Introduction 1:-

Institute Name : Ihub Talent Management / Quality Thought

Websites : www.ihubtalent.com / www.qualitythought.in

Location : Ameerpet , Hyderabad.

Course : Full Stack Java Development + AWS

Duration : 4 Months

Mode : Online/Offline

Project : Employee Management System (ReactJS + Spring Boot + MySQL)

Batch No : IH-JAVA-039

Recorded Videos : Free

Placement : Placement Officer (Prasad Sir)

Mock Test : Every 15 Days

Lab Faculties : Prathamesh, Ravi, Gowthami

Whatsapp Group : Notes

Full Stack Java Development

(Full Stack Developer)

|

|-----------------------------------------------------------------------|

Frontend Technologies Backend Technologies

(Frontend Developer) (Backend Developer)

> HTML5 > Core Java

> CSS3 > Advanced Java

> JavaScript > JDBC

> Bootstrap5 > Servlets

> ReactJS > JSP

> Oracle DB

> SQL

> PL/SQL

> Frameworks

> Spring Boot

> Microservices

Programming language

=====================

Diagram: introduction1.1

A language which is used to communicate between user and computer is called programming language.

Programming language acts like a mediator or interface between user and computer.

Java

====

Object oriented programming language.

Platform independent programming language.

Case sensitive programming language.

Strongly typed checking language.

High level programming language.

Open source programming language.

1995 --> James Gosling --> Sun Micro System (Oracle Corporation)

Java software --> JDK software

C

==

Procedure oriented programming language.

Platform dependent programming language.

Case sensitive programming language.

Loosely typed checking language.

Middle level language (LOW + HIGH).

Interview Questions

==================

Q) What is Java?

Java is a object oriented, platform independent, case sensitive, strongly typed checking, high level , open source programming language developed by James Gosling in the year of 1995.

Courses Fee details

-------------------

JOIP : 30K (Course) + 30K (Placement)

Today : 27K + After placement (30K)

I&I : 75K (Course + Internship +Placement)

Today : 50K

Introduction 2 :-

Q) What is the difference between Python and Java?

Python Java

--------- --------

It is developed by Guido Van Rossum. It is developed by James Gosling.

It is a product of Microsoft. It is a product of Oracle Corporation.

It is a scripting language. It is a object oriented programming language.

It is a interpreted language. It is a compiled language.

It contains PVM. It contains JVM.

It is a dynamically typed language. It is a statically typed language.

ex: ex:

i = 10; int i = 10;

Performance is low. performance is high.

There is less security. It is highly secured.

Note:

-----

.py file --------------------------------> computer

(high level language)

.java file ----------------> .class file -----------> computer

(High level language) (Byte Code)

Note:

----

PVM

.py file ----------------------------------------> computer/machine

compiler JVM

.java file ----------> .class file --------------> JRE

(byte Code)

Here's a breakdown of the developers behind \*C, \*\*C++, \*\*Java, and \*\*Python\*:

1. \*C\*:

- \*Developer: \*\*Dennis Ritchie\*

- \*Year\*: Early 1970s

- \*Organization\*: Bell Labs

- \*Details\*: C was created to develop the UNIX operating system and has since become one of the most widely used programming languages, influencing many modern languages.

2. \*C++\*:

- \*Developer: \*\*Bjarne Stroustrup\*

- \*Year\*: Early 1980s

- \*Organization\*: Bell Labs

- \*Details\*: C++ was initially designed as an extension of the C programming language, with added support for object-oriented programming.

3. \*Java\*:

- \*Developer: \*\*James Gosling\*

- \*Year\*: 1995

- \*Organization\*: Sun Microsystems (now part of Oracle Corporation)

- \*Details\*: Java was developed with the principle of "write once, run anywhere," focusing on cross-platform compatibility.

4. \*Python\*:

- \*Developer: \*\*Guido van Rossum\*

- \*Year\*: 1991

- \*Organization\*: Python was initially developed as a hobby project, but Guido van Rossum worked at Centrum Wiskunde & Informatica (CWI) in the Netherlands at the time.

- \*Details\*: Python was designed to be easy to read and write, making it a popular language for both beginners and professionals.

These languages have significantly impacted software development and continue to be widely used across various domains.

Project

========

Technically, a project is a collection of modules.

We have following list of modules.

ex:

registration module

admin module

login module

report generation

payment module

and etc.

Every project contains two domains.

1) Technical domain

---------------------

Technical domain describes using which technology we developed our project.

ex:

Java

2) Functional domain

--------------------

Functional domain describes state of a project.

ex:

Healthcare domain

Banking domain

ERP domain

Insurance domain

and etc.

Interview Questions

===================

Q) How many modules are there in Java?

We have three modules in Java.

1) JSE/J2SE (Java Standard Edition)

2) JEE/J2EE (Java Enterprise Edition)

3) JME/J2ME (Java Micro Edition)

Class 1:-

Comments in Java

=================

Comments are created for documentation purpose.

Comments are used to improve readability of our code.

It is highly recommended to use comments in our regular programming.

Diagram: class1.1

Comments will not display in output because they won't compiled by the compiler.

In java, we have two types of comments.

1) Single Line Comment

2) Multiple Line Comment

1) Single Line Comment

-----------------------

It is used to comment a single line.

ex:

// comment here

2) Multiple Line Comment

-----------------------

It is used to comment multiple lines.

ex:

/\*

-

- comment here

-

\*/

ex:

---

//class declaration

class Test

{

//main method

public static void main(String[] args)

{

//variable declaration

int x = 10;

//output stmt

System.out.println(x);

}

}

Interview Questions

===================

Q) What is the difference between JDK, JRE and JVM ?

