

Quadratic Equation

```

import java.util.Scanner;

class Quad {
    public static void main (String args) {
        double a,b,c,r1,r2,d;
        Scanner s1 = new Scanner (System.in);
        System.out.print ("Enter variable \n");
        a = s1.nextDouble();
        b = s1.nextDouble();
        c = s1.nextDouble();
        if (a==0) {
            System.out.print ("Invalid input");
        }
        else {
            d = b*b - 4*a*c;
            if (d>0)
                {
                    System.out.print ("\n Roots are real");
                    r1 = (-b + Math.sqrt(d)) / (2*a);
                    r2 = (-b - Math.sqrt(d)) / (2*a);
                    System.out.print ("Roots are " + r1 + " " + r2);
                }
            else if (d<0)
                {
                    System.out.print ("\n Roots imaginary");
                    r1 = -b / (2*a);
                    r2 = (Math.sqrt(Math.abs(d))) / (2*a);
                }
        }
    }
}

```

System.out.println("Roots are equal")

3

else

System.out.print("The Roots are equal")

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

System.out.print("The Roots is r=" + r1);

3

3

3

OUTPUT

Enter variable "a"

-4

+4

~~Roots are equal~~

$$r = 2.00$$

3.4. Standard Input, Output

Other Data Structure

② Grocery

Class Grocery {

String C-name;

String C-Ph;

double total;

Grocery (String C-name, String C-Ph) {

this.C-name = C-name;

this.C-ph = C-Ph;

}

void calc (double q-dal, double q-pulses, double q-sugar)

void calc (double q-dal, double q-pulses, double q-sugar)

{
total = q-dal * 100 + q-pulses * 80 + q-sugar * 50;

}

void display()

{
System.out.println("Name : " + C-name + "
Phone number : " + C-ph + "
Total : " + total);

System.out.print("\nC-name : " + C-name + "
C-ph : " + C-ph + "
Total : " + total);

System.out.println();

}

}

Class Gdemo {

Public static void main (String args[]) {

Grocery g1 = new Grocery ("Rama", "806");

Grocery g2 = new Grocery ("Sharma", "721");

```

Grocery g3 = new Grocery ("Brama", "967");
g1 . calc (32,1);
g1 . display ();
g2 . calc (3.5,2);
g2 . display ();
g3 . calc (1,1,0.5);
g3 . display ();

```

Total amount = 41.00

Name : Brama, 967

Phone number : 9672132000

Date : 10/10/2017

Time : 10:00 AM

Total amount = 41.00

Output

	Phone number	total
Name	9672132000	41.0
Rama	9672132000	8.00
Brama	9672132000	8.00
Srama	9672132000	30.5
T Brama	9672132000	30.5

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

```
class Student {
```

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

```
    String name;
```

```
    int marks[] = new int[6];
```

```
    public void acceptDetails() {
```

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

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

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

```
        System.out.print("Enter marks for 6 subjects: ");
```

```
        for (int i=0; i<marks.length; i++) {
```

```
            System.out.print("Subject " + (i+1) + ": ");
```

```
            this.marks[i] = sc.nextInt();
```

3

3

```
    public void displayDetails() {
```

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

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

```
        System.out.print("Percentage: ");
```

```
        for (int i=0; i<marks.length; i++) {
```

```
            System.out.print(marks[i] + "% ");
```

~~Percentage() + " %");~~

~~System.out.println("Percentage: " + calculatePercentage());~~

?

~~System.out.println("Percentage: " + (marks[i] + (i+1) * 10));~~

~~System.out.println("Percentage: " + calculatePercentage());~~

Percentage() + "%");

System.out.println("Enter number of students");

3

```
public double calculate_Percentage (int marks[])
{
    int total_marks = 0;
    for (int mark : marks)
    {
        total_marks += mark;
    }
    return (double) total_marks / marks.length;
}
```

Public class Main {

```
public static void main (String [] args)
{
    Scanner sc = new Scanner (System.in);
    System.out.println ("Enter number of students:");
    int num_students = sc.nextInt ();
    Student [] students = new Student [num_students];
    for (int i = 0; i < num_students; i++)
    {
        Student [i] = new Student ();
        System.out.print ("Enter details for student " + i + ":");
        students [i].accept_Details();
    }
    System.out.println ("Displaying student details");
    for (Student student : students)
    {
        System.out.println ("Student Details for " + student);
    }
}
```

