

OOJ LAB RECORD

Name Amit Ramachandran

USN 1BM19CS016

LAB PROGRAM – 1 :

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.

Lab Program 1 :
PAGE: / /
DATE: / /

```
// Quadratic Equations
import java.util.Scanner;
class Equation
{
    public static void main (String args[])
    {
        int a,b,c;
        double d;
        double x1=0.0,x2=0.0;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter a,b & c of
Q.E : ax2+bx+c=0");
        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();
        if (d = b*b - 4*a*c > 0)
        {
            x1 = (-b + Math.sqrt(d))/(2*a);
            x2 = (-b - Math.sqrt(d))/(2*a);
            if (d > 0)
                System.out.println ("real & distinct ");
            System.out.printf ("solution : %.2f", x1);
            System.out.printf ("%.2f", x2);
        }
        else if (d == 0)
        {
            x1 = x2 = -b/(2*a);
            System.out.println ("real & equal ");
            System.out.printf ("solution : %.2f", x1);
            System.out.printf ("%.2f", x2);
        }
        else if (d < 0)
            System.out.println ("no real solutions ");
    }
}
```

Command Prompt

Microsoft Windows [Version 10.0.18362.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Amit R>javac assignment1.java

C:\Users\Amit R>java Equation
Enter a, b and c of QE: ax²+bx+c=0
6
11
-35
real and distinct
solution:1.67 -3.50
C:\Users\Amit R>java Equation
Enter a, b and c of QE: ax²+bx+c=0
1
1
2
no real solutions

C:\Users\Amit R>java Equation
Enter a, b and c of QE: ax²+bx+c=0
2
4
2
real and equal
Solution:-1.00 -1.00
C:\Users\Amit R>

LAB PROGRAM – 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

PAGE: / /
DATE: / /

Lab Program 2 :-

```
import java.util.Scanner;
class Student
{
    String usn;
    String name;
    int n;
    int credits [] = new int [100];
    int marks [] = new int [100];
    double sgpa = 0.0;
    double q [] = new double [100];
    void accept()
    {
        System.out.println ("Enter no. of subjects");
        n = sc.nextInt();
        credits = new int [n];
        marks = new int [n];
        System.out.println ("Enter usn & name of student");
        usn = sc.next();
        name = sc.next();
        System.out.println ("Enter credits");
        for (int i=0; i<n; i++)
            credits [i] = sc.nextInt();
        System.out.println ("Enter marks");
        for (int i=0; i<n; i++)
            marks [i] = sc.nextInt();
    }
    void display()
    {
        System.out.println ("usn : " + usn);
        System.out.println ("name : " + name);
        System.out.println ("credits : ");
        for (int i=0; i<n; i++)
            System.out.println (credits [i]);
    }
}
```

```
for (int i=0; i<n; i++)
    cout << marks[i];
```

```
3 double gsum = 0.0;
void calcsgpa()
```

```
{ for (int i=0; i<n; i++)
```

```
    if (marks[i] > 90)
```

```
        g[i] = 10.0;
    else if (marks[i] >= 75 && marks[i] <= 89)
        g[i] = 9.0;
```

```
    else if (marks[i] > 60 && marks[i] <= 74)
        g[i] = 8.0;
```

```
    else if (marks[i] > 50 && marks[i] <= 59)
        g[i] = 7.0;
```

```
    else if (marks[i] > 40 && marks[i] <= 49)
        g[i] = 6.0;
```

```
    else
        g[i] = 0.0;
```

```
    for (int i=0; i<n; i++)
```

```
        gsum = gsum + g[i];
```

```
    for (int i=0; i<n; i++)
        sgpa = sgpa + (g[i] * credits[i]) /
```

```
System.out.println ("sgpa : %.2f", sgpa);
```

```
3 class StudentM
```

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

```
Student s = new Student();  
s. accept();  
s. display();  
s. calcsgpa();  
}  
}
```

```
C:\ Command Prompt  
C:\Users\Amit R>javac labprog3.java  
C:\Users\Amit R>java BookObj  
Enter no. of books  
2  
Enter name of book and author  
harry potter  
Enter price and no. of pages  
400  
50  
Enter name of book and author  
rr  
abc  
Enter price and no. of pages  
45  
78  
Name:harry Author:potter Price:400 No. of pages:50  
Name:rr Author:abc Price:45 No. of pages:78  
C:\Users\Amit R>
```

Lab Program – 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.

