**02-03-2022.**

**Variables**:

A variable is the name of the memory location with type and it can be manipulated.

There are 3 types:

1. **Local Variable**
2. **Instance Variable**
3. **Static Variable**

**Local Variable:**

* Local variables are declared inside the body of a method.
* Local variables are created when the method is entered and it will be destroyed once it exits the method.
* You can use this variable only with that method.

**Instance Variable:**

* Instance variable are declared in a class but outside the method.
* Instance variable is object specific.
* Instance variable are created when an object is created and destroyed when the object is destroyed.

**Static Variable:**

* A variable is declared with the keyword Static then it is called **Static Variable**.
* Static variable are declared in a class but outside the method.
* Static variables are initialized only once, at the start of the program execution.

**Data Types:**

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java.

* **Primitive data types:** Boolean, char, int, short, long, float, double, byte.
* **Non-primitive data types:** Classes, Arrays and Interfaces.

**Operators:**

Operator is a symbol which is used to perform operations on the data.

There are 7 types:

1. **Arithmetic operator (+,-,\*, /, %)**
2. **Logical operator (&& , || , !)**
3. **Relational operator (<, >, <=, >=, ==,!=)**
4. **Shift operator (<<, >>, >>>)**
5. **Bitwise operator (&, ^, |)**
6. **Unary operator (++, --)**
7. **Conditional operator (?, |)**
8. **Assignment operator (=, +=, -=, \*=, /=, %=, &=, ^=)**

**Key Word:**

Key words are all so called as pre-defined words which are defined in the java language.

These key words are case sensitive (**Lower Case**).

They are 49 key words and 3 are literal key words (true, false, null) are available.

**Write a simple program to add two numbers and print the output.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** A {

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

**int** x;

**int** y;

**int** z;

System.***out***.println("Enter the Numbers");

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

x = s.nextInt();

y = s.nextInt();

z = x+y;

s.close();

System.***out***.println(z);

}

}

**Output:**

Enter the Numbers

3

6

9

**Assignment:**

To take the input from the user and print the in data type format.

**package** Demo;

**import** java.util.Scanner;

**public** **class** Datatypes {

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

**int** i;

**long** l;

**short** sh;

**double** d;

**float** f;

**byte** by;

**boolean** b;

String s;

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

System.***out***.print(" Welcome ");

System.***out***.print("\n Enter the Integer\n");

i = sc.nextInt();

System.***out***.print("Enter the Long\n");

l = sc.nextLong();

System.***out***.print("Enter the Short\n");

sh = sc.nextShort();

System.***out***.print("Enter the Double\n");

d = sc.nextDouble();

System.***out***.print("Enter the Float\n");

f = sc.nextFloat();

System.***out***.print("Enter the Byte\n");

by = sc.nextByte();

System.***out***.print("Enter the Boolean\n");

b = sc.nextBoolean();

System.***out***.print("Enter the String\n");

s = sc.next();

System.***out***.print("The Integer is "+i+"\n");

System.***out***.print("The Long is "+l+"\n");

System.***out***.print("The Short is "+sh+"\n");

System.***out***.print("The Double is "+d+"\n");

System.***out***.print("The Float is "+f+"\n");

System.***out***.print("The Byte is "+by+"\n");

System.***out***.print("The Boolean is "+b+"\n");

System.***out***.print("The String is "+s+"\n");

System.***out***.print("Thank You");

}

}

**Output:**

Welcome

Enter the Integer

459

Enter the Long

25000

Enter the Short

25

Enter the Double

36.75

Enter the Float

87.98

Enter the Byte

4

Enter the Boolean

true

Enter the String

kalyan

The Integer is 459

The Long is 25000

The Short is 25

The Double is 36.75

The Float is 87.98

The Byte is 4

The Boolean is true

The String is kalyan

Thank You

**Write a program to perform Addition,Substraction,Multiplication,and Division on the values given by the user.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Calculator {

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

**int** i,j;

System.***out***.println("Enter the Numbers");

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

i = sc.nextInt();

j = sc.nextInt();

sc.close();

System.***out***.println(i+"+"+j+"="+(i+j));

System.***out***.println(i+"-"+j+"="+(i-j));

System.***out***.println(i+"\*"+j+"="+(i\*j));

System.***out***.println(i+"/"+j+"="+(i/j));

System.***out***.println(i+"%"+j+"="+(i%j));

}

}

**Output:**

Enter the Numbers

5

6

5+6=11

5-6=-1

5\*6=30

5/6=0

5%6=5

03-03-2022

**Flow of controls:**

* Java compiler executes the code from top to bottom, in the order that they appear.
* You may require executing or skipping certain set of instructions based on condition, jump to another statement, or execute a set of statements repeatedly.
* In Java, control flow statements are used to alter, redirect, or to control the flow of program execution based on the application logic.

1. **Conditional Statements (if, else, else if, nested if).**
2. **Iterative Statements (loop’s).**

**Loop’s:**

Entry Loop: While loop, For loop.

Exit Loop: Do while loop.

1. **Jumping Statements: (Switch, break).**

**Conditional Statement:**

* Conditional Statement is also called as Decision Making Statement.
* This statements decide which statement to execute and when.
* It is a Boolean Expression (**True or False**) and controls the program flow depending upon the condition provided.

**If statements:**

* Simple if
* If-else
* If-else-if ladder
* Nested if

**Simple if:**

It executes the statement, if the condition is true.

Syntax:

If (Condition){

Statement

}

**If-else:**

It is an extension to the if-statement, which uses another block of code, i.e., else block. The else block is executed if the condition of the if-block is evaluated as false.

Syntax:

If (Condition){

Statement1.

}

Else (Condition){

Statement2.

}

**If-else-if:**

The conditions will be evaluated from top to bottom. As soon as an if condition evaluates to true, then the statements associated with that if are executed, and the remaining part is bypassed. i.e. it skips checking remaining all conditions. The last most else is executed only when all conditions in the whole ladder if are false. In this last else block is optional.

Syntax:

if (condition1)

{

Statements-1;

}

else if (condition2)

{

Statements-2;

}

else if(condition3)

{

Statements-3;

}

.

.

.

else if(condition-N)

{

statements-N;

}

else

{

default statements;

}

**Checking the user is eligible for voting or not by using simple if statement.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Usingif {

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

**int** age;

System.***out***.println("Enter the Age ");

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

age = sc.nextInt();

sc.close();

**if**(age >= 18){

System.***out***.println("Eligible for VOTE: "+age);

}

**else** {

System.***out***.println("Not Eligible for VOTE");

}

}

}

**Output:**

Enter the Age

25

Eligible for VOTE: 25

Enter the Age

17

Not Eligible for VOTE

**Assignment:**

**Write a program to calculate total and average of the student subject marks then print the result as fail, pass, or first class by using if elseif.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** StdMarks {

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

**int** sno,sub1,sub2,sub3,total = 0;

**float** avg;

String sname;

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

System.***out***.println("Enter the Student Number:");

sno = sc.nextInt();

System.***out***.println("Enter the Student ");

sname = sc.next();

System.***out***.println("Enter the Subject 1 marks:");

sub1 = sc.nextInt();

System.***out***.println("Enter the Subject 2 marks:");

sub2 = sc.nextInt();

System.***out***.println("Enter the Subject 3 marks:");

sub3 = sc.nextInt();

sc.close();

total = sub1+sub2+sub3;

avg = total/3;

**if**(avg<=50) {

System.***out***.println("Student Number:"+sno);

System.***out***.println("Student Name:"+sname);

System.***out***.println("Total marks are:"+total);

System.***out***.println("FAIL"+avg);

}

**else** **if** (avg>=60 && avg<=70) {

System.***out***.println("Student Number:"+sno);

System.***out***.println("Student Name:"+sname);

System.***out***.println("Total marks are:"+total);

System.***out***.println("Average:"+avg);

System.***out***.println("PASS");

}

**else** **if** (avg>=70) {

System.***out***.println("Student Number:"+sno);

System.***out***.println("Student Name:"+sname);

System.***out***.println("Total marks are:"+total);

System.***out***.println("Average:"+avg);

System.***out***.println("FIRST CLASS");

}

}

}

**Output:**

Enter the Student Number:

459

Enter the Student

kalyan

Enter the Subject 1 marks:

72

Enter the Subject 2 marks:

73

Enter the Subject 3 marks:

71

Student Number:459

Student Name:kalyan

Total mark are:216

Average:72.0

FIRST CLASS

**04-03-2022**

**Assignment:**

**Write a program to print multiplication table for a given Number.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Tableprog {

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

**int** num,i;

System.***out***.println("Enter the Number:");

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

num = sc.nextInt();

i=1;

sc.close();

**while**(i<=15) {

System.***out***.println(num+"\*"+i+"="+(num\*i));

i++;

}

}

}

**Output:**

Enter the Number:

15

15\*1=15

15\*2=30

15\*3=45

15\*4=60

15\*5=75

15\*6=90

15\*7=105

15\*8=120

15\*9=135

15\*10=150

15\*11=165

15\*12=180

15\*13=195

15\*14=210

15\*15=225

**Write a program to print factorial of a given number.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** While2 {

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

**int** i,num,n,total=1;

System.***out***.println("Enter the Number");

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

num = sc.nextInt();

n = num;

i = 1;

sc.close();

**while**(i<n) {

total = total\*num;

i++;

num--;

}

System.***out***.println("Factoral of "+n+" is "+total);

}

}

**Output:**

Enter the Number

7

Factorial of 7 is 5040

**Write a program to print ODD OR EVEN for a given number using while loop.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** EvenOddprog {

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

**int** i,n;

System.***out***.println("Enter the number:");

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

n = sc.nextInt();

i =1;

**while** (i<=1) {

**if** (n%2==0) {

System.***out***.println("Entered number "+n+" is EVEN");

}

**else** {

System.***out***.println("Entered number "+n+" is ODD");

}

i++;

}

}

}

**Output**:

Enter the number:

6

Entered number 6 is EVEN

Enter the number:

17

Entered number 17 is ODD

**Write a program to reverse the given number by using while loop and print the output.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** ReverseOfNum {

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

**int** num,rev=0,rem,z;

System.***out***.println("Enter the number:");

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

num = sc.nextInt();

sc.close();

z=num;

**while** (num!=0) {

rem = num%10;

rev = (rev\*10)+rem;

num = num/10;

}

System.***out***.println("Reverse of "+z+" is "+rev);

}

}

**Output**:

Enter the number:

457

Reverse of 457 is 754

Enter the number:

654321

Reverse of 654321 is 123456

**Write a program to check whether the given number is Armstrong or not.**

**package** Demo;

**import** java.util.Scanner;

**import** java.lang.Math;

**public** **class** Armstrongnum

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

// **TODO** Auto-generated method stub

**int** num,temp=0,last=0,power=0,sum=0;

System.***out***.println("Enter the number:");

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

num = sc.nextInt();

temp = num;

**while** (temp>0) {

temp /=10;

power++;

}

temp=num;

**while** (temp>0) {

last = temp%10;

sum +=Math.*pow*(last, power);

temp /=10;

}

**if**(sum==num) {

System.***out***.println(num+" is an Armstrong");

}

**else** {

System.***out***.println(num+" is Not Armstrong");

}

}

}

**Output:**

Enter the number:

370

370 is an Armstrong

Enter the number:

152

152 is Not Armstrong

**Write a program to check whether the given number is Prefect Number or not.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Perfectnumwhile {

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

// **TODO** Auto-generated method stub

**int** num,i,sum=0,z;

System.***out***.println("Enter the number:");

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

num = sc.nextInt();

z=num;

i=1;

**while** (i<num) {

**if** (num%i==0) {

sum = sum+i;

}

i++;

}

**if** (sum==z) {

System.***out***.println(num+" is a Perfect Number");

}

**else** {

System.***out***.println(num+" is not a Perfect Number");

}

}

}

**Output:**

Enter the number:

6

6 is a Perfect Number

Enter the number:

7

7 is not a Perfect Number

**07-03-2022**

**For Statement:**

A **for** loop is used to repeat a specific block of code a known number of times.

**Example:**

If we want to check the grade of every student in the class, we loop from 1 to that number. When the number of time is not known before hand, we use a while loop.

**Syntax:**  
for(init;condition;incr/decr) {

// code to be executed

}

**Assignment:**

**Write a program to print a right triangle using for loop.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Forrighttriangle {

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

// **TODO** Auto-generated method stub

**int** i,j,num;

System.***out***.println("Enter the number:");

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

num = sc.nextInt();

**for** ( i = 1; i <=num; i++) {

**for** (j = 1; j <=i; j++) {

System.***out***.print("1");

}

System.***out***.println( );

}

}

}

**Output:**

Enter the number:

5

1

11

111

1111

11111

**Write a program to print numbered right triangle using for loop.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Forrighttriangle {

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

// **TODO** Auto-generated method stub

**int** i,j,num;

System.***out***.println("Enter the number:");

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

num = sc.nextInt();

**for** ( i = 1; i <=num; i++) {

**for** (j = 1; j <=i; j++) {

System.***out***.print(j+" ");

}

System.***out***.println( );

}

}

}

**Output:**

Enter the number:

6

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

1 2 3 4 5 6

**Write a program to print a down words numbered triangle for a given number using for loops.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Forrighttriangle {

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

// **TODO** Auto-generated method stub

**int** i,j,num;

System.***out***.println("Enter the number:");

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

num = sc.nextInt();

**for** ( i = num; i >=0; i--) {

**for** (j = 1; j <=i; j++) {

System.***out***.print(j+" ");

}

System.***out***.println( );

}

}

}

**Output:**

Enter the number:

6

1 2 3 4 5 6

1 2 3 4 5

1 2 3 4

1 2 3

1 2

1

**Write a program to print number triangle for a given row number using for loops.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Forrighttriangle {

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

// **TODO** Auto-generated method stub

**int** i,j,num,k=1;

System.***out***.println("Enter Row number:");

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

num = sc.nextInt();

**for** ( i = 1; i <=num; i++) {

**for** (j = 1; j <i+1; j++) {

System.***out***.print(k++ +" ");

}

System.***out***.println();

}

}

}

**Output:**

Enter Row number:

7

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

**08-03-2022**

**Nested for loop:**

A nested loop has one inside of another loop. These are typically used for working with two dimensions such as printing stars in rows and columns. When a loop is nested inside another loop, the inner loop runs many times the outer loop.

