

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

classmate

Date _____

Page _____

(1) Quadratic Equations

```
import java.lang.Math;  
import java.util.Scanner;  
Class Ques
```

```
{
```

```
public static void main(String arr[])
```

```
{
```

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

```
System.out.println("Enter the coefficients a,b,c");
```

```
double a = s.nextInt();
```

```
double b = s.nextInt();
```

```
double c = s.nextInt();
```

```
double r1, r2, d;
```

```
d = b*b - 4*a*c;
```

```
double res = Math.sqrt(d);
```

```
if(a == 0)
```

```
System.out.println("Invalid input for a");
```

```
else
```

```
{
```

```
if(d > 0)
```

```
{
```

```
System.out.println("The roots are real  
and distinct");
```

~~```
r1 = (-b + Math.sqrt(d)) / (2 * a);
```~~~~```
r2 = (-b - Math.sqrt(d)) / (2 * a);
```~~

```
System.out.println("The roots are: " + r1 + "  
" + r2);
```

```
else if(d == 0)
```

```
{
```

$$r_1 = r_2 = (-b)/(2*a);$$

System.out.println("The roots are real
and equal");

System.out.println("Roots are r1 = r2 = " + r1);

y
else

{

$$r_1 = -b/(2*a);$$

$$r_2 = (\text{math.sqrt}(\text{math.abs}(d))) / (2*a);$$

System.out.println("The roots have no real
solution and are imaginary");

System.out.println(r1 + " + i" + r2 + "i" + r1 + " - i" + r2);

y
y

y
y

Output:-

Enter the coefficients a, b, c.

0

1

2

Invalid input for a

Enter the coefficients a, b, c.

1

2

3

The roots have no real solution and are imaginary

-1.0 + i 1.41421356

1.41421356 - i 1.41421356

Enter The coefficients a, b, c

1

2

1

The roots are real and equal

Roots are $g_1 = g_2 = -1.0$

Enter The coefficients a, b, c

1

4

1

The roots are real and distinct

The roots are: 0.2679491921 -3.7320508075

Command Prompt

```
C:\Users\nbrij>cd C:\Engg\3rd sem\JAVA lab
C:\Engg\3rd sem\JAVA lab>javac quad.java
C:\Engg\3rd sem\JAVA lab>java Quad
Enter the coefficients a,b,c
0
1
2
Invalid input for a
C:\Engg\3rd sem\JAVA lab>java Quad
Enter the coefficients a,b,c
1
2
3
The roots have no real solution and are imaginary
-1.0+i1.4142135623730951
1.4142135623730951-i1.4142135623730951

C:\Engg\3rd sem\JAVA lab>java Quad
Enter the coefficients a,b,c
1
2
1
The roots are real and equal
Roots are r1=r2=-1.0
C:\Engg\3rd sem\JAVA lab>java Quad
Enter the coefficients a,b,c
1
4
1
The roots are real and distinct
The roots are:-0.2679491924311228 -3.732050807568877

C:\Engg\3rd sem\JAVA lab>
```

Week - 2

WAP to create a class student with members USN, name, an array credit and an array mark. Include methods to accept and display details and a method to calculate CGPA of a student.

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

```
class student
```

```
{
```

```
    String name, usn;
```

```
    int marks[] = new int[5];
```

```
    int credits[] = new int[5];
```

```
    void input()
```

```
{
```

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

```
        System.out.println("Enter your name:");
```

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

```
        System.out.println("Enter your USN:");
```

```
        usn = s.next();
```

```
        System.out.println("Enter the marks of each subject:");
```

```
        for(int i=0; i<5; i++)
```

```
            marks[i] = s.nextInt();
```

```
        System.out.println("Enter the no. of credits for each subject");
```

```
        for(int j=0; j<5; j++)
```

```
            credits[i] = s.nextInt();
```

void display()

```

System.out.println("Name: " + name);
System.out.println("UAN: " + UAN);
for (int i = 0; i < 5; i++) {
}