LabProgram 3 :-

import java.util.Scanner;
class Book

```
String name, author;
int price;
int numPages;
Book()
{
    name = " ";
    author = " ";
    price = 0;
    numPages = 0;
```

public void getDetails()

```
{  
Scanner sc = new Scanner (System.in);  
System.out.println ("Enter name of book & ");  
name = sc.next();  
System.out.println ("");  
author = sc.next();  
System.out.println ("Enter price and no. of pages");  
price = sc.nextInt();  
numPages = sc.nextInt();  
}
```

public String toString()

{
return ("Name: " + name + "Author: " + author
+ "Price: " + price + "No. of
pages: " + numPages);

3
class BookObj

{
public static void main (String args[])

{
int n;

Scanner sc = new Scanner (System.in);
System.out.println ("Enter no. of books");

n = sc.nextInt();

Book b[] = new Book [n];

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

b[i] = new Book();

b[i].getDetails();

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

System.out.println (b[i]);

3

```
Command Prompt  
C:\Users\Amit R>javac labprog3.java  
C:\Users\Amit R>java BookObj  
Enter no. of books  
2  
Enter name of book and author  
harry potter  
Enter price and no. of pages  
400  
50  
Enter name of book and author  
rr  
abc  
Enter price and no. of pages  
45  
78  
Name:harry Author:potter Price:400 No. of pages:50  
Name:rr Author:abc Price:45 No. of pages:78  
C:\Users\Amit R>
```

Lab Program – 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 PROGRAM 4 :—  
abstract class Shape  
{  
    int a = 5, b = 4;  
    abstract void printArea();  
}  
  
class Rectangle extends Shape  
{  
    void printArea()  
    {  
        System.out.println("area of rectangle :" + (a*b));  
    }  
}  
  
class Triangle extends Shape  
{  
    void printArea()  
    {  
        System.out.println("area of triangle :" + (a*b/2));  
    }  
}  
  
class Circle extends Shape  
{  
    void printArea()  
    {  
        System.out.println("area of circle :" + (3.14*a*a));  
    }  
}  
  
class labprog4  
{  
    public static void main (String args [ ])  
    {  
        Rectangle r = new Rectangle();  
    }  
}
```

```
    r.printArea();  
Triangle t = new Triangle();  
t.printArea();  
Circle c = new Circle();  
c.printArea();  
3
```

```
C:\Users\Amit R>javac labprog4.java  
C:\Users\Amit R>java labprog4  
area of rectangle:20  
area of triangle:10  
area of circle:78.5  
C:\Users\Amit R>
```

Lab Program – 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 PROGRAM 5 :-

```
import java.util.Scanner;
```

```
abstract class Account
```

```
{
```

```
String name;
```

```
int accno;
```

```
String type;
```

```
Scanner sc = new Scanner(System.in);
```

```
3
```

```
class Curr-acct extends Account
```

```
{
```

```
double amt = 0.0;
```

```
int d;
```

```
void accept()
```

```
{
```

```
System.out.print("Enter name & deposit amt");
```

```
name = sc.nextLine();
```

```
d = sc.nextInt();
```

```
amt = amt + d;
```

```
3
```

```
void display()
```

```
{
```

```
System.out.println("Name: " + name);
```

```
System.out.println("Current Balance: " + amt);
```

```
3
```

```
void checkmin()
```

```
{  
    if (amt < 200.0)
```

```
        System.out.println ("Below minimal balance,  
                           will be charged penalty");
```