**Syntax:**

for(int row =1;row<=3;row++){

for(int col=1;col<=3;col++){

System.out.print(“inner loop”);

}

System.out.print(“outer loop”);

System.out.println();

}

**Assignment:**

**Write a program to an alphabet pattern in the output using for loops.**

**package** Demo;

**public** **class** pattern3 {

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

// **TODO** Auto-generated method stub

**int** row=5;

**int** alphabet =65;

**for** (**int** i = 0; i <=row; i++) {

**for** (**int** j = 0; j <=row-i; j++) {

System.***out***.print((**char**)(alphabet+j));

}

**for** (**int** k = 1; k <=i\*2-1; k++) {

System.***out***.print(" ");

}

**for** (**int** a = row-i; a>=0; a--) {

**if**(a!=row)

System.***out***.print((**char**)(alphabet+a));

}

System.***out***.println();

}

}

}

**Output:**

ABCDEFEDCBA

ABCDE EDCBA

ABCD DCBA

ABC CBA

AB BA

A A

**Write a program to a Pascal triangle using for loops.**

**package** Demo;

**public** **class** pascaltriangle {

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

// **TODO** Auto-generated method stub

**int** n=6;

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

**int** num=1;

System.***out***.printf("%"+(n-i)\*2 + "s", "");

**for** (**int** j = 0; j <=i; j++) {

System.***out***.printf("%4d",num);

num = num\*(i-j)/(j+1);

}

System.***out***.println();

}

}

}

**Output:**

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

**Write a program to print a right angle triangle using for loop.**

**package** Demo;

**public** **class** pattern2 {

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

// **TODO** Auto-generated method stub

**int** num=5,i,j;

**for**(i=1;i<=5;i++) {

**for** (j =1; j<=i; j++) {

System.***out***.print("\*");

}

System.***out***.println();

}

}

}

**Output:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Write a program to print a pattern using for loop.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** pattern {

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

// **TODO** Auto-generated method stub

**int** n=3;

**int** i,j;

**for** ( i = 1; i <=n ; i++) {

**for**(j=1; j<=n; j++) {

System.***out***.println(i+" "+j);

}

}

}

}

**Output:**

1 1

1 2

1 3

2 1

2 2

2 3

3 1

3 2

3 3

**Write a program to a pattern reverse triangle mirror pattern using for loops.**

**package** Demo;

**public** **class** pattern4 {

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

// **TODO** Auto-generated method stub

**int** i,j,num=5;

**for** ( i = 0; i <=num; i++) {

**for** (**int** k = 0; k <=i; k++) {

System.***out***.print(" ");

}

**for** ( j = num; j >=i; j--) {

System.***out***.print("\* ");

}

**for** (**int** m = num; m >=i; m--) {

System.***out***.print(" \*");

}

System.***out***.println();

}

}

}

**Output:**

\* \* \* \* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \*

\* \*

**09-03-2022**

**Write a program to print pascal triangle.**

**package** Demo;

**public** **class** pascalex {

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

// **TODO** Auto-generated method stub

**int** n = 9;

**for** (**int** i = 1; i<n; i++) {

**for** (**int** j = 1; j<=n-i; j++) {

System.***out***.print(" ");

}

**int** temp = 1;

**for**(**int** k = 1; k<=i; k++) {

System.***out***.print(temp+" ");

temp = temp\*(i-k)/k;

}

System.***out***.println();

}

}

}

**Output:**

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

**write a program to number pattern.**

**package** Demo;

**public** **class** pattern5 {

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

// **TODO** Auto-generated method stub

**int** n=6,m=0;

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

**for** (**int** j=n; j>=i; j--) {

System.***out***.print(" ");

}

**for** (**int** k =m ; k==i; k++) {

System.***out***.print("\*");

}

m++;

System.***out***.println();

}

}

}

**Output:**

\*

\*

\*

\*

\*

\*

\*

**Write a program to print a pattern.**

**package** Demo;

**public** **class** pattern1 {

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

// **TODO** Auto-generated method stub

**int** n =5,a=1;

**for** (**int** i = 0; i <5; i++) {

**for** (**int** j =n; j >0; j--) {

**if**(j==4 & j<=2) {

System.***out***.print(" ");

}

**else** {

System.***out***.print("K ");

}

}

n--;

System.***out***.println();

}

**for** (**int** m = 5; m >0 ; m--) {

**for** (**int** p = 0; p <a; p++) {

System.***out***.print("K ");

}

a++;

System.***out***.println();

}

}

}

**Output:**

K K K K K

K K K K

K K K

K K

K

K

K K

K K K

K K K K

K K K K K

**Write a program to print a pattern using for loops.**

**package** Demo;

**public** **class** pattern2 {

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

// **TODO** Auto-generated method stub

**int** num=5,i,j;

**for**(i=1;i<=5;i++) {

**for** (j=1; j<=num; j++) {

System.***out***.print(" \*");

}

num--;

System.***out***.println( );

}

}

}

**Output:**

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

**Write a program to print a pattern using for loops.**

**package** Demo;

**public** **class** pattern6 {

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

// **TODO** Auto-generated method stub

**int** n=6,m=0;

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

**for** (**int** j = 0; j <=i; j++) {

System.***out***.print(" ");

}

**for** (**int** k = m; k ==i; k++) {

System.***out***.print("\*");

}

m++;

System.***out***.println();

}

}

}

**Output:**

\*

\*

\*

\*

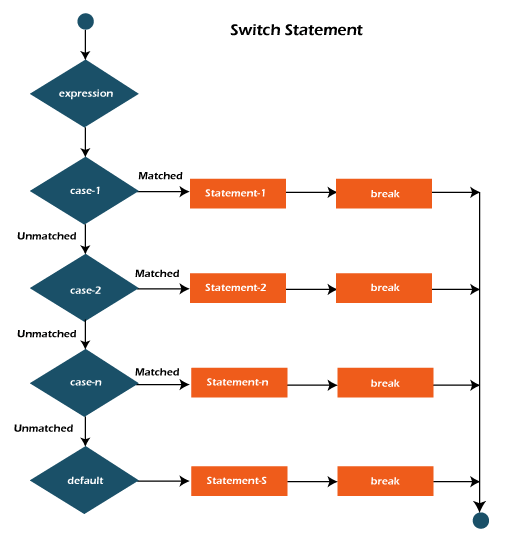
\*

\*

\*

**Switch Statement:**

* The switch statement executes one statement from multiple conditions.
* In the switch case switch condition will check with case condition1 is true then it will print statement in case1 come out from the condition when it hit break statement.
* If the condition1 is false then it will check for next case condition.



**Syntax:**

switch (Expression){

case value1:

statement.

break;

case value2:

statement.

break;

case valuen:

statement.

break;

default:

code will be executed if all the cases are not matched.

}

**Write a program to do arithmetic operation on given numbers using switch case statement.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Switcharthex {

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

// **TODO** Auto-generated method stub

**int** a,b,s;

System.***out***.println("Enter the Numbers:");

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

a = sc.nextInt();

Scanner sc1 = **new** Scanner(System.***in***);

b = sc1.nextInt();

System.***out***.println("Arithmetic operations:\n");

System.***out***.println("Select one operation:");

System.***out***.println("1-Addition.");

System.***out***.println("2-Substration.");

System.***out***.println("3-Multiplication.");

System.***out***.println("4-Division.");

s = sc.nextInt();

**switch** (s) {

**case** 1:

System.***out***.println("Addition of "+a+"&"+b+" is "+(a+b));

**break**;

**case** 2:

System.***out***.println("Substation of "+a+"&"+b+" is "+(a-b));

**break**;

**case** 3:

System.***out***.println("Multiplication of "+a+"&"+b+" is "+(a\*b));

**break**;

**case** 4:

System.***out***.println("Division of "+a+"&"+b+" is "+(a/b));

**break**;

**default**:

System.***out***.println("This is not in options: Try again.");

**break**;

}

}

}

**Output:**

Enter the Numbers:

5

6

Arithmetic operations:

Select one operation:

1-Addition.

2-Substration.

3-Multiplication.

4-Division.

3

Multiplication of 5&6 is 30

**10-03-2022**

**Do-while loop:**

In this exit loop it will first execute the statement and checks the condition which means even though the condition was false it will at least execute the program once.

**Syntax:**

do{

statement;

}

while(condition);

**Assignment:**

**Write a program to print multiplication table using do-while.**

**package** Demo;

**public** **class** dowhile {

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

// **TODO** Auto-generated method stub

**do** {

**for** (**int** i = 1; i <=10; i++) {

**for** (**int** j = 1; j <=10; j++) {

System.***out***.print(j\*i+" ");

}

System.***out***.println();

}

} **while** (**false**);

}

}

**Output:**

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

4 8 12 16 20 24 28 32 36 40

5 10 15 20 25 30 35 40 45 50

6 12 18 24 30 36 42 48 54 60

7 14 21 28 35 42 49 56 63 70

8 16 24 32 40 48 56 64 72 80

9 18 27 36 45 54 63 72 81 90

10 20 30 40 50 60 70 80 90 100

**Write a program to check whether given character is vowel or consonant using switch case statement.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** Switchex1 {

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

// **TODO** Auto-generated method stub

String ch;

System.***out***.println("Enter an Alphabet:");

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

ch =sc.next();

**switch** (ch) {

**case**"a" :

System.***out***.println(ch+" is vowel");

**break**;

**case**"e" :

System.***out***.println(ch+" is vowel");

**break**;

**case**"i" :

System.***out***.println(ch+" is vowel");

**break**;

**case**"o" :

System.***out***.println(ch+" is vowel");

**break**;

**case**"u" :

System.***out***.println(ch+" is vowel");

**break**;

**default**:

System.***out***.println(ch+" is consonant");

**break**;

}

}

}

**Output:**

Enter an Alphabet:

a

a is vowel

Enter an Alphabet:

b

b is consonant

**14-03-2022.**

**Write a program to print 1 to n multiple tables using for loops.**

**package** Pattern;

**import** java.util.Scanner;

**public** **class** Multiple Table {

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

**int** n;

System.***out***.println("Enter a Number:");

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

n = sc.nextInt();

**for** (**int** i = 1; i <=n; i++) {

**for** (**int** j = 1; j <=12; j++) {

System.***out***.println(i+"\*"+j+"="+(i\*j));

}

System.***out***.println();

}

}

}

**Output:**

Enter a Number:

11

1\*1=1

1\*2=2

1\*3=3

1\*4=4

1\*5=5

1\*6=6

1\*7=7

1\*8=8

1\*9=9

1\*10=10

1\*11=11

1\*12=12

2\*1=2

2\*2=4

2\*3=6

2\*4=8

2\*5=10

2\*6=12

2\*7=14

2\*8=16

2\*9=18

2\*10=20

2\*11=22

2\*12=24

3\*1=3

3\*2=6

3\*3=9

3\*4=12

3\*5=15

3\*6=18

3\*7=21

3\*8=24

3\*9=27

3\*10=30

3\*11=33

3\*12=36

4\*1=4

4\*2=8

4\*3=12

4\*4=16

4\*5=20

4\*6=24

4\*7=28

4\*8=32

4\*9=36

4\*10=40

4\*11=44

4\*12=48

5\*1=5

5\*2=10

5\*3=15

5\*4=20

5\*5=25

5\*6=30

5\*7=35

5\*8=40

5\*9=45

5\*10=50

5\*11=55

5\*12=60

6\*1=6

6\*2=12

6\*3=18

6\*4=24

6\*5=30

6\*6=36

6\*7=42

6\*8=48

6\*9=54

6\*10=60

6\*11=66

6\*12=72

7\*1=7

7\*2=14

7\*3=21

7\*4=28

7\*5=35

7\*6=42

7\*7=49

7\*8=56

7\*9=63

7\*10=70

7\*11=77

7\*12=84

8\*1=8

8\*2=16

8\*3=24

8\*4=32

8\*5=40

8\*6=48

8\*7=56

8\*8=64

8\*9=72

8\*10=80

8\*11=88

8\*12=96

9\*1=9

9\*2=18

9\*3=27

9\*4=36

9\*5=45

9\*6=54

9\*7=63

9\*8=72

9\*9=81

9\*10=90

9\*11=99

9\*12=108

10\*1=10

10\*2=20

10\*3=30

10\*4=40

10\*5=50

10\*6=60

10\*7=70

10\*8=80

10\*9=90

10\*10=100

10\*11=110

10\*12=120

11\*1=11

11\*2=22

11\*3=33

11\*4=44

11\*5=55

11\*6=66

11\*7=77

11\*8=88

11\*9=99

11\*10=110

11\*11=121

11\*12=132

**Write a program to print the given number is prime number or not.**

**package** Pattern;

**import** java.util.Scanner;

**public** **class** PrimeNumber {

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

**int** n,i,c=0;

System.***out***.println("Enter a Number:");

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

n = sc.nextInt();

i=1;

**while**(i<=n) {

**if**(n%i==0) {

c++;

}

i++;

}

**if**(c==2) {

System.***out***.println("prime");

}

**else** {

System.***out***.println("Not Prime");

}

}

}

**Output:**

Enter a Number:

5

prime

Enter a Number:

6

Not Prime

**Write a program to A pattern at the output.**

**package** Demo;

**public** **class** Apattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (j==0 || j==6 || i==0 || i==3) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*\*\*

\* \*

\* \*

\*\*\*\*\*\*\*

\* \*

\* \*

\* \*

**Write a program to B pattern at the output.**

**package** Demo;

**public** **class** Bpattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j<=6; j++) {

**if**(i==0 && j<=4 || i==6 && j<=4 || j==0 || i==5 && j>=6 | j<=0 || i==4 && j==6 || i==3 && j==6 || i==2 && j==6 || i==1 && j==6 ||i==3 )

{

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*

\* \*

\* \*

\*\*\*\*\*\*\*

\* \*

\* \*

\*\*\*\*\*

**Write a program to print C pattern in the output.**

**package** Demo;

**public** **class** Cpattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 || i==6 || j==0) {

System.***out***.print(" \*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*\*\*