```

```

System.out.println("Marks of subject " + (i + 1) +
    " : " + marks[i]));

```

```

System.out.println("No. of credits for the subject " +
    " above: " + credits[i]));

```

3
3

void calc()

```
{
}

```

```

int gr = point[ ] = new int[5];

```

```

int sum = 0;

```

```

int avg = 0;

```

```

float res;
for (int i = 0; i < 5; i++)
{
}

```

```

if (marks[i] >= 90)

```

```

gr = point[i] = 10;

```

```

else if (marks[i] >= 80)

```

```

gr = point[i] = 9;

```

```

else if (marks[i] >= 70)

```

```

gr = point[i] = 8;

```

```

else if (marks[i] >= 60)

```

```

gr = point[i] = 7;

```

```

else if (marks[i] >= 50)

```

```

gr = point[i] = 6;

```

```

else if (marks[i] >= 40)

```

```

gr = point[i] = 5;

```

```

else if (marks[i] < 35 && marks[i] > 0)
    gr_point = 0;
else
    System.out.println("Invalid input
        for " + (i + 1) + " subject");
    sgpa += gr_point[i] * credits[i];
    sum += credits[i];
res = (float) sgpa / sum;
System.out.println("SGPA is " + res);
}

class Sgp
{
    public static void main (String ar[])
    {
        Student s1 = new Student();
        s1.input();
        s1.display();
        s1.calc();
    }
}

```

[week 2]

Output [Program -2]

Enter your name:

Brijesh N

enter your usn:

40

enter the ~~marks~~ marks of each subject

66

77

82

56

87

enter the no of credits for each subject

4

3

3

1

3

Name: Brijesh N

USN: 40

marks of subject 1 = 66

no of credits for subj above = 4

marks of subject 2 = 77

no of credits for subj above = 3

marks of subject 3 = 88

no of credits for subj above = 3

marks of subject 4 = 86

~~marks~~ no of credits for subj above = 1

marks of subject 3 = 87

no of credits for subj above = 3

sgpa = 8.0

```
C:\Users\bmsce\Desktop\1BM21CS040>java Sgpa
enter your name:
Brijesh N
enter your usn:
40
enter the marks of each subject
66
77
88
56
87
enter the no of credits for each subject
4
3
3
1
3
name:Brijesh N
usn:40
marks of subject1=66
no of credits for the subj above=4
marks of subject2=77
no of credits for the subj above=3
marks of subject3=88
no of credits for the subj above=3
marks of subject4=56
no of credits for the subj above=1
marks of subject5=87
no of credits for the subj above=3
sgpa=8.0
```

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 object. Include a toString() method that could display the complete details of the book. Develop a Java program to create 3 book objects.

```

import java.util.Scanner;
class Book
{
    String name, author;
    double price;
    int num_pages;
    Book()
    {
        name = " "; author = " ";
        price = 0.0, num_pages = 0;
    }
    void input()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the name of the book:");
        name = s.nextLine();
        System.out.println("Enter author of the book:");
        name = s.nextLine();
        System.out.println("Enter the price of the book:");
        price = s.nextDouble();
    }
}

```

```

System.out.println ("Enter the number of
pages ");
num_pages = s.nextInt ();
}

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

class Book
{
    public static void main (String args[])
    {
        Scanner s = new Scanner (System.in);
        System.out.println ("Enter the number
of books");
        int n = s.nextInt ();
        Book books [] = new Book [n];
        for (int i = 0; i < n; i++)
        {
            books [i] = new Book ();
            books [i] = input ();
            System.out.println ("Book Details");
            System.out.println (books [i].toString ());
        }
    }
}

```

Output :-

Enter no number of books:

1

Enter the name of the book:

Power of thinking

Enter the author of the book:

James

Enter the price of the book:

450

Enter the number of pages:

200

Book Details :