```
        amt = amt - (amt * 16.0 / 100);
```

```
    System.out.println ("Current balance ="  
                       + amt);
```

```
}
```

```
}
```

```
class Sav acct extends Account
```

```
double amt = 0.0;
```

```
int d, w, b, t;
```

```
double ci = 0.0;
```

```
void accept()
```

```
System.out.println ("Enter name & deposit amt");
```

```
name = sc.nextLine();
```

```
d = sc.nextInt();
```

```
amt = amt + d;
```

```
}
```

```
void display()
```

```
System.out.println ("Name : " + name);
```

```
System.out.println ("Current balance = " + amt);
```

```
}
```

```
void interest()
```

```
{
```

```
System.out.println ("Enter principal amt");
```

```
p = sc.nextInt();
```

System.out.println ("Enter rate & time in yrs");

r = sc.nextInt();

t = sc.nextInt();

ci = ci + p * Math.pow((1+r/100.0), t);

System.out.println ("compound interest = "+ ci);

}

void withdraw()

{

System.out.println ("Enter cash to withdraw");

w = sc.nextInt();

amt = amt - w;

System.out.println ("current balance after
withdrawal = "+ amt);

}

}

class labprepg5

{ public static void main (String args[]) }

System.out.println ("Enter choice (1 : current or
2 : savings account ");

int ch;

Scanner sc = new Scanner (System.in);

ch = sc.nextInt();

if (ch == 1)

{

Current c = new Current();

c.accept();

c.display();

c.checkmin();

}

else if (ch == 2)

}

Sav-acct s = new Sav-acct();
s. accept();
s. display();
s. interest();
s. withdraw();

}

~~else~~

System.out.println ("wrong choice");

}

Command Prompt

```
enter choice 1:current or 2:savings account
1
enter name and deposit amt
ww
23
name: ww
current balance=23
below minimal balance, will be charged penalty
current balance=20
```

```
C:\Users\Amit R>java labprog5
enter choice 1:current or 2:savings account
2
enter name and deposit amt
e
4577
name: e
current balance=4577.0
enter principal amt
340
enter rate and time in yrs
2
3
compound interest=360.81072000000006
current balance after deposit=4937.81072
enter cash to withdraw
470
current balance after withdrawal=4467.81072
```

```
C:\Users\Amit R>
```

Lab 6: Create a package CIE which has two classes- Student and Internals. The class Personal has members usn, name, sem. class Internals has an array that stores the internal marks scored in 5 courses of the current semester of 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 PROGRAM 6 :-

Internal.java

```
package cie;
import java.util.Scanner;
class Student
{
    public String name;
    public String usn;
    public int sem;
    public void details()
    {
        Scanner sc = new Scanner (System.in);
        System.out.println ("enter Name : ");
        name = sc.next();
        System.out.println ("enter USN = ");
        usn = sc.next();
        System.out.println ("enter Semesters : ");
        sem = sc.nextInt();
    }
}
```

3

```
public class Internal extends Student
{
    public double ciemarks [] = new double [5];
    public void accept()
    {
        Scanner sc = new Scanner (System.in);
        for (int i=0; i<5; i++)
    }
```

{
 System.out.println ("CIE marks/50 for subject
 elements [i] = sc. nextDouble () + (i+1) + ":");
 ciemarks [i] = sc. nextDouble ();

3

3
3External.java

package see;
 import cie.*;
 import java.util.Scanner;
 public class External extends cie.Internal
 {
 public double seemarks [] = new double [5];
 public void accept ()

Scanner sc = new Scanner (System.in);
 for (int i=0; i<5; i++) {
 System.out.println ("seemarks (" + i + ") for
 subject " + (i+1) + ":");
 seemarks [i] = sc.nextDouble();

3

3
public void display ()

double t;
 for (int i=0; i<5; i++)

System.out.println ("subject " + (i+1) + "="
 + (ciemarks [i] + (seemarks [i]/2.0)));

3

3
3

Driver class:

import cie.*;

import ee.*;

import java.util.Scanner;

class LabCode

{

 public static void main (String C) args

 int n;

 Scanner sc = new Scanner (System.in);

 System.out.println ("Enter no. of student")

 n = sc.nextInt();

 ee.External e [] = new ee.External [n];

 for (int i=0; i<n; i++)

 e[i] = new ee.External (C);

 e[i] = details();

 e[i] = accept();

 e[i] = acceptee();

 e[i] = display();

 }

}

Procedure:

- Create cie package.
- Under that → create public class Internal that inherits a Student class.
- Create ee package & import cie package in it.
- Under it, create public class External,
- Create a driver class to call the packages & its respective classes & functions.