\*

\*

\*

\*

\*

\*\*\*\*\*\*\*

**Write a program to print D pattern at the output.**

**package** Demo;

**public** **class** Dpattern {

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

// **TODO** Auto-generated method stub

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 && j<=4 || i==6 && j<=4 || j==0 || i==5 && j>=6 | j<=0 || i==4 && j==6 || i==3 && j==6 || i==2 && j==6 || i==1 && j==6) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*

\* \*

\* \*

\* \*

\* \*

\* \*

\*\*\*\*\*

**Write a program to print E pattern at the output.**

**package** Demo;

**public** **class** Epattern {

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

// **TODO** Auto-generated method stub

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 || i==6 || j==0 || i==3) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*\*\*

\*

\*

\*\*\*\*\*\*\*

\*

\*

\*\*\*\*\*\*\*

**Write a program to print F pattern at the pattern.**

**package** Demo;

**public** **class** Fpattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 || j==0|| i==3 && j<=3) {

System.out.print("\*");

}

**else** {

System.out.print(" ");

}

}

System.out.println();

}

}

}

**Output:**

\*\*\*\*\*\*\*

\*

\*

\*\*\*\*

\*

\*

\*

**Write a program to print G pattern at the output.**

**package** Demo;

**public** **class** Gpattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 && j<=5 || i==6 && j<=5 || j==0 || i==3 && j>=3 || i==4 && j==6 || i==5 && j==6 || i==1 && j==6) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*\*

\* \*

\*

\* \*\*\*\*

\* \*

\* \*

\*\*\*\*\*\*

**Write a program to print H pattern at the output.**

**package** Demo;

**public** **class** Hpattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (j==0 || j==6 || i==3) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\* \*

\* \*

\* \*

\*\*\*\*\*\*\*

\* \*

\* \*

\* \*

**Write a program to print I pattern at the output.**

**package** Demo;

**public** **class** Ipattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 || i==6 || j==3) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*\*\*

\*

\*

\*

\*

\*

\*\*\*\*\*\*\*

**Write a program to print J pattern at the output.**

**package** Demo;

**public** **class** Jpattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 || j==3 || i==6 && j==2 || i==6 && j==1 || i==5 && j==0 || i==4 && j==0) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*\*\*

\*

\*

\*

\* \*

\* \*

\*\*\*

**Write a program to print K pattern at the output.**

**package** Demo;

**public** **class** Kpattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (j==0 || i==0 && j==6 || i==1 && j==4 || i==2 && j==2 || i==3 && j==1 || i==4 && j==2 || i==5 && j==4 || i==6 && j==6 ) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\* \*

\* \*

\* \*

\*\*

\* \*

\* \*

\* \*

**Write a program to print L pattern at the output.**

**package** Demo;

**public** **class** Lpattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==6 || j==0) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*

\*

\*

\*

\*

\*

\*\*\*\*\*\*\*

**Write a program to print M pattern at the output.**

**package** Demo;

**public** **class** Mpattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (j==0 || j==6 || i==1 && j==2 | j==4 || i==2 && j==3) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\* \*

\* \* \* \*

\* \* \*

\* \*

\* \*

\* \*

\* \*

**Write a program to print N pattern at the output.**

**package** Demo;

**public** **class** Npattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (j==0 || j==6 || i==1 && j==1 || i==2 && j==2 || i==3 && j==3 || i==4 && j==4 || i==5 && j==5) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

**}**

**Output:**

\* \*

\*\* \*

\* \* \*

\* \* \*

\* \* \*

\* \*\*

\* \*

**Write a program to print O pattern at the output.**

**package** Demo;

**public** **class** Opattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 && j<=5 && j>=1 || i==6 && j<=5 && j>=1 || j==0 && i<=5 && i>=1 || j==6 && i<=5 && i>=1) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*

\* \*

\* \*

\* \*

\* \*

\* \*

\*\*\*\*\*

**Write a program to print P pattern at the output.**

**package** Demo;

**public** **class** Pnattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 && j<=5 || j==0 || i==3 && j<=5 || i==1 && j==6 || i==2 && j==6) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*\*

\* \*

\* \*

\*\*\*\*\*\*

\*

\*

\*

**Write a program to print Q pattern at the output.**

**package** Demo;

**public** **class** Qpattern {

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

// **TODO** Auto-generated method stub

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 && j<=5 && j>=1 || i==6 && j<=4 && j>=1 || j==0 && i<=5 && i>=1 || j==6 && i<=5 && i>=1 || i==6 && j==6 || i==5 && j==4) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*

\* \*

\* \*

\* \*

\* \*

\* \* \*

\*\*\*\* \*

**Write a program to print S pattern at the output.**

**package** Demo;

**public** **class** Spattern {

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

**for** (**int** i = 0; i <=6; i++) {

**for** (**int** j = 0; j <=6; j++) {

**if** (i==0 && j>=1 & j<=5 || i==6 && j>=0 & j<=5 || j==0 && i<=2 || i==3 && j>=1 & j<=5 || j==6 && i>=4 & i<=5) {

System.***out***.print("\*");

} **else** {

System.***out***.print(" ");

}

}

System.***out***.println();

}

}

}

**Output:**

\*\*\*\*\*\*

\*

\*

\*\*\*\*\*

\*

\*

\*\*\*\*\*\*

**Access Modifiers:**

* The access modifier in java specifies the accessibility of a field, method, constructor, or class.
* We can change the access level of field, constructor, method and class by applying the access modifier in it.

They are four types of java modifier:

1. Private modifier
2. Public modifier
3. Protected modifier
4. Default modifier

**Private Modifier:**

* Private modifiers can only be used with in the class.
* It cannot be accessed by outside the class.

**Public Modifier:**

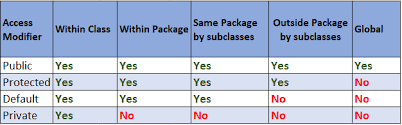
* Public modifiers can be used anywhere.
* It can be accessed from within the class, outside the class, within the package and outside the package.

**Protected Modifier:**

* Protected modifiers can be used within the package and outside the package through child class.
* If you don’t make child class, it cannot be accessed outside the package.

**Default Modifier:**

* Default modifier can be used within the package.
* It cannot be accessed from outside the package.
* It we do not specify any modifier it will be considered as default.

****

**15-03-2022.**

**Assignment:**

**1.Write a program to print prime numbers 1 to 100.**

**package** Demo;

**public** **class** PrimeNumber {

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

**int** i,j,c,p=0;

System.***out***.println("Prime Numbers from 1 to 100:");

**for** ( i = 1; i <=100; i++) {

c=0;

**for** (j = i; j>=1; j--) {

**if** (i%j==0) {

c++;

}

}

**if**(c==2) {

System.***out***.print(i+" ");

}

}

}

}

**Output:**

Prime Numbers from 1 to 100:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

**2.Write a program for basic calculator and take inputs from user.**

1. **Enter the calculator print the operations and the user should select an operation.**
2. **Here user can do any operation until he enters the exit button.**
3. **When he enters the exit button the calculator should be terminated.**

**package** Demo;

**import** java.util.Scanner;

**public** **class** BasicCalculator {

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

**int** a,b,c;

**int** i=0,n=0;

**while** (i<=n) {

System.***out***.println("Enter the numbers:");

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

System.***out***.println("Enter the 'a' value:");

a=sc.nextInt();

System.***out***.println("Enter the 'b' value:");

b=sc.nextInt();

System.***out***.println("Select an operation:");

System.***out***.println("1.Addiction.\n2.Substraction.\n3.Multiplication.\n4.Division.\n5.Exit.");

c=sc.nextInt();

**switch** (c) {

**case** 1:

System.***out***.println("Addiction of "+a+" and "+b+" is "+(a+b));

**break**;

**case** 2:

System.***out***.println("Subtraction of "+a+" and "+b+" is "+(a-b));

**break**;

**case** 3:

System.***out***.println("Multiplication of "+a+" and "+b+" is "+(a\*b));

**break**;

**case** 4:

System.***out***.println("Division of "+a+" and "+b+" is "+(a/b));

**break**;

**case** 5:

System.***out***.println("Exited");

n = 0;

**break**;

**default**:

System.***out***.println("Operation does not existed");

**break**;

}

i++;

n++;

}

}

}

**Output:**

Enter the numbers:

Enter the 'a' value:

36

Enter the 'b' value:

6

Select an operation:

1.Addiction.

2.Substraction.

3.Multiplication.

4.Division.

5.Exit.

4

Division of 36 and 6 is 6

Enter the numbers:

Enter the 'a' value:

7

Enter the 'b' value:

8

Select an operation:

1.Addiction.

2.Substraction.

3.Multiplication.

4.Division.

5.Exit.

3

Multiplication of 7 and 8 is 56

Enter the numbers:

Enter the 'a' value:

3

Enter the 'b' value:

5

Select an operation:

1.Addiction.

2.Substraction.

3.Multiplication.

4.Division.

5.Exit.

5

Exited

**3.Write a program to print Fibonacci series up to n .**

**package** Demo;

**import** java.util.Scanner;

**public** **class** FibonacciSeries {

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

**int** i,a=0,b=1,c,n;

System.***out***.println("Enter a Number:");

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

n = sc.nextInt();

**for** (i = 1; i <=n; i++) {

c=a+b;

a=b;

b=c;

System.***out***.println(a);

}

}

}

**Output:**

Enter a Number:

10

1

1

2

3

5

8

13

21

34

55

**4.Write a program to check the entered number is palindrome or not ?**

**package** Demo;

**import** java.util.Scanner;

**public** **class** PalindromeNumber {

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

**long** n,rem,rev=0,temp;

System.***out***.println("Enter the number:");

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

n =sc.nextLong();

temp=n;

**while**(n!=0) {

rem = n%10;

rev = (rev \* 10)+rem;

n = n/10;

}

System.***out***.println(rev);

**if**(rev==temp) {

System.***out***.println("Given number is palindrome");

}

**else** {

System.***out***.println("not palindrome");

}

}

}

**Output:**

Enter the number:

220022

220022

Given number is palindrome

Enter the number:

220033

330022

not palindrome

**16-03-2022.**

**Assignment:**

**Write a program with no arguments and with no return value.**

**package** functions;

**import** java.util.Scanner;

**class** Add {

**int** idno;

String name;

**long** phoneno;

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

**public** **void** read\_data() {

System.***out***.println("Enter the idno,name,phoneno:");

idno = sc.nextInt();

name = sc.next();

phoneno = sc.nextLong();

}

**public** **void** print\_data() {

System.***out***.println("Idno:"+idno);

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

System.***out***.println("Phoneno:"+phoneno);

}

}

**public** **class** NoArgsNoReturn {

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

Add ad = **new** Add();

ad.read\_data();

ad.print\_data();

}

}

**Output:**

Enter the idno,name,phoneno:

459

Kalyan

8374767245

Idno:459

Name:Kalyan

Phoneno:8374767245

**Write a program with arguments and with return values.**

**package** functions;

**import** java.util.Scanner;

**class** Calculator{

**int** a,b;

**public** **int** add(**int** a, **int** b) {

**int** sum =a+b;

**return** sum;

}

**public** **int** sub(**int** a,**int** b) {

**int** total = a-b;

**return** total;

}

}

**public** **class** WithArgsWithReturnValue {

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

**int** n1,n2,result;

System.***out***.println("Enter two numbers:");

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

n1 = sc.nextInt();

n2 = sc.nextInt();

Calculator cl = **new** Calculator();

result = cl.add(n1, n2);

System.***out***.println(result);

result = cl.sub(n1, n2);

System.***out***.println(result);

}

}

**Output:**

Enter two numbers:

5

9

Addition is:14

Subtraction is:-4

**Write a program with arguments and with no return values.**

**package** functions;

**import** java.util.Scanner;

**class** Details {

**int** idno;

String name;

**float** salary;

**public** **void** read\_data(**int** a,String b,**float** c) {

idno = a;

name = b;

salary = c;

}

**public** **void** print\_data() {

System.***out***.println("Idno:"+idno);

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

System.***out***.println("Salary:"+salary);

}

}

**public** **class** ArgsNoReturn {

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

**int** x;

String y;

**float** z;

System.***out***.println("Enter the idno,name,salary:");

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

x = sc.nextInt();

y = sc.next();

z = sc.nextFloat();

Details d = **new** Details();

d.read\_data(x, y, z);

d.print\_data();

}

}

**Output:**

Enter the idno,name,salary:

456

venu

15000

Idno:456

Name:venu

Salary:15000.0

**17-03-2022.**

**Array:**

It is a variable which call’s the sequence of elements of similar data type.

They are two types:

* **One dimensional array.**
* **Two or multi-dimensional array.**

**One dimensional array:**

**Syntax:**

Data type variable name [] = new data type [SIZE].

**Ex:**

Int a[] = new int[10];

**Two or multi-dimensional array:**

**Syntax:**

Data type variable name[][] = new data type [SIZE][SIZE].

**Ex:**

Int a[][] = new int[3][3].

Matrix format.

