Lab Program:

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

```
Import java.util.Scanner;
public class Main {
public static void main(String[] args) {double a,b,c,
root 1, root 2; double det;

Scanner sc = new Scanner (System.in);
System.out.println ("Enter the value of b:");
b = sc.nextDouble ();
System.out.println("Enter the value of a:");
a = sc.nextDouble ();
System.out.println("Enter the value of c:");
c = sc.nextDouble ();

det = b*b-4*a*c;
if (det >0) }
 { root 1 = (-b+ Math.sqrt (b*b- 4*a*c))/2*a;
System.out.println ("First root is :"+ root 1);
System.out.println ("Second root is :"+ root 2); }
else if (det == 0) }
root 1 = -b/2*a);
```

```
System.out.println ("Both roots are same and are
equal to :" + root 1) ; }

 else if ( det < 0) }
 System. out. println ("Real roots don't exist");

 } } }
```

Algorithm :

```
Start

double a, b, c, root 1, root 2, det

input a, b, c

det = b * b - 4 x a x b

If (det > 0)
 root 1 = $(-b + \sqrt{b \times b - 4ac}))\ 2 \times a$
 root 2 = $(-b - \sqrt{-b * b - 4ac}))\ 2 \times a$
 print root 1, root 2

else if (det = 0)
 root 1 = $-b/(2 \times a)$
  print root 1
else
print "Imaginary root"

End
```

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 calculate SGPA of a student

```
Import Java.util.Scanner,

Class student {
private String USN;
private String name;
private int n;
private double SGPA = 0;
private int total Credits = 0;
Scanner ss = new Scanner( System . in);
void Details () {
System. out. println ("Enter USN of the student
USN = ss.next Line ();
System. out. println( "Enter Name of the student"
name = ss. next Line ();
System . out . println ( "Enter no of subjects");
n = ss.nextInt ();
Int credits [] = new int [n];
double marks [] = new double [n]:
System . out. println ("Enter details of the [illegible]
  for ( int i = 0 ; i<n; i++) {
System. out. println ( "Enter [illegible] +
(i+1)) ;
marks [i] = ss.nextInt ();
Calculate ( credits [i], marks [i],i) ; } }
  Calculate ( int credit, double mark, int j) {
```

```
total Credits = total Credits + credit;
if ( mark >= 90&& mark <= 100)
SGPA = SGPA + (10* Credit);
else if ( mark >=80 && mark <= 89)
SGPA = SGPA + ( 9* credit) ;
else if ( mark >= 70 && mark <= 79)
SGPA = SGPA + ( 8* credit) ;
else if ( mark >= 60&& mark <= 69)
SGPA = SGPA + (7* credit);
else if ( mark >= 50 && mark <= 59)
SGPA = SGPA + (6* credit);
else if ( mark >= 40 && mark <= 49)
SGPA = SGPA + (5* credits); else
System. out. println ("Failed in subject "+( j+1)) ;}
void display () {
System. out. println ("Details of the student");
System. out. println ( "Name :"+ name) ;
System. out. println ("USN:" + USN);
System. out. println ("SGPA of student" + (SGPA/total
Credits )); }}
public class Lab 2 {
public static void main( String args []) {
student s1 = new Student ();
s1. Details ();
s1. Display (); } }
```

Algorithm :

Start
Input USN , Name , no. of subjects and the details of subjects i.e Credits and marks as USN, name, n, credits [i] , marks [i] .
Set totalcredits = totalcredits + Credit
Set SGPA = SGPA + (Credit * number) where number = 10, 9, 8, 6, 7, 5 acc to marks.
Else print "Failed in subjects"
Print " Details of the student", name, USN and the calculated SGPA of the student.
End.