```
C:\Users\Amit R>javac lab6code.java
C:\Users\Amit R>java lab6code
enter no of students
2
enter Name:
amit
enter USN:
cs016
enter Semester:
3
CIE marks /50 for subject1:
45
CIE marks /50 for subject2:
46
CIE marks /50 for subject3:
44
CIE marks /50 for subject4:
47
CIE marks /50 for subject5:
45
see marks /100 for subject1:
78
see marks /100 for subject2:
77
see marks /100 for subject3:
66
see marks /100 for subject4:
78
see marks /100 for subject5:
79
subject1=84.0
subject2=84.5
subject3=77.0
subject4=86.0
subject5=84.5
enter Name:
james
enter USN:
cs011
enter Semester:
3
CIE marks /50 for subject1:
50
CIE marks /50 for subject2:
```

```
enter Name:  
james  
enter USN:  
cs011  
enter Semester:  
3  
CIE marks /50 for subject1:  
50  
CIE marks /50 for subject2:  
40  
CIE marks /50 for subject3:  
30  
CIE marks /50 for subject4:  
44  
CIE marks /50 for subject5:  
45  
see marks /100 for subject1:  
66  
see marks /100 for subject2:  
56  
see marks /100 for subject3:  
77  
see marks /100 for subject4:  
100  
see marks /100 for subject5:  
25  
subject1=83.0  
subject2=68.0  
subject3=68.5  
subject4=94.0  
subject5=57.5
```

```
C:\Users\Amit R>
```

Lab7: Write a program to demonstrate generics with multiple object parameters.

LAB PROGRAM : 7

class myGen<A,B> {

 A o1;

 B o2;

 myGen<A o1, B o2> {

 this.o1 = o1;

 this.o2 = o2;

}

 void display() {

 System.out.println(obj1);

 System.out.println(obj2);

}

}

public static void main(String args[]) {

 myGen<String, Integer> G1 = new myGen<String, Integer>("Hello", 100);

 myGen<Double, Character> G2 = new myGen<Double, Character>(98.7f, 'J');

G1. display();

G2. display();

}

}

Procedure:-

- Create generic class of 2 objects.
- Create constructor with object arguments & display function.
- In main, create two objects to be displayed.

```
C:\Users\Amit R>javac lab7.java
```

```
C:\Users\Amit R>java lab7
```

```
hello
```

```
100
```

```
98.77
```

```
J
```

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 inputage=father’s age.

LAB PROGRAM: 8

class WrongAge extends Exception

{

 public String testing()
 {
 return "enter the right age!";
 }

}

class Father

{

 public int agef = 0;
 Father (int agef) throws WrongAge

{

 agef = agef;

 if (agef < 0)

 throw new WrongAge();

 else

 System.out.println ("Father's age : " + agef);

}

}

class Son extends Father

{

 int agee = 0;

 Son (int age1, int age2) throws WrongAge

super (age);
ages = age²;
if (ages > agef)
throw new WrongAge();
else
System.out.println ("Son age: " + ages);

3

class lab8

{

public static void main (String args[])

{ int j = 30;

int k = 53;

try {

Father f = new Father (j);

Son s = new Son (j, k);

} catch (WrongAge e) {

System.out.println ("Exception: " + e);

}

3

3

Procedure:

Create custom exception class.

Create Father class, throw exception if age less than zero.

Create Son subclass of Father class, throw exception if son's age > father's age.

Create father & son objects & try to catch exception.