1 2 3

4 5 6

7 8 9

**Write a program to print one dimensional array by taking input values from the user.**

**package** arrays;

**import** java.util.Scanner;

**class** Array{

**int** i;

**int** a[] = **new** **int**[10];

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

**public** **void** read() {

**for** ( i = 0; i <10; i++) {

a[i] = sc.nextInt();

}

}

**public** **void** show() {

System.***out***.println("Output:");

**for** (**int** i = 0; i <10; i++) {

System.***out***.println(a[i]);

}

}

}

**public** **class** OneDimArray {

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

System.***out***.println("One Dimensional Array");

Array oa = **new** Array();

oa.read();

oa.show();

}

}

**Output:**

One Dimensional Array

1

2

3

4

5

6

7

8

9

10

Output:

1

2

3

4

5

6

7

8

9

10

**Write a program to print reverse numbers of array.**

**package** arrays;

**import** java.util.Scanner;

**class** Ar{

**int** i;

**int** a[] = **new** **int** [10];

**public** **void** read\_n() {

System.***out***.println("Enter the numbers:");

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

**for** (**int** i = 0; i <10; i++) {

a[i] = sc.nextInt();

}

}

**public** **void** show\_n() {

System.***out***.println("One Dimensional Array Revers Of Numbers:");

**for** (**int** i = 9; i>=0; i--) {

System.***out***.println(a[i]);

}

}

}

**public** **class** OneDimeArrayReversNumber {

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

System.***out***.println("One Dimensional Array");

Ar an = **new** Ar();

an.read\_n();

an.show\_n();

}

}

**Output:**

One Dimensional Array

Enter the numbers:

10

9

8

7

6

5

4

3

2

1

One Dimensional Array Revers Of Numbers:

1

2

3

4

5

6

7

8

9

10

**Write a program to search the element in the array and print the position of the element.**

**package** arrays;

**import** java.util.Scanner;

**class** Element{

**int** i ;

**int** a[] = **new** **int**[10];

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

**public** **void** read\_e() {

System.***out***.println("Enter the Numbers:");

**for** ( i = 0; i <10; i++) {

a[i] = sc.nextInt();

}

}

**public** **void** show\_e() {

System.***out***.println("Enter the number you want to search:");

**int** t = sc.nextInt();

**for** ( i = 0; i <10; i++) {

**if** (a[i]==t) {

System.***out***.println("The Number is "+a[i]+" and the position of the numbers is "+i+".");

}

}

}

}

**public** **class** SearchElementInArray {

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

System.***out***.println("One Dimensional Array:");

Element se = **new** Element();

se.read\_e();

se.show\_e();

}

}

**Output:**

One Dimensional Array:

Enter the Numbers:

78

45

21

13

49

85

68

71

92

53

Enter the number you want to search:

49

The Number is 49 and the position of the numbers is 4.

**Write a program to print sum of even index numbers of array and sum of odd index numbers of array.**

**package** arrays;

**import** java.util.Scanner;

**class** EvenOdd{

**int** i,sume,sumo;

**int** a[] = **new** **int**[10];

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

**public** **void** read\_eo() {

System.***out***.println("Enter the Numbers:");

**for** (i = 0; i <10; i++) {

a[i] = sc.nextInt();

}

}

**public** **void** SumEven() {

**for** (i = 0; i <10; i++) {

**if** (a[i]%2==0) {

sume += a[i];

} **else** {

sumo +=a[i];

}

}

System.***out***.println("Sum of Even Numbers:"+sume);

System.***out***.println("Sum of Odd Numbers:"+sumo);

}

}

**public** **class** SumEvenOdd {

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

System.***out***.println("One Dimensional Array:");

EvenOdd eo = **new** EvenOdd();

eo.read\_eo();

eo.SumEven();

}

}

**Output:**

One Dimensional Array:

Enter the Numbers:

1

2

3

4

5

6

7

8

9

10

Sum of Even Numbers:30

Sum of Odd Numbers:25

**Write a program to print Ascending and Descending order of given numbers in array.\**

**package** arrays;

**import** java.util.Scanner;

**class** AscDes{

**int** i,j,temp;

**int** a[] = **new** **int**[10];

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

**public** **void** read\_ad() {

System.***out***.println("Enter the numbers: ");

**for** (i = 0; i <10; i++) {

a[i] = sc.nextInt();

}

}

**public** **void** Ascending() {

System.***out***.println("Ascending Order:");

**for** (i = 0; i <10; i++) {

**for** (j = i+1; j <10; j++) {

**if** (a[i]>a[j]) {

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

**for** (j = 0; j <10; j++) {

System.***out***.print(a[j]+",");

}

System.***out***.println();

}

**public** **void** Descending () {

System.***out***.println("Descending Order:");

**for** (i = 0; i <10; i++) {

**for** (**int** k =i+1; k<10; k++) {

**if** (a[i]<a[k]) {

temp=a[i];

a[i]=a[k];

a[k]=temp;

}

}

}

**for** (**int** k = 0; k <10; k++) {

System.***out***.print(a[k]+",");

}

}

}

**public** **class** AscendingDescendingOrder {

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

System.***out***.println("One Dimensional Array:");

AscDes ad = **new** AscDes();

ad.read\_ad();

ad.Ascending();

ad.Descending();

}

}

**Output:**

One Dimensional Array:

Enter the numbers:

52

12

86

45

79

96

31

5

2

34

Ascending Order:

2,5,12,31,34,45,52,79,86,96,

Descending Order:

96,86,79,52,45,34,31,12,5,2,

**21-03-2022.**

**Assignment:**

**Two Dimensional Arrays:**

**Write a program to print matrix by using two dimensional array.**

**package** arrays;

**import** java.util.Scanner;

**class** TwoDimArray{

**int** i,j;

**int** a[][] = **new** **int**[3][3];

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

**public** **void** read\_td() {

System.***out***.println("Enter the numbers:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

a[i][j] = sc.nextInt();

}

}

}

**public** **void** show\_td() {

System.***out***.println("The Matrix is:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

System.***out***.print(a[i][j]+" ");

}

System.***out***.println();

}

}

}

**public** **class** TwoDimensionalArray {

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

System.***out***.println("Two Dimensional Array:");

TwoDimArray tda = **new** TwoDimArray();

tda.read\_td();

tda.show\_td();

}

}

**Output:**

Two Dimensional Array:

Enter the numbers:

1

2

3

4

5

6

7

8

9

The Matrix is:

1 2 3

4 5 6

7 8 9

**Write a program print Diagonal matrix by using two dimensional array.**

**package** arrays;

**import** java.util.Scanner;

**class** Diagonal{

**int** i,j;

**int** a[][] = **new** **int**[3][3];

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

**public** **void** read\_dia() {

System.***out***.println("Enter the Numbers:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

a[i][j] = sc.nextInt();

}

}

}

**public** **void** show\_dia() {

System.***out***.println("The Diagonal Matrix is:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

**if** (i==j) {

a[i][j] = a[i][j];

System.***out***.print(" "+a[i][j]);

}

**else** {

a[i][j]=0;

System.***out***.print(" "+a[i][j]);

}

}

System.***out***.println();

}

}

}

**public** **class** DiagonalMatrix {

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

System.***out***.println("The Two Dimensional Array:");

Diagonal dar = **new** Diagonal();

dar.read\_dia();

dar.show\_dia();

}

}

**Output:**

The Two Dimensional Array:

Enter the Numbers:

1

2

3

4

5

6

7

8

9

The Diagonal Matrix is:

1 0 0

0 5 0

0 0 9

**Write a program to print right triangle of matrix by using two dimensional array.**

**package** arrays;

**import** java.util.Scanner;

**class** Right\_tri{

**int** i,j;

**int** a[][] = **new** **int**[3][3];

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

**public** **void** read\_tri() {

System.***out***.println("Enter the Numbers:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

a[i][j] = sc.nextInt();

}

}

}

**public** **void** show\_tri() {

System.***out***.println("The Right Triangle Matrix:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j<3; j++) {

**if** (j<=i) {

a[i][j]=a[i][j];

System.***out***.print(" "+a[i][j]);

}

**else** {

a[i][j]=0;

System.***out***.print(" "+a[i][j]);

}

}

System.***out***.println();

}

}

}

**public** **class** TriangleMatrixArray {

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

System.***out***.println("The Two Dimensional Array:");

Right\_tri rt = **new** Right\_tri();

rt.read\_tri();

rt.show\_tri();

}

}

**Output:**

The Two Dimensional Array:

Enter the Numbers:

1

2

3

4

5

6

7

8

9

The Right Triangle Matrix:

1 0 0

4 5 0

7 8 9

**Write a program to print other half of the right triangle by using two dimensional array.**

**package** arrays;

**import** java.util.Scanner;

**class** Triangle{

**int** i,j;

**int** a[][] = **new** **int**[3][3];

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

**public** **void** read\_tri() {

System.***out***.println("Enter the Numbers:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

a[i][j] = sc.nextInt();

}

}

}

**public** **void** show\_tri() {

System.***out***.println("The Right Triangle Matrix:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j<3; j++) {

**if** (j>i) {

a[i][j]=a[i][j];

System.***out***.print(" "+a[i][j]);

}

**else** {

a[i][j]=0;

System.***out***.print(" "+a[i][j]);

}

}

System.***out***.println();

}

}

}

**public** **class** TriangleMatrixArray2 {

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

System.***out***.println("The Two Dimensional Array:");

Triangle tri = **new** Triangle();

tri.read\_tri();

tri.show\_tri();

}

}

**Output:**

The Two Dimensional Array:

Enter the Numbers:

1

2

3

4

5

6

7

8

9

The Right Triangle Matrix:

0 2 3

0 0 6

0 0 0

**Write a program to print upper triangle of the matrix by using two dimensional array.**

**package** arrays;

**import** java.util.Scanner;

**class** Upper\_tri{

**int** i,j;

**int** a[][] = **new** **int**[3][3];

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

**public** **void** read\_upp() {

System.***out***.println("Enter the Numbers:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

a[i][j] =sc.nextInt();

}

}

}

**public** **void** show\_upp() {

System.***out***.println("Upper Triangle Matrix:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j<3; j++) {

**if** (j>=i) {

a[i][j] = a[i][j];

System.***out***.print(" "+a[i][j]);

}

**else** {

a[i][j] =0;

System.***out***.print(" "+a[i][j]);

}

}

System.***out***.println();

}

}

}

**public** **class** UpperTriangleMatrixArray {

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

System.***out***.println("The Two Dimensional Array:");

Upper\_tri upt = **new** Upper\_tri();

upt.read\_upp();

upt.show\_upp();

}

}

**Output:**

The Two Dimensional Array:

Enter the Numbers:

1

2

3

4

5

6

7

8

9

Upper Triangle Matrix:

1 2 3

0 5 6

0 0 9

**Write a program to upper triangle matrix other half by two dimensional array.**

**package** arrays;

**import** java.util.Scanner;

**class** Upper\_tr{

**int** i,j;

**int** a[][] = **new** **int**[3][3];

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

**public** **void** read\_upp() {

System.***out***.println("Enter the Numbers:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

a[i][j] = sc.nextInt();

}

}

}

**public** **void** show\_upp() {

System.***out***.println("The Diagonal Matrix is:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

**if** (j<i) {

a[i][j] = a[i][j];

System.***out***.print(" "+a[i][j]);

}

**else** {

a[i][j]=0;

System.***out***.print(" "+a[i][j]);

}

}

System.***out***.println();

}

}

}

**public** **class** UpperTriangleMatrixArray1 {

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

System.***out***.println("Two Dimensional Array:");

Upper\_tr up = **new** Upper\_tr();

up.read\_upp();

up.show\_upp();

}

}

**Output:**

Two Dimensional Array:

Enter the Numbers:

1

2

3

4

5

6

7

8

9

The Diagonal Matrix is:

0 0 0

4 0 0

7 8 0

**Write a program to print transpose matrix for a given matrix by using two dimensional array.**

**package** arrays;

**import** java.util.Scanner;

**class** Transpose{

**int** i,j;

**int** a[][] = **new** **int**[3][3];

**int** b[][] = **new** **int**[3][3];

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

**public** **void** read\_tra() {

System.***out***.println("Enter the element ");

**for** (i = 0; i< 3; i++) {

**for** (j = 0; j< 3; j++) {

a[i][j] = sc.nextInt();

}

}

}

**public** **void** show\_tra() {

System.***out***.println("The Transpose Matrix is:");

**for** (i = 0; i < 3; i++) {

**for** (j = 0; j < 3; j++) {

b[i][j] = a[j][i];

System.***out***.print(" "+b[i][j]);

}

System.***out***.println();

}

}

}

**public** **class** TransposeMatrixArray {

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

System.***out***.println("Two Dimensional Array:");

Transpose tp = **new** Transpose();

tp.read\_tra();

tp.show\_tra();

}

}

**Output:**

Two Dimensional Array:

Enter the element

1

2

3

4

5

6

7

8

9

The Transpose Matrix is:

1 4 7

2 5 8

3 6 9

**Write a program to add two matrix’s by two dimensional array.**

**package** arrays;

**import** java.util.Scanner;

**class** Add\_Matrix{

**int** a[][] = **new** **int**[3][3];

**int** b[][] = **new** **int**[3][3];

**int** c[][] = **new** **int**[3][3];

**int** i,j;

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

**public** **void** read\_add() {

System.***out***.println("Enter the Elements for First Matrix:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

a[i][j] = sc.nextInt();

}

}

System.***out***.println("Enter the Elements for Second Matrix:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

b[i][j] = sc.nextInt();

}

}

}

**public** **void** show\_add() {

**int** c[][] = **new** **int**[3][3];

System.***out***.println("Addition of Two matrix is:");

**for** (i = 0; i <3; i++) {

**for** (j = 0; j <3; j++) {

c[i][j] = a[i][j]+b[i][j];

System.***out***.print(c[i][j]+" ");

}

System.***out***.println();

}

}

}

**public** **class** AdditionMatrix {

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

Add\_Matrix am = **new** Add\_Matrix();

am.read\_add();

am.show\_add();

}

}

**Output:**

Enter the Elements for First Matrix:

2

7

4

9

7

8

5

6

1

Enter the Elements for Second Matrix:

10

17

21

29

15

18

12

11

22

Addition of Two matrix is:

12 24 25

38 22 26

17 17 23

**Write a program to multiple two matrix by using two dimensional array.**

**package** arrays;

**import** java.util.Scanner;

**class** Mul\_Matrix{

**int** a[][] = **new** **int**[3][3];

**int** b[][] = **new** **int**[3][3];

**int** c[][] = **new** **int**[3][3];

**int** i,j;

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

**public** **void** read\_add() {

System.***out***.println("Enter the Elements for First Matrix:");

**for** (i = 0; i<3; i++) {

**for** (j = 0; j <3; j++) {

a[i][j] = sc.nextInt();

}

}

System.***out***.println("Enter the Elements for Second Matrix:");

**for** (i = 0; i<3; i++) {

**for** (j = 0; j <3; j++) {

b[i][j] = sc.nextInt();

}

}

}

**public** **void** show\_add() {

**int** c[][] = **new** **int**[3][3];

System.***out***.println("Addition of Two matrix is:");

System.***out***.println();

**for** (i = 0; i<3; i++) {

**for** (j = 0; j<3; j++) {

**for**(**int** k =0; k<3;k++) {

c[i][j] += a[i][k]\*b[k][j];

}

System.***out***.print(c[i][j]+" ");

}

System.***out***.println();

}

}

}

**public** **class** MultipleMatrix {

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

Mul\_Matrix mm = **new** Mul\_Matrix();

mm.read\_add();

mm.show\_add();

}

}

**Output:**

Enter the Elements for First Matrix:

10

12

16

20

25

29

32

37

30

Enter the Elements for Second Matrix:

1

9

7

6

3

4

5

2

8

Addition of Two matrix is:

162 158 246

315 313 472

404 459 612

**Method overloading :**

If a class has multiple methods having same name but different in parameters.it is a known as Method overloading.