Name : Power of thinking

Author : James

Number of pages : 200

```
Enter the number of books:  
2  
Enter the name of the book:  
Power of thinking  
Enter the author of the book:  
James  
Enter the price of the book:  
450  
Enter the number of pages of the book:  
200
```

```
BOOK DETAILS:  
NAME: Power of thinking  
AUTHOR: James  
PRICE: 450.0/-  
NUMBER OF PAGES: 200
```

```
Enter the name of the book:  
think like a monk  
Enter the author of the book:  
jay shetty  
Enter the price of the book:  
500  
Enter the number of pages of the book:  
450
```

```
BOOK DETAILS:  
NAME: think like a monk  
AUTHOR: jay shetty  
PRICE: 500.0/-  
NUMBER OF PAGES: 450
```

Program - 4

WAP to create an abstract class called shape that contains 2 integer and an empty method named printArea(). Provide 3 classes named Rectangle, triangle and circle such that each one of the classes extends the class shape. Each one of the classes contains only one method printArea() that prints the area of the given shape.

import java.util.Scanner;
abstract class shape

{

shape() { }

int h, l;

abstract void printArea();

{}

Class rectangle extends shape

{

Scanner s = new Scanner(System.in);
void printArea()

{

System.out.println("Enter length and breadth of rectangle")

l = s.nextInt();

b = s.nextInt();

System.out.println("Area of rectangle is "+l*b)

rectangle(); }

class triangle extends shape

{
Scanner s = new Scanner(System.in);
void printArea()
{

System.out.print("Enter height
and base of triangle");

b = s.nextInt();

l = s.nextInt();

System.out.println("Area of Triangle
is "+0.5*l*b));

}
triangle()
{ }
}

class circle extends shape

{
Scanner s = new Scanner(System.in);
void printArea()
{

System.out.print("Enter radius
of circle");

r = s.nextInt();

System.out.println("Area of circle
is "+3.14*r*r));

}
circle()
{ }
}

Class main

{

public static void main (String a [])

rectangle r = new rectangle();

r.printArea();

triangle t = new triangle();

t.printArea();

circle c = new circle();

c.printArea();

3

3

Output

Enter length and breadth of rectangle

2

3

Area of Rectangle is 6

Enter height and base of triangle

3

6

Area of Triangle is 9.0

Enter ~~the~~ radius of circle

4

Area of Circle is 50.24

```
Enter height and width of rectangle
2
3
Area of Rectangle is 6
Enter height and base of rectangle
3
6
Area of Trianle is 9.0
Enter radius of Circle
4
Area of Circle is 50.24
```

```
C:\Engg\3rd sem\JAVA lab\p4>
```

Program - 5

WAP 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 withdraw 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, acc number and type of account. From this derive the classes cur-account 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 no balance
- (c) Compute and deposit interest
- (d) Permit withdrawal and update balance

Check for minimum balance, impose
penalty and ~~update~~ update the
balance.

```
import java.util.Scanner;  
import java.lang.Math;
```

```
class account
```

```
{  
    String name = new String();  
    int accno;  
    double bal;
```

```
    Scanner s = new Scanner(System.in);  
    void set()  
{
```

```
        System.out.println("Enter customer name");  
        name = s.nextLine();
```

```
        System.out.println("Enter " + name + " account number");  
        accno = s.nextInt();
```

```
        System.out.println("Enter balance amount");  
        bal = s.nextDouble();
```

```
}
```

```
void display()
```

```
{  
    System.out.println("Customer Name: " + name);  
    System.out.println("Your account number:  
                      " + accno);
```

```
    System.out.println("Your Account Balance: " +  
                      bal);
```

```
} account() {} }
```

```
}
```

class Savacct extends account

{

Scanner s = new Scanner(System.in);
Savacct()

{

System.out.println("Cheque
Faculty not available")

}
void deposit()

{

int ch;

double amt;

System.out.println("Please enter
deposit");

ch = s.nextInt();

if (ch == 1)

{

System.out.println("Enter amount to be
deposited");

amt = s.nextDouble();

bal = bal + amt;

}

else

System.out.println("Invalid Input")

}
void in()