LAB-3

```
import java.util.Scanner;
Class Book {
String name;
String author;
int price;
int num-pages;
Book ()
{}
Book (String name, String author, int price, int num_pages)
{
this.name = name;
this.author = author;
this.price = price;
this.num_pages = num_pages;
}
```

```
void accept ()
{
Scanner s = new Scanner (System.in);
System.out.println ("Enter the name of the book");
name = s.next();
System.out.println ("Enter the author of the book");
author = s.next();
System.out.println ("Enter the price of the book");
price = s.nextInt ();
System.out.println ("Enter the number of pages of the book");
num_pages = s.nextInt ();
}
public String toString ()
{
return ("Name : " + name + "\n" + "Author : " + author +
"\n" + "Price : " + price + "\n" + "Number of pages :
" + num_pages );
}
}
Class BookMain {
public static void main (String s[]) {
Scanner a = new Scanner (System.in);
Book b1 = new Book ("Mystic", "Anne", 299, 575);
System.out.println ("Sample input : \n" + b1);
System.out.println ("Enter the number of books");
int n = a.nextInt ();
Book b[] = new Book [n];
for (int i = 0; i < n; i++)
{
b[i] = new Book ();
System.out.println ("Enter the details of " + (i+1) +
" book");
```

```
b[i].accept ();
}
for (int i = 0; i < n; i++)
{
System.out.println ("Details of book " + (i+1));
System.out.println (b[i]);
}
}
}
```

```
Q. 4
import java.util.Scanner;
abstract class Shape {
int length, breadth;
void print Area ()
{} }
class Rectangle extends Shape {
double areaR;
void printArea (){
areaR = (length * breadth);
System.out.println ("The area of rectangle is " + areaR + "cm^2");
}}
class Triangle extends Shape
{
double area T;
void printArea (){
area T = (0.5) * (length * breadth);
System.out.println ("The area of Triangle is " + area T + "cm^2");
}}
class Circle extends Shape {
double area C;
void print Area (){
area C = (3.14) * (length * length);
System.out.println (" The area of circle is " + area C + "cm^2");
} }
class Main {
```

```
public static void main ( String args [] ) {
Scanner A = new Scanner ( system . in ) ;
Rectangle R1 = new Triangle () ;
Circle C1 = new Circle () ;
System . out . println (" Enter the length ad breadth of which
u have to find the area of rectangle in cm\n") ;
R1. length = A . nextInt () ;
R1 breadth = A . nextInt () ;
System . out . println ("Enter the length ad breadth of which u have
to find the area of triangle in cm\n") ;
T1. length = A. nextInt () ;
T1. breadth = A. nextInt () ;
System. out. println ("Enter the length of which u have to
find the area of circle in cm\n") ;
C1 length = A. nextInt () ;
R1. print Area () ;
T1. print Area () ;
C1. print Area () ;

}
}
```

LAB 5

```
import java.util.Scanner;

class Bank
{
int deposit_balance;
String customername;
String Account_Number;
String Account_Type;
int Balance = 27810;
void accept () {
Scanner s = new Scanner (System.in);
System.out.println ("Enter the customer name\n");
customername = s.next ();
System.out.println ("Enter the Account Number\n");
Account_Number = s.next ();
System.out.println ("Enter the Account type\n");
Account_Type = s.next (); }
void display () {
System.out.println ("CUSTOMER NAME:" + customername);
System.out.println ("ACCOUNT NUMBER:" + Account_Number);
System.out.println ("ACCOUNT TYPE :" + Account_Type);

}
}
class curr_acct extends Bank {
int updated_balance;
int After_cwithdrawn;
int updated_lost_cbalance;
int cdepo_ba(){
updated_balance = Balance + deposit_balance;
return updated_balance; }
int cwith_ba(){
After_cwithdrawn = ((updated_balance)-(withdrawn_balance));
return After_cwithdrawn; }
int minimum () {
if ((After_cwithdrawn)<=(2000)) {
updated_lost_cbalance = ((After_cwithdrawn)-(200));
System.out.println("you have minimum balance below 2000
so u have lost 200 rs");

return updated_lost_cbalance; }

else
return After_cwithdrawn; } }
class sav_acct extends Bank {
int supdated_balance;
int After_swithdrawn;
int updated_lost_sbalance;
int compound_interest;
int sdepo_ba(){
supdated_balance = Balance + deposit_balance;
return supdated_balance; }
int interest () {
```

```
double r = 0.08;
int n = 12;
int t = 5;
compound _ interest = (int)
(( supdated _ balance)* (Math. pow ((1 + (r/n)), (n*t))));
return compound _ interest ; }

int minimum () {
if (( After _ swithdrawn) <= ( 1000)) {
updated _ last _ sbalance = (( After _ swithdrawn ) - ( 100));
return updated _ last _ sbalance ; }
else
return After _ swithdrawn ; } }

class Transactions {
public static void main ( String args []) {
Scanner r = new Scanner ( System . in);
Curr _ acct CA = new Curr _ acct ();
CA . accept () ;
System . out . println ("Enter the money u want to deposit in current
account in rupees");
CA. deposit _ balance = r. nextInt ();
CA. display () ;
System . out . println ("After your deposition of " + CA. deposit _
balance + "\nNow your total balance is Rs-" + CA . cdepo - ba ();
System . out . println ("Enter the money you want to withdraw
in rupees");
CA. withdraw _ balance = r. nextInt ();
System. out. println ("After your withdrawal of
" + CA. withdraw _ balance + "\n Now your total balance is
RS-" + CA cwith _ba ());
System . out . println ("After checking
if u have minimum balance are not your updated total
balance is RS-" + CA . minimum ()} ;
sav _ acct SA = new sav _ acct (); SA. accept ();
System. out . println ("Enter the money u want to deposit in
Saving account");
SA. deposit _ balance = r. nextInt ();
SA. display ();
System . out. println ("After your deposition of
" + SA . deposit _ balance + "\n Now your total balance is
RS-" + SA. sdepo_ba ());
System . out . println ("Enter the money you want to withdraw
in Saving account")
SA. withdraw _ balance = r. nextInt ();
System . out. println ("After your withdrawal of
RS -" + SA. withdraw _ balance + "\n Now your total balance is
RS-" + SA. Swith _ ba ()) ;
System . out. println ("After checking if u have minimum balance are
not your updated total balance is
RS-" + SA. minimum ());
} }
```