**Method overriding:**

If subclass(Child class) has the same method as declared in the parent class it is known as Method Overrid

ing.

**24-03-2022.**

**Polymorphism:**

* Polymorphism is used to perform single action in different ways. The word poly means MANY, morphi means FORM.
* Polymorphism has many forms. In other words polymorphism is allows us to defined one interface and multiple implementations.
* In java polymorphism mainly divided into 2 types.

1. Compile time polymorphism.
2. Run time polymorphism.

**Compile time polymorphism:**

* Compile time polymorphism is known as static polymorphism.
* This type polymorphism is achieved through method Over Loading.
* Over loading in java is example for compile time polymorphism.

**Method Overloading:**

When there are multiple functions with same name and same method name but different parameter.

or

Same method name with different parameters is knows as method over loading.

**Method Overloading Programs:**

**package** polimorphism;

**class** OverLoading{

**public** **static** **int** add(**int** a,**int** b) {

System.***out***.println("int and int");

**int** s=a+b;

**return** s;

}

**public** **static** **double** add(**double** p,**double** q) {

System.***out***.println("double and double");

**return** p+q;

}

}

**public** **class** MethodOverLoading {

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

**double** x=10;

**double** y=18.6;

**int** result;

System.***out***.println("Method Over Loading:");

OverLoading ol = **new** OverLoading();

result = ol.*add*(11,11);

System.***out***.println(result);

System.***out***.println(ol.*add*(x, y));

}

}

**Output:**

Method Over Loading:

int and int

22

double and double

28.6

**package** polimorphism;

**class** Addition{

**public** **static** **int** add(**int** a,**int** b) {

System.***out***.println("Addition Of Two numbers:");

**return** a+b;

}

**public** **static** **int** add(**int** a,**int** b,**int** c) {

System.***out***.println("Addition Of Three Numbers:");

**return** a+b+c;

}

}

**public** **class** MethodOverLoading1 {

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

System.***out***.println("Method Overloading:");

Addition ad = **new** Addition();

System.***out***.println(ad.*add*(12,42));

System.***out***.println(ad.*add*(6,72,180));

}

}

**Output:**

Method Overloading:

Addition Of Two numbers:

54

Addition Of Three Numbers:

258

**package** polimorphism;

**class** Add{

**public** **static** **double** sum(**int** a ,**double** b) {

System.***out***.println("int And double:");

**return** a+b;

}

**public** **static** **double** sum(**double** a ,**double** b) {

System.***out***.println("double and double");

**return** a+b;

}

**public** **static** **double** sum(**double** a ,**int** b) {

System.***out***.println("double and int");

**return** a+b;

}

}

**public** **class** MethodOverLoading2 {

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

System.***out***.println("Method Overloading:");

Add ma = **new** Add();

System.***out***.println(ma.*sum*(34,45.23));

System.***out***.println(ma.*sum*(16.89,56.43));

System.***out***.println(ma.*sum*(56.13,45));

}

}

**Output:**

Method Overloading:

int And double:

79.22999999999999

double and double

73.32

double and int

101.13

**package** polimorphism;

**class** Char{

**public** **static** **int** Cadd(**int** a,**char** ch) {

System.***out***.println("int and char");

**return** a+ch;

}

**public** **static** **int** Cadd(**char** a,**char** ch) {

System.***out***.println("char and char");

**return** a+ch;

}

**public** **static** **int** Cadd(**int** a,**int** ch) {

System.***out***.println("int and int");

**return** a+ch;

}

**public** **static** **int** Cadd(**char** a,**int** ch) {

System.***out***.println("char and int");

**return** a+ch;

}

}

**public** **class** MethodOverLoading3 {

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

System.***out***.println("Method Overloadind:");

Char ca = **new** Char();

System.***out***.println(ca.*Cadd*(1, 'k'));

System.***out***.println(ca.*Cadd*('a', 4));

System.***out***.println(ca.*Cadd*(17, 4));

System.***out***.println(ca.*Cadd*('s', 'k'));

}

}

**Output:**

Method Overloadind:

int and char

108

char and int

101

int and int

21

char and char

222

**28-03-2022.**

**Method Overriding:**

If the child class as the same method declared in a parent class is known as method overriding.

It is used for run time polymorphism.

**Rules for java Method overriding:**

* The method must have same name as in the parent class.
* The method must have same parameters as in the parent class.
* There must be an IS-A relationship (inheritance).

**Advantage:**

The main advantage of method overriding is that the class can give its own specific implementation to inherited method without even modifying the parent class code.

**Assignment:**

**Method Overriding:**

**package** polimorphism;

**class** Parent{

**public** **void** add() {

System.***out***.println("Parent class ");

}

}

**class** Child **extends** Parent{

@Override

**public** **void** add() {

System.***out***.println("Child Class");

}

}

**public** **class** MethodOverriding {

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

Parent p = **new** Parent();

p.add();

Child obj1 = **new** Child();

obj1.add();

}

}

**Output:**

Parent class

Child Class

**Method Overriding:**

**package** polimorphism;

**class** Vehicle{

**public** **void** run() {

System.***out***.println("Vehicle is Running.");

}

}

**class** Bike **extends** Vehicle{

@Override

**public** **void** run() {

//First Preference to class

System.***out***.println("Bike is Running.");

}

}

**public** **class** MethodOverriding1 {

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

Bike bk = **new** Bike();

bk.run();

}

}

**Output:**

Bike is Running.

**Write a program for ATM machine.**

package Demo;

import java.util.Scanner;

class Bank{

long AccNum=1234567891;

String Name = "seera kalyan";

double amount = 500000.00;

int pin = 4590;

String Branch = "Donparty Branch";

String Bank\_name = "AXIS";

double wd,amt;

Scanner sc = new Scanner(System.in);

public void Balance() {

System.out.println("Enter the pin:");

int p = sc.nextInt();

if(p==pin) {

System.out.println("Account Balance is "+amount);

}

else {

System.out.println("Entered a wrong pin.\nTry Again ");

}

}

public void withdraw() {

System.out.println("Enter the pin:");

int p = sc.nextInt();

if (p==pin) {

System.out.println("Enter the Amount:");

wd = sc.nextDouble();

if (wd<amount) {

amount =amount - wd;

System.out.println("The Amount is Withdrawed.");

System.out.println("The Remaining Balance is "+amount);

}

else {

System.out.println("In suffacient Balance.");

}

}

else {

System.out.println("Invailed Pin");

}

}

public void deposite() {

System.out.println("Enter the Account Number:");

long Acc = sc.nextLong();

if (Acc==AccNum) {

System.out.println("Enter the Pin:");

int p = sc.nextInt();

if (p==pin) {

System.out.println("Enter the Amount:");

amt= sc.nextDouble();

amount =amount + amt;

System.out.println("The Total Balance is "+amount);

} else {

System.out.println("Invalid Pin");

}

}else {System.out.println("Invaild Account Number.");

}

}

public void detailes() {

System.out.println("Enter the Pin:");

int p = sc.nextInt();

if (p==pin) {

System.out.println("Account No:"+AccNum+"\nAcc Houlder Name:"+Name+"\nBranch:"+Branch+"\nBank Name:"+Bank\_name+"Balance:"+amount);

}else {System.out.println("Invaild Pin");

}

}

}

public class BankingProgram {

public static void main(String[] args) {

int i,num=0,n=1;

Scanner s = new Scanner(System.in);

System.out.println("Welcome to the ATM.");

System.out.println("Choose options \n1.Balance Enquiry.\n2.WithDraw.\n3.Deposite.\n4.Account Details.\n5.Exit.");

i=1;

Bank atm = new Bank();

while(i<=n) {

System.out.println("Enter the Option:");

num = s.nextInt();

switch (num) {

case 1:

atm.Balance();

break;

case 2:

atm.withdraw();

break;

case 3:

atm.deposite();

break;

case 4:

atm.detailes();

break;

case 5:

System.out.println("Exited");

n=0;

break;

default:

System.out.println("Invalid Option");

break;

}

n++;

i++;}

}

}

**Output:**

Welcome to the ATM.

Choose options

1.Balance Enquiry.

2.WithDraw.

3.Deposite.

4.Account Details.

5.Exit.

Enter the Option:

1

Enter the pin:

4590

Account Balance is 500000.0

Enter the Option:

2

Enter the pin:

4590

Enter the Amount:

250000

The Amount is Withdrawed.

The Remaining Balance is 250000.0

Enter the Option:

3

Enter the Account Number:

1234567891

Enter the Pin:

4590

Enter the Amount:

500000

The Total Balance is 750000.0

Enter the Option:

4

Enter the Pin:

4590

Account No:1234567891

Acc Houlder Name:seera kalyan

Branch:Donparty Branch

Bank Name:AXISBalance:750000.0

Enter the Option:

5

Exited

**29-03-2022.**

**Constructor:**

* Constructor is a type of method, whose name is same as a class name and which does not contain any return type (Even void also).
* Constructor invoked and executed automatically at the time of object creation.
* Only one object is created for one constructor, if you want to call the constructor again create new object again.
* If you don’t declare any constructor the JVM will provide default constructor without arguments.

**Basic Constructor:**

**package** constructor;

**class** Add{

Add(){

**int** a=10,b=26,c;

c=a+b;

System.***out***.println("This is Constructor:");

System.***out***.println(c);

}

}

**public** **class** ConstructorEx {

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

System.***out***.println("This is main method:");

**new** Add();//we can call without any reference variable.

}

}

**Output:**

This is main method:

This is Constructor:

36

Constructor can be defined two ways:

**1.Default Constructor:**

Constructor with no arguments is called Default Constructor.

**Syntax:**

class name( ) {

Statement t;

}

**2.Parameterized Constructor:**

Constructor with arguments is called Parameterized Constructor.

**Syntax:**

class name(arguments){

Statements;

}

**Default Constructor:**

**package** constructor;

**class** Basic{

//so,here jvm will create Default Constructor for this Constructor.

//That is why is not throwing any Error.

}

**public** **class** Default\_Constructor {

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

System.***out***.println("This is main method.");

**new** Basic();

}

}

**Output:**

This is main method.

**EX2**

**package** constructor;

**class** Car {

String carname;

**int** price;

String color;

**int** capacity;

**public** Car(){//no arguments

carname="Hyundai i20";

price=700000;

color="Red";

capacity=4;

System.***out***.println("Car Name: "+carname);

System.***out***.println("Colour: "+color);

System.***out***.println("Capacity: "+capacity);

System.***out***.println("Price: "+price);

}

}

**public** **class** ConstructorEx2 {

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

**new** Car();

}

}

**Output:**

Car Name: Hyundai i20

Colour: Red

Capacity: 4

Price: 700000

**Parameterized Constructor:**

**package** constructor;

**class** Student

{

String name;

**int** age;

//parameterized constructor

**public** Student(String sname,**int** sage)

{

name=sname;

age=sage;

}

}

**public** **class** parametrisedConstructor {

**public** **static** **void** main(String args[])

{

Student s1=**new** Student("John",22);

System.***out***.println(s1.name+" "+s1.age);

Student s2=**new** Student("Rohan",24);

System.***out***.println(s2.name+" "+s2.age);

Student s3=**new** Student("Rahul",24);

System.***out***.println(s3.name+" "+s3.age);

}

}

**Output:**

John 22

Rohan 24

Rahul 24

**EX2:**

**Constructor Overloading:**

**package** constructor;

**class** Calculation{

Calculation(**int** x,**int** y){

**int** z;

z=x+y;

System.***out***.println("Additions is: "+z);

}

Calculation(**double** d,**float** f) {

**double** m;

m=d\*f;

System.***out***.println("Multiplication is: "+m);

}

Calculation(**double** d,**int** i) {

**double** di;

di =d-i;

System.***out***.println("Substraction is: "+di);

}

Calculation(String name,**int** id) {

System.***out***.println("Name: "+name+"\nId: "+id);

}

}

**public** **class** ConstructorOverloading {

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

System.***out***.println("This is main method:");

**new** Calculation(77, 11);

**new** Calculation(86.56, 41.16f);

**new** Calculation(55.0, 7);

**new** Calculation("kalyan",459);

}

}

**Output:**

This is main method:

Additions is: 88

Multiplication is: 3562.8095867919924

Substraction is: 48.0

Name: kalyan

Id: 459

**30-03-2022.**

**Inheritance:**

* Inheritance means acquiring the properties from parent class to child class is called inheritance.
* By using “extends” keyword we can inherit the classes.

**Advantages of Inheritance:**

**Code Reusability:** We can use variables and methods from one class to another class.

**Extensibility:** We can add or extend new features to the existing class with another class.

**Obtain Functionalities:** We can get functionalities of a class into another class.

**Types of Inheritance:**

**1) Single Inheritance.**

**2) Multi level Inheritance.**

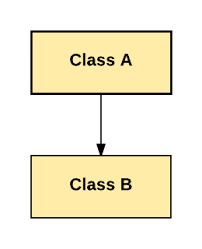
**3) Hierarchical Inheritance.**

**4) Multiple Inheritances.**

**5) Hybrid Inheritance.**

**Single Inheritance:**

Acquiring the properties from one class (parent) to only one class (child) is known as Single Inheritance.