3

3

Output:

Enter number of students : 1

Enter details for student 1;

Enter your name :

Valeed

Enter your USN :

13

Enter marks for 6 subjects:

Subject 1 :
98

Subject 2 :
97

Subject 3 :
96

Subject 4 :
95

Subject 5 :
94

Subject 6 :
93

Subject 7 :
92

Subject 8 :
91

Subject 9 :
90

Subject 10 :
89

Subject 11 :
88

Subject 12 :
87

Subject 13 :
86

Subject 14 :
85

Subject 15 :
84

Subject 16 :
83

Subject 17 :
82

Percentage Ans. : 87.5

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

Class Books

```
String name;
String author;
int price;
int numPages;

Books() { }

Books (String name, String author, int price,
      int numPages)
```

```
{ this.name = name;
  this.author = author;
  this.price = Price;
  this.numPages = numPages;
}
```

```
String name, Author, Price, numPages;
public void printing()
{
    System.out.println("Name : " + name);
    System.out.println("Author : " + Author);
    System.out.println("Price : " + Price);
    System.out.println("Number of Pages : " + numPages);
}
```

numPages = "number of Pages" + "numPages" + "\n";
return name + author + Price + numPages;

3. Create a class named Books with
name, author, price, numPages
as instance variables.
Create a constructor with
name, author, price, numPages
as parameters.
Create a method named printing
which prints all the information
of books.

Class Run 2

```
Public static void main (String args[])
```

{

```
Scanner s = new Scanner (System.in);  
int n;  
  
String name;  
String author;  
int price;
```

int count;

System.out.print ("Enter the number of books : ");

n = s.nextInt ();

books b [] = new books [n];

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

```
System.out.print ("Enter name of book : ");
```

z

```
System.out.print ("Enter price : ");
```

```
System.out.print ("Enter name of book : ");
```

name = s.nextLine ();

```
System.out.print ("Enter author : ");
```

author = s.nextLine ();

```
System.out.print ("Enter price : ");
```

```
price = s.nextInt ();
```

```
System.out.println ("Book added successfully");
```

```
System.out.println ("Total number of books : ");
```

```
numBooks = s.nextInt ();  
books [i] = new book (name, author, price, numBooks);
```

}

for cont : 0, <0 ; 0++)
System.out.println("book" + (i+1) +
"in " + b[i] + " by " + auth[i])

out put:
enter the number of books : 1
enter name of book : jungle book
book 1 :
enter name of book : jungle book
enter author : Rudyard Kipling
Rudyard Kipling
enter price :
enter price :

5
Enter no. of Pages : 2920.
book 1 :
book name : jungle book
Author name : Rudyard Kipling
Price : 515

Number of Pages : 2920

Jungle book
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

2019-2020

Class that contain 2 integers

1) abstract class Shape {

 double a, b;

 abstract void printArea();

}

class Rectangle extends Shape

{

 this.a = a;

 this.b = b;

 @Override

 void printArea() {

 System.out.println("Area is equal to " + (a * b));

}

class Triangle extends Shape

{

 this.a = a;

 this.b = b;

 @Override

 void printArea() {

 System.out.println("Area is equal to " + (0.5 * a * b));

}

 class Circle extends Shape

}

class Circle extends Shape

{

 this.a = a;

 @Override

 void printArea() {

 System.out.println("Area is equal to " + (3.14 * a * a));

}

Run 2

```
class Public static void main (String args[])
{
    Shape s;
```

```
s = new Rectangle(20, 30);
s.printarea();
s = new Circle(10);
s.printarea();
s = new Triangle(6, 7);
s.printarea();
```

3
150.00000000000002

?