{

System.out.println("Enter rate of
interest");

double r = s.nextDouble();

System.out.println("Enter number of
times interest applied per time period")

```

int n = s.nextInt();
System.out.println("Enter number of
time periods");
int t = s.nextInt();
double x = (1 + (r / 100));
double ci = bal * Math.pow(x, n);
System.out.println("Interest amount
= " + ci + " Balance amount
without interest is " + bal);
System.out.println("Available balance
after updating is " + ci);

3 void wd()
{
    System.out.println("Press 1 to withdraw
amount");
    int ch = s.nextInt();
    if(ch == 1)
    {
        System.out.println("Enter amount to be
withdrawn");
        double wdraw = s.nextDouble();
        bal = bal - wdraw;
        System.out.println("Available Balance: " + bal);
    }
    else
        System.out.println("Invalid input");
    if(bal < 1000)
    {
        System.out.println("Balance below
minimum amount. A penalty
of 50Rs has been credited");
        bal = bal - 50;
    }
}

```

System.out.println("Your Available
Balance: " + bal);

public class Lab5

public static void main(String[] args)

Scanner s = new Scanner(System.in);
int ch;

System.out.println("Enter your
Account type [1. Savings account
2. Current account]");

ch = s.nextInt();
switch(ch)

{
case 1:

Savaccct s1 = new Savaccct();
s1.set();

s1.display();

s1.deposit();

s1.in();

s1.withdraw();

break;

case 2:

Current c1 = new Current();

c1.set();

c1.display();

c1.deposit();

c1.withdraw();

break;

default: s1.set();
3 3 3

Output :-

Enter your Name:

Brijesh

Enter your Account Number:

123456

Enter your Account type: (Savings/CURRENT)
Savings

Enter the Bank Balance:

3000

Enter your choice

1. Deposit
2. calculate interest
3. Withdraw
4. Display
5. Exit

1

Enter the amount to be deposited:

5000

Balance Amount: 35000.0

Enter your choice

1. Deposit
2. calculate interest
3. withdraw
4. Display
5. Exit

~~Enter amount~~ 3

Enter amount to be withdrawn:

5000

Account Balance after withdrawal is:
30000.0

```
Command Prompt — X

Enter your account type:
1. Savings account
2. Current account
1
Cheque Facility not available
Enter customer name
ee
Enter ee's account number
22
Enter balance amount
10000
Customer Name:ee
Your account number:22
Your Account Balance:10000.0
Press 1 to deposit
2
Invalid Input
Enter rate of interest
5
Enter number of times interest applied per time period
3
Enter number of time periods
2
Interest amount=11576.25000000002
Balance amount without interest is10000.0
Available balance after updating is11576.25000000002
Press 1 to withdraw ammount
1
Enter the amount to be withdrawn
500
Available Balance:9500.0
C:\Users\bmsce\Desktop>java Lab5
```

```
 Command Prompt
C:\Users\bmsce>javac quadratic.java
C:\Users\bmsce>java quadratic
enter the value of a b c
1
a is 1
1
b is1
1
c is1
d is-3
roots are imaginary
r1=0.0+i0.8660254037844386
r2=0.0-i0.8660254037844386

C:\Users\bmsce>java quadratic
enter the value of a b c
1
a is 1
4
b is4
2
c is2
d is8
roots are real and distinct
the roots are -2.585786437626905 and -5.414213562373095

C:\Users\bmsce>java quadratic
enter the value of a b c
2
a is 2
4
b is4
2
c is2
d is0
roots are real and equal
the roots are -1.0 and -1.0
```

```
Enter your account type:  
1. Savings account  
2. Current account  
2  
Cheque Facility available  
Enter customer name  
er  
Enter er's account number  
23  
Enter balance amount  
50000  
Customer Name:er  
Your account number:23  
Your Account Balance:50000.0  
Press 1 to deposit  
1  
Enter amount to be deposited  
4500  
Press 1 to withdraw amount  
1  
Enter the amount to be withdrawn :  
2500  
Available Balance:52000.0  
C:\Users\bmsce\Desktop>
```

Activate Windows
Go to Settings to activate Windows.