**Syntax:**

class Parent{

statement1;

}

class Child extends Parent{

statement2;

}

**Note:**

**In inheritance we should create an object for only sub class .therefore super or parent class will get memory as a part of sub or child class object.**

**Examples: Single Inheritance**

**Ex1.**

**package** inheritance;

**class** Parent{

**public** **void** land() {

System.***out***.println("Land was Brought by Parent.");

}

}

**class** Child **extends** Parent{

**public** **void** house() {

System.***out***.println("Build a House in the Land by child.");

}

}

**public** **class** SingleInheritance {

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

System.***out***.println("Single Inheritance:");

Child ch = **new** Child();

ch.land();

ch.house();

}

}

**Output:**

Single Inheritance:

Land was Brought by Parent.

Build a House in the Land by child.

**Ex2.**

**package** inheritance;

**import** java.util.Scanner;

**class** Number{

**int** x,y;

**public** **void** add(**int** x, **int** y) {

**this**.x=x;

**this**.y=y;

}

}

**class** Add **extends** Number{

**int** total;

**public** **void** sum() {

total = x+y;

System.out.println("Addiction of Two is: "+total);

}

}

**public** **class** SingleInheritance2 {

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

**int** a,b;

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

System.out.println("Single Inheritance:");

System.out.println("Enter the first number:");

a=sc.nextInt();

System.out.println("Enter the Second number:");

b=sc.nextInt();

Add ai = **new** Add();

ai.add(a, b);

ai.sum();

}

}

**Output:**

Single Inheritance:

Enter the first number:

23

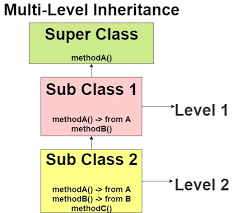
Enter the Second number:

56

Addiction of Two is: 79

**Multi-Level Inheritance:**

Acquiring properties from grandparent class to parent class and parent to child class and child to another child class and so on is known as Multilevel Inheritance.



**Syntax:**

class A{  
//class A is parent of class B  
//class A is grand parent of class C  
public A(){  
//A constructor  
}  
public void fun1(){  
//function in Parent Class  
}  
}  
class B extends A{  
//class B is a child class of class A  
//class B is a parent class of class C  
public B(){  
//class B constructor  
}  
}  
class C extends B{  
//class C is a child class of class B  
//class C is grandchild class of class A  
public C(){  
//Class C constructor  
}  
}  
public class Test{  
public static void main(String[] args){  
C obj = new C();  
}  
}

**Examples: Multi level inheritance.**

**Ex1.**

**package** inheritance;

**class** Company{

**static** String *cname*="Marolix";

**static** String *loc* = "Hyderabad";

**public** **void** getdata() {

System.***out***.println("Company Name: "+*cname*);

System.***out***.println("Location: "+*loc*);

}

}

**class** Emp\_Details **extends** Company{

**int** id;

String ename;

String dob;

String roll;

**public** **void** detail(**int** id,String ename,String dob,String roll) {

**this**.id=id;

**this**.ename=ename;

**this**.dob=dob;

**this**.roll=roll;

System.***out***.println("Employee ID: "+id);

System.***out***.println("Employee Name: "+ename);

System.***out***.println("Employee DOB: "+dob);

System.***out***.println("Employee Designation: "+roll);

}

}

**public** **class** MultiLevelInheritance {

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

System.***out***.println("Multi Level Inheritance:");

Emp\_Details ed = **new** Emp\_Details();

ed.getdata();

ed.detail(459, "kalyan", "19-02-1997", "Developer");

}

}

**Output:**

Multi Level Inheritance:

Company Name: Marolix

Location: Hyderabad

Employee ID: 459

Employee Name: kalyan

Employee DOB: 19-02-1997

Employee Designation: Developer

**Ex2.**

**package** inheritance;

**import** java.util.Scanner;

**class** CompanyDetails{

**static** String *Compname* = "Marolix";

**static** **long** *phoneno*= 1234567890;

**static** String *loc* = "Hyderabad";

**public** **void** print() {

System.***out***.println("Company Name: "+*Compname*);

System.***out***.println("Phone No: "+*phoneno*);

System.***out***.println("Location: "+*loc*);

}

}

**class** EmpDetails **extends** CompanyDetails {

**int** id;

String name;

**long** empphoneno;

String empmail;

String dob;

String emproll;

String empexp;

**public** **void** details(**int** id,String name,**long** empphoneno,String empmail,String d0b,String emproll,String empexp ) {

**this**.id=id;

**this**.name=name;

**this**.empphoneno=empphoneno;

**this**.empmail=empmail;

**this**.dob=dob;

**this**.emproll=emproll;

**this**.empexp=empexp;

System.***out***.println("Employe ID: "+id);

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

System.***out***.println("Employee PhoneNo: "+empphoneno);

System.***out***.println("Employee Mail: "+empmail);

System.***out***.println("Employee DOB: "+dob);

System.***out***.println("Employee Designation: "+emproll);

System.***out***.println("Employee Experience: "+empexp);

}

}

**public** **class** MultiLevelInheritance2 {

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

**int** id;

String name;

**long** empphoneno;

String empmail;

String dob;

String emproll;

String empexp;

System.***out***.println("Multi Level Inheritence:");

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

System.***out***.println("Employee ID");

id=sc.nextInt();

System.***out***.println("Employee Name");

name = sc.next();

System.***out***.println("Employee Phoneno");

empphoneno=sc.nextLong();

System.***out***.println("Employee Mail");

empmail= sc.next();

System.***out***.println("Employee DOB");

dob=sc.next();

System.***out***.println("Employee Designation");

emproll=sc.next();

System.***out***.println("Employee Experience");

empexp=sc.next();

EmpDetails empd = **new** EmpDetails();

empd.print();

empd.details(id, name, empphoneno, empmail, dob, emproll, empexp);

}

}

**Output:**

Multi Level Inheritence:

Employee ID

457

Employee Name

venu

Employee Phoneno

9876543212

Employee Mail

venu1997@gmail.com

Employee DOB

23-02-1997

Employee Designation

Developer

Employee Experience

2years

Company Name: Marolix

Phone No: 1234567890

Location: Hyderabad

Employe ID: 457

Employe Name: venu

Employee PhoneNo: 9876543212

Employee Mail: venu1997@gmail.com

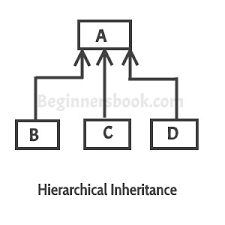
Employee DOB: null

Employee Designation: Developer

Employee Experience: 2years

**Hierarchical Inheritance:**

Acquiring properties from a super class into multiple sub classes is known as Hierarchical Inheritance.



**Syntax:**

class Superclass{

// variables and methods

}

class Subclass1 extends Superclass  
{  
// variables and methods  
}  
class Subclass2 extends Superclass  
{  
// variables and methods  
}

**Example: Hierarchical Inheritance.**

**Ex1.**

**package** inheritance;

**class** SuperClass{

**int** a=10;

**int** b=34;

**public** **void** sclass() {

System.***out***.println("a and b values in the Superclass are "+a+" & "+b);

}

}

**class** Child1 **extends** SuperClass{

**int** c;

**public** **void** adds() {

c=a+b;

System.***out***.println("Addiction of Two numbers is: "+c);

}

}

**class** Child2 **extends** SuperClass{

**int** c;

**public** **void** muls() {

c=a\*b;

System.***out***.println("Multiplication of two numbers is: "+c);

}

}

**public** **class** HierarchicalInheritance {

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

System.***out***.println("Hierarchical Inheritance:");

Child1 ch1 = **new** Child1();

ch1.adds();

Child2 ch2 = **new** Child2();

ch2.muls();

}

}

**Output:**

Hierarchical Inheritance:

Addiction of Two numbers is: 44

Multiplication of two numbers is: 340

**06-04-2022.**

**Single Inheritance:**

Ex1:

**package** inheritance;

**import** java.util.Scanner;

**class** Student {

**int** sid;

String sname;

**public** **void** sread() {

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("Enter the Student ID and Student Name:");

sid = scan.nextInt();

sname = scan.next();

}

**public** **void** sshow() {

System.***out***.println("Student ID: "+sid);

System.***out***.println("Student Name: "+sname);

}

}

**class** Result **extends** Student{

**int** sub1,sub2,sub3,total,avg;

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

**public** **void** sread1() {

**super**.sread();

System.***out***.println("Enter the Sub1 sub2 and sub3 marks:");

sub1=sc.nextInt();

sub2=sc.nextInt();

sub3=sc.nextInt();

}

**public** **void** sshow1() {

**super**.sshow();

total=sub1+sub2+sub3;

avg = total/3;

System.***out***.println("Total marks are "+total);

System.***out***.println("Average is "+avg);

}

}

**public** **class** SingleInheritanceEx3 {

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

System.***out***.println("Single Inheritance:");

Result rrr = **new** Result();

rrr.sread1();

rrr.sshow1();

}

}

**Output:**

Single Inheritance:

Enter the Student ID and Student Name:

459

kalyan

Enter the Sub1 sub2 and sub3 marks:

78

89

92

Student ID: 459

Student Name: kalyan

Total marks are 259

Average is 86

**Multi-Level Inheritance:**

**Ex1:**

**package** inheritance;

**import** java.util.Scanner;

**class** StudentM{

**int** sid;

String sname;

**public** **void** readM() {

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("Enter the Student id and Student Name:");

sid = scan.nextInt();

sname = scan.next();

}

**public** **void** showM() {

System.***out***.println("Student ID no: "+sid);

System.***out***.println("Student Name: "+sname);

}

}

**class** ResultM **extends** StudentM{

**int** sub1,sub2,sub3,total,avg;

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

**public** **void** readM1() {

System.***out***.println("Enter the sub1,sub2 and sub3 Marks:");

sub1=sc.nextInt();

sub2=sc.nextInt();

sub3 =sc.nextInt();

}

**public** **void** showM1() {

total= sub1+sub2+sub3;

avg=total/3;

System.***out***.println("Total Marks are "+total);

}

}

**class** RankM **extends** ResultM{

**public** **void** readM2() {

**super**.readM();

**super**.readM1();

}

**public** **void** showM2() {

**super**.showM();

**super**.showM1();

**if** (avg<=50) {

System.***out***.println("Average of marks is "+avg+" and the Rank is FAIL.");

}

**else** **if** (avg>=50 && avg<=60) {

System.***out***.println("Average of marks is "+avg+" and the Rank is PASS.");

}

**else** **if**(avg>=60 && avg<=70) {

System.***out***.println("Average of marks is "+avg+" and the Rank is FIRST CLASS.");

}

**else** **if**(avg>70) {

System.***out***.println("Average of marks is "+avg+" and the Rank is UNIVERSITY TOPPER.");

}

}

}

**public** **class** MultiLevelInheritanceEx3 {

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

System.***out***.println("Multi-Level Inheritance:");

RankM r = **new** RankM();

r.readM2();

r.showM2();

}

}

**Output:**

Multi-Level Inheritance:

Enter the Student id and Student Name:

459

Kalyan

Enter the sub1,sub2 and sub3 Marks:

78

89

96

Student ID no: 459

Student Name: Kalyan

Total Marks are 263

Average of marks is 87 and the Rank is UNIVERSITY TOPPER.

**07-04-2022.**

**Interface:**

* Interface is declared by using interface keyword.
* The class that implements an interface must implements all the methods in the interface.
* It is used for achieving abstraction by using interface we can support the functionalities of multiple inheritance.
* It provides protection abstraction which means all the methods in interface are declared with empty body.

**Ex1:**

**package** interfaceEx;

**interface** inter{

**int** ***x***=10;

**public** **void** show();

}

**class** Display **implements** inter{

**public** **void** show(){

System.***out***.println("Interface.");

}

}

**public** **class** InterfaceEx1 {

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

System.***out***.println("Main Method:");

Display d= **new** Display();

d.show();

}

}

**Output:**

Main Method:

Interface.

**Ex2:**

**package** interfaceEx;

**import** java.util.Scanner;

**interface** Inter1{

**int** ***x***=10;

**public** **void** message();

**public** **void** calculate();

}

**class** Actual **implements** Inter1{

**int** a,b;

Actual(**int** a,**int** b){

**this**.a=a;

**this**.b=b;

}

**public** **void** message() {

System.***out***.println("Good Morning.....");

}

**public** **void** calculate() {

System.***out***.println("Addiction is "+(a+b));

}

}

**public** **class** InterfaceEx2 {

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

**int** x,y;

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

System.***out***.println("Enter the Values for x and y:");

x=sc.nextInt();

y=sc.nextInt();

System.***out***.println("Interface:");

Actual ac = **new** Actual(x,y);

ac.message();

ac.calculate();

}

}

**Output:**

Enter the Values for x and y:

3

9

Interface:

Good Morning.....

Addiction is 12

**Ex3:**

**package** interfaceEx;

**interface** Interface1{

**public** **void** showI1();

}

**interface** Interface2{

**public** **void** showI2();

}

**class** ActualInterface **implements** Interface1,Interface2{

**public** **void** showI1() {

System.***out***.println("From First Interface.");

}

**public** **void** showI2() {

System.***out***.println("From Second Interface.");

}

}

**public** **class** InterfaceEx3 {

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

System.***out***.println("Multiple Inharitance by using interface:");

ActualInterface ai = **new** ActualInterface();

ai.showI1();

ai.showI2();

}

}

**Output:**

Multiple Inharitance by using interface:

From First Interface.

From Second Interface.

**Ex4:**

**package** interfaceEx;

**import** java.util.Scanner;

**interface** ADD{

**public** **void** add();

}

**interface** MULT{

**public** **void** mult();

}

**interface** SUB{

**public** **void** sub();

}

**class** Calculator **implements** ADD,MULT,SUB{

**int** a,b;

Calculator(**int** a,**int** b){

**this**.a=a;

**this**.b=b;

System.***out***.println("Calculator:");

}

**public** **void** add() {

**int** c=a+b;

System.***out***.println("Addiction of "+a+" and "+b+" is "+c+".");

}

**public** **void** mult() {

**int** c = a\*b;

System.***out***.println("Multiplication of "+a+" and "+b+" is "+c+".");

}

**public** **void** sub() {

**int** c = a-b;

System.***out***.println("Substraction of "+a+" and "+b+" is "+c+".");

}

}

**public** **class** InterfaceEx4 {

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

**int** x,y;

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println("Enter the value's of x and y:");

x=scan.nextInt();

y=scan.nextInt();

System.***out***.println("Multiple Inheritance by using Interface:");

Calculator cal = **new** Calculator(x,y);

cal.add();

cal.mult();

cal.sub();

}

}

**Output:**

Enter the value's of x and y:

45

89

Multiple Inheritance by using Interface:

Calculator:

Addiction of 45 and 89 is 134.

