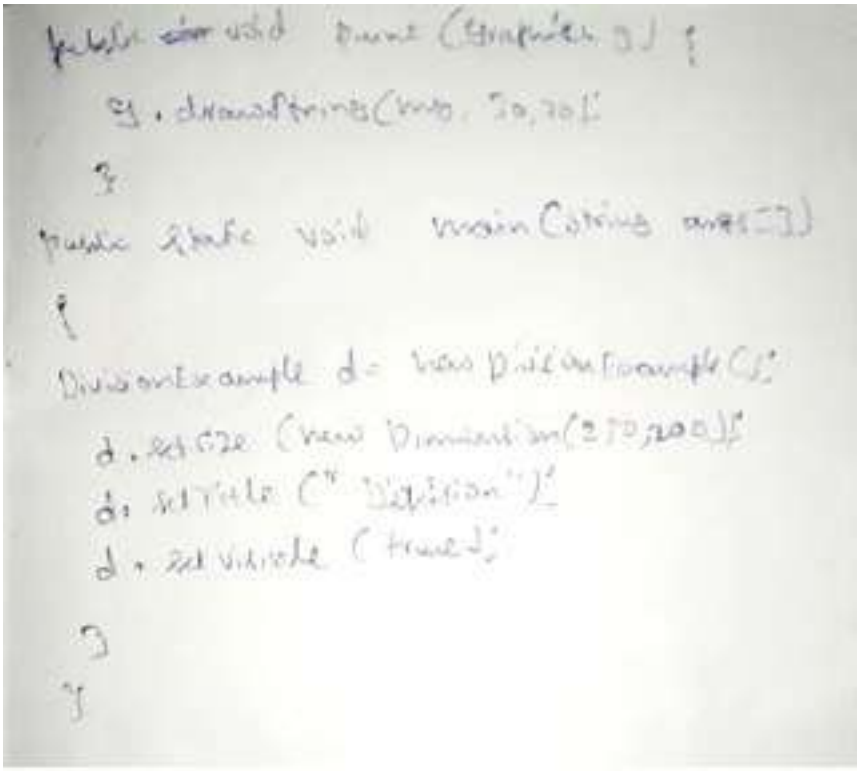
Arithmetic Exception Display the exception in a message dialog box.

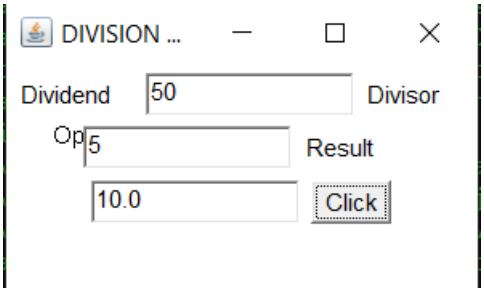
Lab-10

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import
import javax.swing.*;

Public class DivisionExample extends Frame
        implements ActionListener{

String msg;
TextField num1, num2, res;
Label l1, l2, l3;
Button div;

Public DivisionExample(){
setLayout(new FlowLayout());
l1 = new Label("number 1");
l2 = new Label("number2");
l3 = new Label("Result");
num1 = new TextField(10);
num2 = new TextField(10);
res  = new TextField(10);
div  = new Button("divide");
add(l1);
add(num1);
```

```
add(l2);
add(num2);
add(l3);
add(sub);
add(div);
div.addActionListener(this);
addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent we){
System.exit(0);
}
});
}
public void actionPerformed(ActionEvent ae){
String arg = ae.getActionCommand();
int num1=0, num2=0;
if(arg.equals("divide"))
if(this.num1.getText().isEmpty() || this.
num2.getText().isEmpty()){
msg = "Enter the valid numbers!";
repaint();
}
else{
```

```
num1 = Integer.parseInt(this.num1.getText());
num2 = Integer.parseInt(this.num2.getText());

try {
if (num2 == 0)
{
int a = n/0;
}
int num3 = num1 / num2;
res.setText(String.valueOf(num3));
msg = "Operation Done";
repaint();
}
catch (ArithmeticException e){
JOptionPane.showMessageDialog(null,"
ArithmeticException  Division by zero not
defined");
res.setText("");
msg = "cant't be divided by zero";
repaint();
}

}
}
}
```




DIVISION ...
Dividend
50
Divisor
Op
5
Result
10.0
Click


Message
OK


Can't be divided by Zero
number1
123
number2
0
Result
divide


Message
OK

*******THE END*******