Output:

Area is equal to 600.0
Area is equal to 314.0
Area is equal to 21.0
~~Area is equal to 314.0~~

Circle
Triangle
Rectangle
Circle
Triangle
Rectangle

21.000000000000002
314.00000000000003
600.0000000000001
314.00000000000003
21.000000000000002
600.0000000000001

21.000000000000002
314.00000000000003
600.0000000000001
314.00000000000003
21.000000000000002
600.0000000000001

Area of circle is 314.00000000000003

Area of triangle is 21.000000000000002

27 Bank maintain two kinds of account
import java.util.Scanner

Public class Account {

```
String CName;
int accNo;
String acctype;
double balance;
```

```
public Account(String CName, int accNo,
                String acctype,
                double balance){}
```

```
this.CName = CName,
```

```
this.accNo = accNo;
```

```
this.acctype = acctype;
```



```
void deposit(double amt){balance += amt;}
```

```
double t = amt;
```

```
System.out.print("Account Balance: " + balance)
```

```
System.out.print("Updated balance: " + balance);
```

3

```
void balance(){}
```

```
System.out.print("Account Balance: " + balance)
```

3

void withdraw(double amt){

```
if(balance >= amt){
```

```
balance -= amt;
```

System.out.print("Withdrawal Complete. Updated
balance: " + balance);

3 else

```
System.out.print("Insufficient funds");
```

3)

private double minBal = 1000;

private double interestRate = 5;

public CurrentAccount(string cName, int accNo,

double balance)

super(cName, accNo, "Current", balance);

System.out.println("Initial Balance : " + balance);

public void checkMinBal()

```
if (balance < minBal) {
    System.out.println("Insufficient balance");
    balance -= serviceCharge;
    System.out.println("Service Charge imposed.");
    balance -= 5;
}
```

else

```
System.out.println("Balance above minimum
                    to balance");
}
```

}

Class SavAcct extends Account {

```
double rate = 5;
System.out.println("Interest Rate : " + rate);
SavAcct( String cName, int accNo, "Savings", balance);
super(cName, accNo, "Savings", balance);
```

3

void depositInterest()

```
double interest = balance * (rate / 100);
balance += interest;
System.out.println("Interest deposited. balance : " +
```

```
balance);
System.out.println("Interest deposited. balance : " +
```

3

Class Main

Public Static void main (String [] args) {

 SA = new Sav Acct ("Varni", 123456, 1000);

 CA = new Current ("Vachan", 234567, 2000.0);

 SA . deposit (500);

 SA . balance ();

 SA . depositInterest ();

 SA . withdrawl (200);

 CA . deposit (1000);

 CA . balance ();

 CA . withdrawl (1500);

 CA . checkMinBal ();

 CA . withdrawl (100);

 CA . checkMinBal ();

}

Output

Updated balance : 500.0

Account Balance : 500.0

Interest deposited in balance is 525.0

Withdrew : Complete, Updated balance: 325.0

Updated balance : 1000.0

Account Balance is 1000.0

Insufficient funds

Balance above minimum balance

Withdrawl completed Updated balance: 900.0

Service Charge imposed. balance: 800.0

Interest deposited in balance is 825.0

Service Charge imposed. balance: 800.0

19/12/24

Threads

```
Class MessageThread extends Thread {
    String message;
    int interval;
}
```

```
MessageThread( String message, int interval) {
```

```
    this.message = message;
    this.interval = interval;
}
```

3

```
public void run() {
```

```
try {
```

```
while(true) {
```

```
System.out.println(message);
Thread.sleep(interval);
```

3-

```
} }
```

```
catch (InterruptedException e) {
```

```
e.printStackTrace();
```

```
} }
```

```
} }
```

c lang Thread Examples

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

~~MessageThread thread1 = new messageThread~~
~~"BMS College of Engineering,~~
~~10000;"~~

Message Thread Thread 2 = new Thread(fire2);
fire2.setPriority(Thread.NORM_PRIORITY + 200);

THEODORE STORCK

Head2.start();

21

2

四
卷之三

THURS

CSE

14
C5

11

三

192

卷之三

62

卷之三

卷之三

Exception

class WrongAge { Extends Father { } }

 WrongAge (String s) {
 super(s);
 }

}

class Father { String name; int age; }

Father f;

int age;

Father (int age) { throws WrongAge e;
 if (age < 0)

 throws new WrongAge ("Age cannot be negative");
 this.age = age; }

?