Multiplication of 45 and 89 is 4005.

Substraction of 45 and 89 is -44.

**Ex5:**

**package** interfaceEx;

**interface** Uses{

**public** **void** Show();

}

**class** Employee1 **implements** Uses{

**public** **void** Show() {

System.***out***.println("Uses is implemented in the Employee1 Class:");

System.***out***.println("Employee1 is working in the Marolix Company.");

}

}

**class** Employee2 **implements** Uses{

**public** **void** Show() {

System.***out***.println("Uses is implemented in the Employee2 Class:");

System.***out***.println("Employee2 is working in the Marolix Company.");

}

}

**public** **class** InterfaceEx5 {

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

System.***out***.println("Multiple Inheritance by using interface:");

Employee1 emp1 = **new** Employee1();

emp1.Show();

Employee2 emp2 = **new** Employee2();

emp2.Show();

}

}

**Output:**

Multiple Inheritance by using interface:

Uses is implemented in the Employee1 Class:

Employee1 is working in the Marolix Company.

Uses is implemented in the Employee2 Class:

Employee2 is working in the Marolix Company.

**Ex6:**

**package** interfaceEx;

**import** java.util.Scanner;

**interface** Variables{

**public** **void** var();

}

**class** ADDICTION **implements** Variables{

**int** a,b,c;

ADDICTION(**int** a,**int** b){

**this**.a=a;

**this**.b=b;

}

**public** **void** var() {

c=a+b;

System.***out***.println("Addiction of "+a+" and "+b+" is "+c+".");

}

}

**class** SUBSTRACTION **extends** ADDICTION **implements** Variables{

**int** a,b,c;

SUBSTRACTION(**int** a ,**int** b){

**super**(a,b);

**super**.var();

**this**.a=a;

**this**.b=b;

}

**public** **void** var() {

c=a-b;

System.***out***.println("Substraction of "+a+" and "+b+" is "+c+".");

}

}

**public** **class** InterfaceEx6 {

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

**int** x,y;

Scanner scan = **new** Scanner(System.***in***);

System.***out***.println(" Multiple Inheritance by using Interface:");

System.***out***.println("Enter the Value's of x and y:");

x=scan.nextInt();

y=scan.nextInt();

// ADDICTION aa = new ADDICTION(x, y);

// aa.var();

SUBSTRACTION ss = **new** SUBSTRACTION(x, y);

ss.var();

}

}

**Output:**

Multiple Inheritance by using Interface:

Enter the Value's of x and y:

45

76

Addiction of 45 and 76 is 121.

Subtraction of 45 and 76 is -31.

**Ex7:**

**package** interfaceEx;

**import** java.util.Scanner;

**interface** Company{

**public** **void** read();

**public** **void** show();

}

**class** Employee{

**int** eid;

String ename;

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

**public** **void** read1() {

System.***out***.println("Enter the Employee ID and Name:");

eid=sc.nextInt();

ename=sc.next();

}

**public** **void** show1() {

System.***out***.println("Employee ID: "+eid);

System.***out***.println("Employee Name: "+ename);

}

}

**class** Call **extends** Employee **implements** Company{

String cname;

String clocation;

**public** **void** read() {

cname="Marolix";

clocation="Hyderabad";

**super**.read1();

}

**public** **void** show() {

System.***out***.println("Company Name: "+cname);

System.***out***.println("Company Location: "+clocation);

**super**.show1();

}

}

**public** **class** InterfaceEx7 {

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

System.***out***.println("Multiple Inheritance by using interface:");

Call cl = **new** Call();

cl.read();

cl.show();

}

}

**Output:**

Multiple Inheritance by using interface:

Enter the Employee ID an Name:

459

kalyan

Company Name: Marolix

Company Location: Hyderabad

Employee ID: 459

Employee Name: kalyan

**Ex8:**

**package** interfaceEx;

**interface** Dinter{

**public** **void** Dshow1();

**default** **void** dshow() {

System.***out***.println("This is Default method.");

}

}

**class** Test **implements** Dinter{

**public** **void** Dshow1() {

System.***out***.println("This is Print Method in the Test class");

}

}

**public** **class** InterfaceEx8 {

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

System.***out***.println("Multiple Inheritance by using Interface:");

Test t = **new** Test();

t.Dshow1();

t.dshow();

}

}

**Output:**

Multiple Inheritance by using Interface:

This is Print Method in the Test class

This is Default method.

**Ex9:**

**package** interfaceEx;

**interface** Drawble{

**public** **void** draw();

**static** **int** cube(**int** i ) {

**return** i\*i\*i;

}

}

**class** Rectangle **implements** Drawble{

**public** **void** draw() {

System.***out***.println("Reactangle Triangle.");

}

}

**public** **class** InterfaceEx9 {

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

System.***out***.println("Multiple Interitance by using Interface:");

Rectangle rec = **new** Rectangle();

rec.draw();

System.***out***.println(Drawble.*cube*(4));

}

}

**Output:**

Multiple Interitance by using Interface:

Reactangle Triangle.

64

**13-04-2022.**

**Abstraction:**

Abstraction is the process of hiding the implementations details and showing only functionality to the user.

They are two ways to achieve abstraction.

* Abstract class.
* Interface.

**Abstract class:**

* A class which is declared as an abstract which is known as abstract.
* It can have both abstract and non-abstract methods .
* Its need to be extended and it abstract method need to be implemented.
* Abstract class cannot be instantiated.
* Abstract method can only be used in abstract class only.
* An abstract method doesn’t have body.

**Ex1:**

**package** abstractEx;

**abstract** **class** A{

**int** a;

// abstract class

**abstract** **void** a();

**int** b(**int** a) {

**this**.a =a;

System.***out***.println(a);

**return** a;

}

**abstract** **int** c(**int** ca);

}

**abstract** **class** B **extends** A{

@Override

**void** a() {

System.***out***.println("a inside B");

}

}

**class** C **extends** B{ //concrete class

@Override

**public** **int** c(**int** ca) {

System.***out***.println(ca);

**return** ca;

}

}

**public** **class** AbstracteEx2 {

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

C cc = **new** C();

cc.a();

cc.b(10);

cc.c(22);

}

}

**Output:**

a inside B

10

22

**14-04-2022.**

**Exception:**

* Exception is an event which doesn’t allow normal flow of program.
* This in an abject which thrown at run time.

**Exception Handling:**

Exception handling is used to handle runtime error’s or exceptions it maintain normal flow of application.

**Types of Exceptions:-**

* **Checked Exception.**
* **Un-checked Exception.**
* **Error.**

**Checked Exception:-**

* Checked exception are checked at compile time.

**EX:-** sql Exception, io Exception ,Class Not Found Exception.

**Un-checked Exception:-**

* Unchecked exception are not checked at compile time but they checked at run time.

**EX:-**Null Pointer Exception, Athematic Exception, Number Format Exception, Index Out Of Bounds Exception.

**Error:-**

Error is irrecoverable.

**EX:-**Out of Memory, Stack Overflow Error.

**Keywords In the Exception Handling:**

* **try**
* **catch**
* **finally**
* **throw**
* **throws**

**try:-**

* Try block is were we write the code.
* It must be followed by either catch **or** finally.

**catch:-**

* The catch block is used to handle the exception.
* It must be return after try block.
* It can be followed by finally block if it necessary.

**Finally:-**

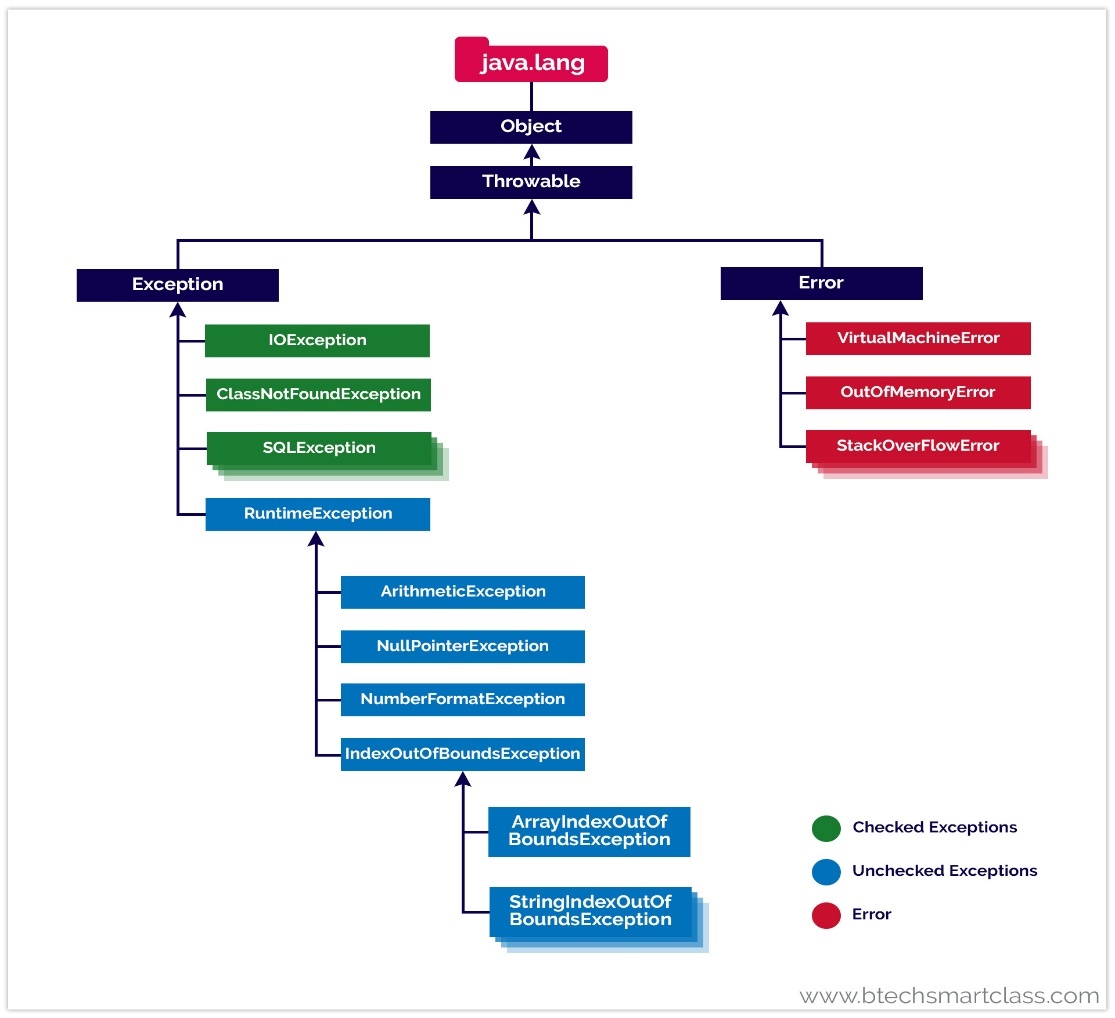
* Finally block is used to execute the necessary code of the program.
* It is executed whether an exception is handled or not.

**Throw:-**

* The **“throw “** keyword is used to throw an exception.

**Throws:-**

* It is used to declare exceptions.
* It always used with method signature.

****

**Try Catch Block:**

**Ex1:**

**package** exceptionHandling;

**import** java.io.\*;

**class** Excep{

**int** a=100,b=0,c;

**public** **void** read() {

**try** {

c=a/b;

System.***out***.println(c);

} **catch** (Exception e) {

System.***out***.println(e);

}

}

}

**public** **class** TryCatchBlocks {

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

System.***out***.println("Main Method:");

Excep ex = **new** Excep();

ex.read();

}

}

**Output:**

Main Method:

java.lang.ArithmeticException: / by zero

**Ex2:**

package exceptionHandling;

import java.util.Scanner;

import java.io.\*;

class Excep1{

int a,b,c;

Scanner scan = new Scanner(System.in);

public void readE() {

try {

System.out.println("Enter the Value's of a and b:");

a=scan.nextInt();

b=scan.nextInt();

c=a/b;

System.out.println("Output: "+c);

} catch (Exception e1) {

System.out.println(e1);

}

}

}

public class TryCatchBlockEx1 {

public static void main(String[] args) {

System.out.println("Main Method:");

Excep1 ex1 = new Excep1();

ex1.readE();

}

}

**Output:**

Main Method:

Enter the Value's of a and b:

23

0

java.lang.ArithmeticException: / by zero

**Try Multi-Catch Block:**

**Ex1:**

**package** exceptionHandling;

**import** java.util.Scanner;

**import** java.lang.\*;

**class** Excep2{

**int** a,b,c;

String name ="kalyan";

Scanner scan = **new** Scanner(System.***in***);

**public** **void** readE2(){

**try** {

System.***out***.println("Enter the values of a and b");

a=scan.nextInt();

b=scan.nextInt();

c=a/b;

System.***out***.println(c);

System.***out***.println(name.charAt(6));

} **catch** (ArithmeticException a) {

System.***out***.println(a);

}

**catch** (StringIndexOutOfBoundsException si) {

System.***out***.println(si);

}

**finally** {

System.***out***.println("This finally Block.");

}

}

}

**public** **class** TryMultiCatchEx1 {

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

System.***out***.println("Main Method:");

Excep2 ex2 = **new** Excep2();

ex2.readE2();

}

}

**Output:**

Main Method:

Enter the values of a and b

100

0

java.lang.ArithmeticException: / by zero

This finally Block.

**20-04-2022:**

**Strings:**

* String is the sequence of characters.
* In java string can be declared using "String" predefined class. It is available in "java.lang" package. The "java.lang" is the default package in java.
* String will maintain an index in the memory to store the characters.
* String index starts with zero.
* In java String is a single dimensional character array.

**Ex1:-**

**package** strings;

**public** **class** StringEx1 {

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

System.***out***.println("Main Method:");

//it is == operator String comparing.