Lab Program 6

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() either the input age is < 0. In son class, implement a constructor that uses both father and son's age and renews an exception if son's age is > - father's age.

import java.util.Scanner;

Class WrongAge extends Exception

```

String msg = new String();
WrongAge(String s)
{
    msg = s;
}
public String toString()
{
    return msg;
}
```

Class ErrorAge extends WrongAge

```

String msg1 = new String();
ErrorAge(String ss)
{
    msg1 = ss;
}
```

```

msg1 = ss;
public String toString()
{
    return msg1;
}
```

class Father

int age;

Scanner ss = new Scanner(System
Father());System.out.println("Enter
the father's age:");

age = ss.nextInt();

void ex1() throws WrongAge

if (age <= 0)

throw new WrongAge

("Invalid input. Father's
age cannot be lesser than 0");

age = ss.nextInt();

class Son extends Father {

int age;

son();

System.out.println("Enter the
age of son:");

age = in.nextInt();

void ex2() throws ErrorAge

if (age <= 0 || age > super.age)

{

throw new ErrorAge("age of
father less than son");

}

else

{

System.out.println("Father's Age: " +
+ user.age + " Son's Age: " + age);

}

}

}

class A{}

{

public static void main(String[] args)

{

try

{ s.exit(); }

catch(WrongAge e)

{ System.out.println(e); }

try

{ s.exit(); }

catch(ErrorAge e)

{ System.out.println(e); }

}

Output

Enter the father's age
30

Enter the age of son:
10

Father's Age: 30

Son's Age: 10

④ Enter the father's age:
5

Enter the son's age:

10

age of father less than son

Enter the father's age:

0

Enter the age of son:

5

Invalid input. Father's age cannot
be lesser than 0.
age of father less than son.

(With)
10
5
10
5

```
C:\Users\nbrij>cd C:\Engg\3rd sem\JAVA lab\p4
```

```
C:\Engg\3rd sem\JAVA lab\p4>javac p4.java
```

```
C:\Engg\3rd sem\JAVA lab\p4>java A_main
```

```
Enter the father's age:
```

```
0
```

```
Enter the age of son:
```

```
12
```

```
Invalid input. Father's age can not be lesser than 0  
age of father less than son
```

```
C:\Engg\3rd sem\JAVA lab\p4>java A_main
```

```
Enter the father's age:
```

```
23
```

```
Enter the age of son:
```

```
2
```

```
FATHER'S AGE:23
```

```
SON'S AGE:2
```

```
C:\Engg\3rd sem\JAVA lab\p4>java A_main
```

```
Enter the father's age:
```

```
22
```

```
Enter the age of son:
```

```
25
```

```
age of father less than son
```

```
C:\Engg\3rd sem\JAVA lab\p4>
```

Lab Program 7

Write a program which creates 2 threads,
 1 thread displaying 'BMS College of Engineering'
 once every 10 sec and other displaying 'CSE'
 once every 2 sec.

class Call implements Runnable

{ String a;

int t_n, t_i, t_{er};

Thread t;

call(String tn, int ti, int er)

{

a = tn;

x = ti;

time = t_{er};

t = new Thread(this, a);

t.start();

}

public void run()

{

try{

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

System.out.println(a);

Thread.sleep(time);

}

catch(InterruptedException ie)

{

System.out.println("Interrupted");

}

}

}

}

Class ~~Object~~ Program

public static void main (String args [])

{
 ~~System.out.println~~

 new Call ("BMS college of Engineering", 10000, 1);
 new Call ("CSE", 2000, 10);

}
}

Output

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

CSE

CSE

CSE

CSE

C:\Engg\3rd sem\JAVA lab>jav

BMS College of Enginnering

CSE

CSE

CSE

CSE

CSE

CSE

CSE

BMS College of Enginnering

CSE

CSE

CSE

CSE

C:\Engg\3rd sem\JAVA lab>