(Class Son Extends Father { })

Son (int sonAge, int fatherAge) {

 super (fatherAge);
 son (int fatherAge, int sonAge) { throws WrongAge e;

 super (fatherAge);

 if (sonAge >= fatherAge)

 throws new WrongAge ("Son's age should
 be less than Father's age.");
 this.sonAge = sonAge;

 else (sonAge < 0)

 throws new WrongAge ("Son's age must be positive.");

Public class Extension Inheritance Demo

public static void main(String[] args) {

```
    try {
        Father father = new Father(50),
```

```
        son = new Son(25);
```

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

```
        System.out.print("Son's age: " + son.age);
```

```
    } catch (AgeException e) {
```

```
        System.out.println("Exception caught: " +
```

```
        e.getMessage());
```

Output:

Exception: Wrong Age: Age cannot be negative!

+ son.age = 70

Wrong age: Son's age should be less than 100

Father age: 50

Son's age: 25

Exception: Wrong Age: Age cannot be negative!

Father age: 50

Son's age: 25

Exception: Wrong Age: Age cannot be negative!

Father age: 50

Son's age: 25

Exception: Wrong Age: Age cannot be negative!

lab 6:

1) Create Package

Package CTE;

Public class Personal {

 Public string vnn;

 Public string name;

 Public int sem;

 Public Personal Cstring vnn, string name, int sem;

 this . vnn = vnn;

 this . name = name;

 this . sem = sem;

}

}

import java.util.ArrayList;

Public class Internal extends Personal {

 Public ArrayList<String> markets;

 Public int internal_markets;

 Public Internal(ArrayList<String> markets)

 this . internal_markets = markets;

 }

}

Package SEM;

import CTE.Personal;

Public class External extends Personal {

 Public ArrayList<String> markets;

 Public External(ArrayList<String> markets)

 this . markets = markets;

superUser.name, sem);
this. See Marks -> See Marks;

3

Package Main:

```
import java.util.Arrays;
import CTE.internals;
import CTE.Personal;
import System.Externals;
```

Public class Main:

```
public static void main (String [] args) {
    int n = 2;
```

```
Student [] student = new Student [n];
```

for

```
for (int i = 0; i < n; i++) {
    int [] internalMarks = {80, 75, 90, 85, 91, 31};
    int [] seenMarks = {70, 80, 75, 80, 86, 31};
    Student [] o] = new Student [new Personal ("User" +
```

student[i].name, student[i].id, student[i].internalMarks]);

~~student[i] = new Student [new Internal ("User" + student[i].name, student[i].id, student[i].internalMarks));~~

~~student[i] = new External ("User" + student[i].name, student[i].id, student[i].internalMarks));~~

See Marks:

3

System.out.println ("Internal Marks: " + student[0].internalMarks);
System.out.println ("External Marks: " + student[0].externalMarks);
System.out.println ("Total Marks: " + student[0].totalMarks);
System.out.println ("Name: " + student[0].name);
System.out.println ("ID: " + student[0].id);
System.out.println ("Personal Details: " + student[0].personalDetails);
System.out.println ("Internal Marks: " + student[1].internalMarks);
System.out.println ("External Marks: " + student[1].externalMarks);
System.out.println ("Total Marks: " + student[1].totalMarks);
System.out.println ("Name: " + student[1].name);
System.out.println ("ID: " + student[1].id);
System.out.println ("Personal Details: " + student[1].personalDetails);

```
for (int i = 0; i < student.length; i++) {
    Student student = students[i];
    System.out.print ("\"Student" + student.Personal name");
    System.out.print ("\"Internal marks\" Arroyo, to string
        Student.internals .internal marks"));
    System.out.print ("\"SEEEmarks\" + Arrays.toString (
        Student .See .See marks));
}
```

```
static class Student {
    public Person Personal;
    public Internals Internals;
    public External SEE;
    public Student (Person Personal, Internals Internals,
        Person Personal);
    public Internals Internals;
}
```

Static class Student

```
    public Person Personal;
    public Internals Internals;
    public External SEE;
    public Student (Person Personal, Internals Internals,
        Person Personal);
    public Internals Internals;
```

}

Dot part

```
student = student ();
Internal Marks = [80, 75, 90, 85, 91]
SEE Marks = [70, 80, 75, 90, 86]
```

student = student ()

Internal marks [80, 75, 90, 85, 91]
SEE Marks [70, 80, 75, 90, 86]

16/10/24