LAB Program 6 -

```
package CIE ;
public class Student {
  public int usn;
  public String name;
  public int sem;
public ~~class~~ Student (int usn, String name,
   int sem) {
      this. usn = usn;
      this. name = name;
      this. sem = sem;
}}

 public class internals extends Student {
   public int [] cieMarks = new int [5];
  public internals (int usn, String name,
  int sem, int [] cieMarks) {
   super (usn, name, sem)
      this. cieMarks = cieMarks;
  }}

package SEE;
 import CIE.*;
 public class externals extends Student {
  int [] seeMarks = new int [5];
public externals (int usn, String name, int sem,
  int [] seeMarks) {
  super (usn, name, sem);
  this. seeMarks = seeMarks;
}}
```

```
// In default package
import ~~se~~ SEE.*;
import CIE.*;
class Main{
 public static void main(String[] args){
 int usn = 131;
 String name = "Ritika";
 int sem = 3;
 int[] cie = {44, 48, 46, 40, 41};
 int[] see = {90, 88, 89, 92, 91};
 internals in = new internals(usn, name,
  sem, cie);
 internals ex = new externals(usn, name,
 sem, see);
 System.out.println(" name:"+ ~~name~~
  in.name+" usn: "in.usn+" sem"
  +in.sem);
 int final = 0;
  for(int i=0; i<5; i++)
 { final = in.cieMarks[i]+ ex.seeMarks
 [i];
 System.out.println("Final marks are:"
   +final+"   ");
 }}}
```

LAB-7

```
class Generics<T, U >{
T ob1;
U ob2;
Generics(T x, U y){
   ob1=x;
   ob2=y; }
```

```
T getob1(
 return
U getob2
  retur
void displ
System.o
 + getob
U join(){
 if(ob1
     Int
 int i1
 int i2
 return
else if(
Double
double
double
return (
else if (
 String
 String
 String
return
else {
 return
 Type N
 }}}

class
 publi
  Ge
<Inte
```

```
T getob1(){
  return ob1; }
U getob2(){
   return ob2; }
void display(){
System.out.println("Ob1: "+ getob1() + "ob2: "
  + getob2()); }
U join(){
  if(ob1 instanceof Integer && ob2 instanceof
      Integer) {
  int i1 = (Integer) getob1();
  int i2 = (Integer) getob2();
 return (U) new Integer(i1+i2); }
else if(ob1 instanceof Double && ob2 instanceof
Double) {
 double d1 = (Double) getob1();
double d2 = (Double) getob2();
return (U) new Double(d1+d2); }
else if (ob1 instanceof String && ob2 instanceof
  String)
  String s1 = (String) getob1();
  String s2 = (String) getob2();
return (U) new String(s1+s2); }
else {
return (U) new String("ERROR! ob1 and ob2
Type Mismatch");
}}}

class MyMain{
 public static void main(){
  Generics < Integer, Integer > Obj= new Generics
<Integer, Integer> (5,4);
```

```
iobj.display();
System.out.println("Sum: "+ iobj.join());
Generics<Double, Double> dobj= new Generics
 <Double , Double >(03.05, 4.02);
 dobj.display();
System.out.println("Sum : "+ dobj.join());
Generics <String, String> sobj = new
Generics < String, String > ("Hello",
 "How are you");
 sobj.display();
System.out.println ~~"Sum: "+obj.~~
  ("Concatanation: "+sobj.join());
}}
```