```
C:\Users\Amit R>javac lab8.java
```

```
C:\Users\Amit R>java lab8
```

```
Fathers' age:30
```

```
Fathers' age:30
```

```
Exception enter the right age:
```

```
C:\Users\Amit R>
```

Lab9: 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 PROGRAM : using
class NewThread implements Runnable
{
    String name;
    Thread t;
    int time;
    NewThread (String n, int tm)
    {
        name = n;
        time = tm;
        t = new Thread (this, n);
        System.out.println ("New Thread : " + t);
        t.start();
    }
    public void run()
    {
        try
        {
            for (int i=5; i>0; i--)
                System.out.println (name + " : " + i);
            Thread.sleep (time);
        }
        catch (InterruptedException e)
        {
            System.out.println (name + " Interrupted");
        }
    }
}
class lab9
{
    public static void main (String args[])
    {
```

NewThread o₁ = new NewThread ("BMS College of
Engineering", 10000);

NewThread o₂ = new NewThread ("CSE", 2000);

3

3

Procedure:-

- Create runnable implemented class.
- Create a constructor with name & time arguments.
It also creates new thread object.
- Create run function to print from object.
- Use sleep() to ~~wait~~ adjust time.
- Invoke the constructor twice in ~~the~~ main function.

Command Prompt

```
C:\Users\Amit R>java lab9
New thread: Thread[BMS College of Engineering,5,main]
New thread: Thread[CSE,5,main]
BMS College of Engineering: 5
CSE: 5
CSE: 4
CSE: 3
CSE: 2
CSE: 1
BMS College of Engineering: 4
BMS College of Engineering: 3
BMS College of Engineering: 2
BMS College of Engineering: 1
```

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 theResult 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 ArithmeticException. Display the exception in a message dialog box.

LAB PROGRAM : 10

```
import java.awt.*;  
import java.awt.event.*;  
public class lab10 extends Frame implements ActionListener
```

```
{  
    JTextField num1, num2;  
    JButton divide = new JButton ("Divide");  
    float res = 0;  
    String error = " ";  
    public lab10 ()
```

```
{  
    setLayout (new FlowLayout ());  
    num1 = new JTextField (1);  
    num2 = new JTextField (1);  
    Label result = new Label ("num1:", Label.RIGHT);  
    Label error = new Label ("num2:", Label.RIGHT);  
    add (num1);  
    add (num2);  
    add (divide);  
    divide.addActionListener (this);  
    addWindowListener (new WinAdapter ());
```

```
3  
{  
    public void actionPerformed (ActionEvent ae)
```

```
{  
    if (ae.getSource () == divide)
```

```
try
```

```
{
```

```
int n1 = Integer.parseInt (num1.getText ());  
int n2 = Integer.parseInt (num2.getText ());
```

```
res = (float) n1 / n2;
```

```
2  
} catch (NumberFormatException ex) {}
```

```

    res < 0;
    error = "entered non. not integer";
    } catch (ArithmaticException e) {
        res = 0;
        error = "you divided by 0";
    }
    repaint();
}

public void paint (Graphics g) {
    g.drawString ("result : "+String.valueOf(res));
    g.drawString ("error : "+error, 20, 150);
}

public static void main (String args[])
{
    Label l = new Label ();
    l.setSize (new Dimension (400, 400));
    l.setTitle ("Divide");
    l.setVisible (true);
}

class WinAdapter extends WindowAdapter
{
    public void window (WindowEvent we)
    {
        System.exit (0);
    }
}

Procedure :—
Create 2 JTextField objects num1 & num2.

```

- Create divide button.
- Create function to help divide the entered & look for exceptions.

 Divide

—

□

×

result:7.3333335

error:

 Divide

—

□

×

result:0.0

error:entered no. not integer