String s1 ="java";

String s2 ="jvaa";

String s3 = **new** String("this is java class");

String s4 = **new** String("JAVA");

System.***out***.println(s1.compareTo(s2));//Here it will compare by the hashcode of s1 and s2.

System.***out***.println(s1==s3);//Here s3 will be created in the heap memory then point to the "java" in the String Constant Pool.

System.***out***.println(s3.compareTo(s3));

System.***out***.println(s3.compareToIgnoreCase(s4));

System.***out***.println(s1.toUpperCase());

System.***out***.println(s4.toLowerCase());

System.***out***.println(s1.startsWith(s2));

System.***out***.println(s3.startsWith(s1,8));

System.***out***.println(s3.replace('t', 'T'));

System.***out***.println(s1.indexOf('k'));

}

}

**Output:**

Main Method:

-21

false

0

10

JAVA

java

false

true

This is java class

-1

**Ex2-**

**package** strings;

**public** **class** StringMutability {

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

String name="Seera Kalyan";

System.***out***.println(name);

System.***out***.println(name.hashCode());

String full\_name =name.concat(" Kumar");

System.***out***.println(full\_name);

System.***out***.println(full\_name.hashCode());

}

}

**Output:**

Seera Kalyan

585426222

Seera Kalyan Kumar

389453346

**Ex3:-**

**package** strings;

**public** **class** StringSplit {

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

String s1 = "java by kalyan.";

String x[]=s1.split(" ");

**for**(String s:x)

System.***out***.println(s);

String dob ="03-02-1997";

String y[] =dob.split("-");

**for**(String s2:y)

System.***out***.println(s2);

}

}

**Output:**

java

by

kalyan.

03

02

1997

**Ex4:-**

**package** strings;

**public** **class** SubString {

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

System.***out***.println("Main Method:");

String str ="Getting a SubString";

System.***out***.println(str);

String sub = str.substring(0, 9);

System.***out***.println(sub);

System.***out***.println(str.substring(10));

}

}

**Output:**

Main Method:

Getting a SubString

Getting a

SubString

**25-04-2022.**

**Thread:**

* A thread is a part of a process. i.e. Thread is a small process or a sub-process under a large process.
* Thread is a separate (or) independent path of execution of a program because each thread runs in a different stack frame. It means for every Thread a new stack frame will be created in Stack area of JVM.
* All the threads of a process share the same address space.
* Each thread can perform a particular task.
* Thread is a light weight task because it will take less memory and less execution time. i.e. Cost (Time, memory)Communication between the thread is low.
* One thread can read, write or change another threads data.

**Ex1:**

**package** threadingAndMultiThreading;

**public** **class** Threading {

**public** **static** **void** main(String[] args) **throws** InterruptedException {

Thread t = Thread.*currentThread*();

System.***out***.println(t);

t.setName("Kalyan");

System.***out***.println(t);

System.***out***.println(t.getId());

System.***out***.println(t.getName());

t.setPriority(9);

System.***out***.println(t.getPriority());

System.***out***.println(t.getState());

System.***out***.println(t.getClass());

System.***out***.println(t.getContextClassLoader());

System.***out***.println(t.getStackTrace());

System.***out***.println(t.getThreadGroup());

System.***out***.println(t.getUncaughtExceptionHandler());

System.***out***.println(Thread.*getAllStackTraces*());

System.***out***.println(t.isAlive());

System.***out***.println(t.isDaemon());

System.***out***.println(Thread.*activeCount*());

// t.wait();

System.***out***.println(t);

}

}

**Output:**

Thread[main,5,main]

Thread[Kalyan,5,main]

1

Kalyan

9

RUNNABLE

class java.lang.Thread

jdk.internal.loader.ClassLoaders$AppClassLoader@c387f44

[Ljava.lang.StackTraceElement;@7a81197d

java.lang.ThreadGroup[name=main,maxpri=10]

java.lang.ThreadGroup[name=main,maxpri=10]

{Thread[Common-Cleaner,8,InnocuousThreadGroup]=[Ljava.lang.StackTraceElement;@5305068a, Thread[Reference Handler,10,system]=[Ljava.lang.StackTraceElement;@1f32e575, Thread[Signal Dispatcher,9,system]=[Ljava.lang.StackTraceElement;@279f2327, Thread[Finalizer,8,system]=[Ljava.lang.StackTraceElement;@2ff4acd0, Thread[Kalyan,9,main]=[Ljava.lang.StackTraceElement;@54bedef2, Thread[Attach Listener,5,system]=[Ljava.lang.StackTraceElement;@5caf905d}

true

false

1

Thread[Kalyan,9,main]

**Ex2:**

**package** threadingAndMultiThreading;

**import** java.util.Scanner;

**public** **class** ThreadingEx1 {

**public** **static** **void** main(String[] args) **throws** Exception {

System.***out***.println("Main Method:");

**int** accno;

**int** password;

**int** balance = 50000;

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

System.***out***.println("Enter the Account Number:");

accno=sc.nextInt();

System.***out***.println("Enter the Password:");

password=sc.nextInt();

Thread.*sleep*(3000);

System.***out***.println("Your in the online Portal");

**if** (balance>=0) {

Thread.*sleep*(5000);

System.***out***.println(balance);

}

System.***out***.println("Thank you.");

}

}

**Output:**

Main Method:

Enter the Account Number:

12345678

Enter the Password:

234567

Your in the online Portal

50000

Thank you.

**Multithreading:**

* It is the process of executing multiple threads simultaneously to perform multiple tasks at the same time.
* Using Multithreading concept we can develop applications to perform multiple tasks at the same time.
* Each thread starts its own stack based on the flow (or) priority of the threads.
* The word "Multithreading" is derived from multitasking.

**Ex1:**

**package** threadingAndMultiThreading;

**import** java.util.Scanner;

**class** MulThread1 **extends** Thread{

**int** accno;

**int** password;

**public** **void** run() {

**try** {

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

System.***out***.println("Enter the Account Number:");

accno=sc.nextInt();

System.***out***.println("Enter the password:");

password=sc.nextInt();

Thread.*sleep*(5000);

System.***out***.println("Your is the online portal.");

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

**class** Multhread2 **extends** Thread{

**public** **void** run() {

**int** balance=50000;

**try** {

**if** (balance>0) {

Thread.*sleep*(3000);

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

}

}**catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

**class** MulThread3 **extends** Thread{

**public** **void** run() {

**try** {

Thread.*sleep*(2000);

System.***out***.println("This is Axis Bank.");

}

**catch** (Exception e) {

// **TODO**: handle exception

}

}

}

**public** **class** MultiThreadingEx1 {

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

System.***out***.println("Main Method:");

MulThread1 md1= **new** MulThread1();

Multhread2 md2 = **new** Multhread2();

MulThread3 md3 =**new** MulThread3();

md1.start();

md2.start();

md3.start();

}

}

**Output:**

Main Method:

Enter the Account Number:

12345

Enter the password:

355

This is Axis Bank.

Balance is :50000

Your is the online portal.

**26-04-2022.**

**Join( ) Method:-**

* Main thread is a thread who will start the execution almost in all the scenario’s.
* It will be completing the execution first .
* Many times in the program other threads have to use some resources. Which available in main thread.
* Hence, at the time main thread should wait inside for other threads to complete the execution.
* It can be achieved by making use of join( ) method.
* To check whether the thread is Alive or not we use isAlive( ) method.
* It will return Boolean type of return.

1. multithreading can be achieved by creating abject for a class.
2. If we create an object foe a class two activities will take place.
3. Extra stack will be created.
4. Extra thread will be assigned for extra stack.

**Note:-**

**It will always provide body for run method with paralysis { } , but one should never call run() method. we can call start( ) method**

**Thread Scheduler:-**

Thread scheduler is a software which manages and schedules the extra thread.

We can achieve Multithreading by two ways:-

1. By extends Thread class.
2. By Implements Runnable Interface.

**Note:-**

**If we implement the Runnable Interface to the class then we must implement at least one method from the Runnable interface.**

**27-04-2022.**

**StringBuffer:-**

String Buffer and String Builder are mutable class where as String class immutable.it is synchronized.

**Mutable String:-**

A string that can be modified (or) can be changed is called Mutable

**Ex1:-**

**package** strings;

**public** **class** StringBuilderEx1 {

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

// StringBuilder sb = new StringBuilder("java");

// Any thing is same but StringBuilder is faster then StringBuffer

StringBuffer sb = **new** StringBuffer("java");

sb.append("code");

System.***out***.println(sb);

sb.insert(4,'@');

System.***out***.println(sb);

sb.replace(0, 4,"Python");

System.***out***.println(sb);

sb.delete(6,11);

System.***out***.println(sb);

sb.reverse();

System.***out***.println(sb);

}

}

**Output:**

javacode

java@code

Python@code

Python

nohtyP

**Ex2:-**

**package** strings;

**public** **class** StringBuilderEx2 {

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

**int** count=0;

String s="String Buffer and String Builder are mutable class whereas String class immutable.it is synchronized.";

StringBuilder stb = **new** StringBuilder(s);

stb.chars();

System.***out***.println(stb);

**for** (**int** i = 0; i <stb.length(); i++) {

**if** (stb.charAt(i)==' ' || stb.charAt(i)=='.') {

count++;

}

}

System.***out***.println("In the above Sentence there are "+count+" words.");

}

}

**Output:-**

String Buffer and String Builder are mutable class whereas String class immutable.it is synchronized.

In the above Sentence there are 16 words.

**29-04-2022.**

**Collections:**

In java to store the data arrays are use. but arrays have some disadvantages, which are.

1. Arrays are fix in size.
2. Arrays can store only homogenies data.
3. Arrays required to contagious memory location to store data.

To remove this disadvantages we use collection hierarchy containing 7 inbuild classes.Collections means a single unit of objects collections was introduce in java from jdk 1.2.

Collection contains following 7 classes:- **1.Array List( i )2.Linked List( i )3.Array Deque(c)4.Priorty Queue(c)5.Tree Set(c)6.Hash Set(i)7.Linked Hash Set(c)**

**Array List:**

* Array List is an in build class of collection hierarchy, it is present in java.util package.
* This class implements List interface.
* Array list make use of dynamic data structure to store the elements of different data types.
* The element store Array list class randomly access.

**Ex1:-**

**package** arrayList;

**import** java.util.ArrayList;

**public** **class** ArrayListEx1 {

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

ArrayList al = **new** ArrayList();

al.add(45);

al.add("Java");

al.add(59.23f);

al.add(**true**);

System.***out***.println(al);

al.add(2, 89);

System.***out***.println(al);

ArrayList al1 = **new** ArrayList();

al1.add(12);

al1.add(66);

al1.add(88);

al1.add(46);

al1.add(100);

System.***out***.println(al1);

al.addAll(al1);

System.***out***.println(al);

al.add(5, "Program");

System.***out***.println(al);

System.***out***.println(al.contains(220));

System.***out***.println(al.get(3));

System.***out***.println(al.indexOf(66));

al.addAll(3, al1);

System.***out***.println(al);

System.***out***.println(al.indexOf(100));

System.***out***.println(al.lastIndexOf(100));

al.remove(2);

System.***out***.println(al);

al.removeAll(al1);

System.***out***.println(al);

// al.clear();

System.***out***.println(al);

al.set(2, 20);

System.***out***.println(al);

}

}

**Output:-**

[45, Java, 59.23, true]

[45, Java, 89, 59.23, true]

[12, 66, 88, 46, 100]

[45, Java, 89, 59.23, true, 12, 66, 88, 46, 100]

[45, Java, 89, 59.23, true, Program, 12, 66, 88, 46, 100]

false

59.23

7

[45, Java, 89, 12, 66, 88, 46, 100, 59.23, true, Program, 12, 66, 88, 46, 100]

7

15

[45, Java, 12, 66, 88, 46, 100, 59.23, true, Program, 12, 66, 88, 46, 100]

[45, Java, 59.23, true, Program]

[45, Java, 59.23, true, Program]

[45, Java, 20, true, Program]

**Ex2:-**

**package** arrayList;

**import** java.util.Scanner;

**import** java.util.ArrayList;

**public** **class** ArrayListEx2 {

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

**int** n;

String s = "";

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

System.***out***.println("Enter the Array Size:");

n = sc.nextInt();

ArrayList al = **new** ArrayList();

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

System.***out***.println("Enter the Charactor:");

s = sc.next();

System.***out***.println(s);

al.add(s);

}

System.***out***.println("Output:");

System.***out***.println(al);

}

}

**Output:-**

Enter the Array Size:

5

Enter the Charactor:

Audi

Audi

Enter the Charactor:

skoda

skoda

Enter the Charactor:

suziki

suziki

Enter the Charactor:

kia

kia

Enter the Charactor:

Honda

Honda

Output:

[Audi, skoda, suziki, kia, Honda]

**Ex3:-**

**package** arrayList;

**import** java.util.Scanner;

**import** java.util.ArrayList;

**import** java.util.Collections;

**public** **class** ArrayListEx3 {

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

**int** n;

String s = "";

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

System.***out***.println("Enter the array size:");

n = sc.nextInt();

ArrayList al3 = **new** ArrayList();

System.***out***.println("Enter the Elements:");

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

s = sc.next();

al3.add(s);

}

Collections.*sort*(al3);

System.***out***.println("After Sorting the Array List:");

System.***out***.println(al3);

}

}

**Output:-**

Enter the array size:

5

Enter the Elements:

Skoda

kia

Honda

Suziki

Audi

After Sorting the Array List:

[Audi, Honda, Skoda, Suziki, kia]

**04-05-2022.**

**Collection:-**

* It is a framework that provide an architecture to store and manipulate the group of objects.
* Manipulation means delection,sorting,updation,searching etc.
* It is interface which is extend the iterable interface.

collections also as sub interfaces those are

1. **List.**
2. **Queue.**
3. **Set**.

**Hierarchy of Collection Framework**



**List:-**

* List is an Interface which extends **Collection Interface**.
* List allows duplicate values dynamically.
* We can send bulk amount data into list.

List interface is implemented in three class those are

1. **ArrayList(class).**
2. **LinkedList(class).**
3. **Vector(class).**