LAB-8

```
class father {
static void acceptNameF(int inputAge)
throws ArithmeticException{
try
{ if(inputAge < 0)
throw new ArithmeticException("Wrong
 Age");
}
catch(ArithmeticException e){
System.out.println("Caught "+e);
 }}}
Class Son extends father {
static void checkSFAge(int S_Age,
int F_Age) throws ArithmeticExcep-
tion
{ try{
```

```
if (S_Age >=
throw new
should be s
System. out.
 "Father's
 }
catch (Arith
System. ou
}}}
public cl
public st
 Father.
 Son. Ch
}}
```

LAB-9

```
import ja
import ja

class N
Thread
String s
int x;
Newthre
 s = the
this. x
 t = ne
System.
 t. s

public
```

```
if (S_Age >= F_Age)
throw new ArithmeticException ("Son's age
 should be smaller than father's age");
System.out.println("Son's age is" + S_Age +
 "Father's Age is" + F_Age);
 }
catch (ArithmeticException.e)
 System.out.println ("F Caught" + e);
 }}}
 public class MyClass {
 public static void main (String[] args) {
  Father.acceptNameF(-10);
  Son.CheckSFage(30,20);
}}
```

LAB-9

```
import java.util.*;
import java.lang.*;

class newthread implements Runnable {
 Thread t;
 String s;
 int x;
Newthread (String threadname, int x) {
  s = threadname;
  this.x = x;
  t = new Thread (this, s);
 System.out.println(" Thread created");
   t.start(); }

public void run() {
```

```
try {
for (int i = 0; i<10 ; i++) {
 System.out.println (S);
 Thread. sleep (x);
}}
catch (InterruptedException e) {
System. out. println("Thread interrupted");
}}}

public class MyClass {
public static void main (String [] args){
 new NewThread ("BMS Colleage of
 Engineering", 10000);
 new NewThread ("CSE", 2000);
}}
```

Lab 10

```
import java. ~~util~~ awt
import java. awt. event. *;
import java. applet
class division extends Applet implements ActionListener{ int
String msg;
Textfield num1, num2, res;
Label l1, l2, l3;
Button div;
public void int() {
l1 = new Label ("Dividend");
l2 = new Label ("Division");
l3 = new Label ("Result");
num1 = new Textfied (10);
num2 = new Textfied (10);
res = new Textfied (10);
```

```
div = new Button
div = addAction
add ( ~~[illegible]~~ l1);
add (l2);
add ( num2);
add (l3);
add (res);
add (div);
}
public void acti
string arg = ac
int num1 = 0
if( arg. equals (
if ( this. num1
   is Empty ()){
msg = "Enter H
repaint ();
else {
 try{
num1 = Intege
num2 = Integer
num3 = nu
res. setText (S
msg = "Operat
repaint () }
catch (Number
 System. out.
 res. setText
msg = " can't
repaint(); }
 public void
   g. dra
) {

e) {
 interrupted");

ring[] args){
eage of

000);

ts ActionListner { int
```

```
div = new Button ("click");
div = addAction Listner ("this");
add ( ~~num~~ l1);  add( num1);
add (l2);
add ( num2);
add (l3);
add (res);
add (div);
}
public void actionperformed (ActionEvent as) {
String arg = ae.getActionCommand();
int num1 = 0, num2 = 0;
if (arg.equals ("click")){
if ( this.num1.getText().isEmpty() && this.num2.getText().
   isEmpty()){
msg = "Enter the valid numbers";
repaint();
else {
try {
num1 = Integer.parseInt(this.num1.getText());
num2 = Integer.parseInt ( this.num2.getText());
num3 = num1/num2;
res.setText (String Valueof(num3));
msg = "Operation successfull";
repaint(); }
catch (NumberFormat Exception ex) {
System.out.println(ex);
res.setText (" ")
msg = " can't be divided by 0"}
repaint(); }}}}
public void paint (Graphics g){
   g.drawstring(msg, 30, 70); }}
```