JDK

----

JDK stands for Java Development Kit.

JDK is a installable software which consist of Java Runtime Environment( JRE), Java Virtual Machine (JVM) , compiler (javac), interpreter (java), an archiever (.jar) , document generator (javadoc) and other tools needed for java application development.

JRE

----

JRE stands for Java Runtime Environment.

It provides very good environment to run java applications only.

JVM

---

JVM stands for Java Virtual Machine.

JVM is a interpreter which is used to execute our program line by line procedure and it

converts byte code to machine code.

Diagram: class1.2

Q) Is java platform independent or dependent?

Java is platform independent.

Diagram: class1.3

Q) Is JVM platform independent or dependent?

It is platform dependent.

Q) A .class file contains \_\_\_ code ?

byte code

Class 2:-

Naming conventions in Java

===========================

In java, uppercase letters will consider as different and lowercase letters will consider as

different . Hence we consider java is a case sensitive programming language.

As java is a case sensitive, we must and should follow naming conventions for following things.

ex:

classes

interfaces

variables

methods

keywords

packages and

constants

classes

--------

In java, a class name must and should starts with uppercase letter and if it contains multiple

words then each inner word must starts with initcap.

ex:

predefined classes userdefined classes

-------------------- ---------------------

System Test

FileWriter DemoApp

BufferedReader ExampleApp

File EmployeeDetails

StringBuilder StudentInfo

and etc. and etc.

interfaces

-----------

In java, an interface name must and should starts with uppercase letter and if it contains

multiple words then each inner word must and should starts with initcap.

ex:

predefined interfaces userdefined interfaces

--------------------- ---------------------

Runnable ITest

Serializable IDemoApp

ListIterator IStudentInfo

Enumeration IDepartment

and etc. and etc.

variables

---------

In java, a variable name must and should starts with lowercase letter and if it contains multiple

words then each inner word must and should starts with initcap.

ex:

predefined variables userdefined variables

--------------------- ---------------------

out i

in empId

err studName

length deptNo

and etc. and etc.

Methods

--------

In java, a method name must and should starts with lowercase letter and if it contains multiple

words then each inner word must and should starts with initcap.

ex:

predefined methods userdefined methods

------------------- --------------------

getPriority() getStudentInfo()

setName() setBillDetails()

getClass() calculateBillAmt()

hashCode() and etc.

toString()

and etc.

keywords

--------

In java, all keywords we need to declare under lowercase letters only.

ex:

predefined keywords

-------------------

if , else, do, while, break, public, static, void and etc.

packages

----------

In java, all packages we need to declare under lowercase letters only.

ex:

predefined packages userdefined packages

----------------- --------------------

java.lang com.ihub.www

java.util com.google.www

java.time com.qt.www

java.text and etc.

java.util.stream

java.sql

and etc.

constants

-----------

In java, all constants we need to declare under uppercase letters only.

ex:

predefined constants userdefined constants

----------------- -------------------

MAX\_PRIORITY LIMIT=10;

MIN\_PRIORITY

NORM\_PRIORITY

MAX\_VALUE

MIN\_VALUE

and etc.

Assignment

===========

1) class : QualityThought

2) Interface : IQualityThought

3) variable : qualityThought

4) Method : qualityThought()

5) package : com.qualitythought.www

6) constant : QUALITY\_THOUGHT

Interview Questions

===================

Q) Which package is a default package in java?

java.lang package

Q) What is package?

A package is a collection of classes and interfaces.

Class 3:-

History of Java

===============

In 1990, Sun Micro System took one project to develop a software called consumer electronic

device which can be controlled by a remote like setup. That time project was called Stealth

project and later it is renamed to Green project.

James Gosling, Mike Sheradin and Patrick Naughton were there to develop the project. They met

in a place called Aspan/Colarado to start the work with Graphic System. James Gosling thought

to use C and C++ languages to develop the project. But the problem what they have faced is

C and C++ languages are system dependent. Then James Gosling decided , why don't we create

our own programming language which is system independent.

In 1991, They have developed a programming language called an OAK. OAK means strength, itself

is a coffee seed name and it is a national tree for many contries like Germany , France,

USA and etc.

Later in 1995, they have renamed OAK to Java. Java is a island of an Indonasia where first

coffee of seed was produced. During the development of project they were consuming lot of

coffee's. Hence the symbol of java is a cup of coffee with saucer.

Interview Questions

===================

Q) Who is the creator of java?

James Gosling

Q) Java originally known as \_\_\_?

OAK

Q) In which year java was developed?

In 1995

Q) What are the features of Java?

We have following important features in java.

1) Simple 6) Robust

2) Object oriented 7) Portable

3) Platform independent 8) High secured

4) Architecture neutral 9) Dynamic

5) Multithreaded 10) Distributed and etc.

class 4:-

Java

=====

Version : Java 11

Software : JDK 11

Creator : James Gosling

Vendor : Oracle Corporation

Website : www.oracle.com/in/java

Open source : Open source

Tutorial : www.javatpoint.com

www.w3school.com

www.tutorialspoint.com

www.javaus.com

www.dzone.com

and etc.

Download link :

https://drive.google.com/file/d/1GtRLHXK4y3s97BH2UcYiJPNBaROR1DBV/view?usp=drive\_link

Steps to setup environmental variables

======================================

step1:

-------

Make sure JDK 11 installed successfully.

step2:

-----

Copy a "lib" directory from java\_home folder.

ex:

C:\Program Files\Java\jdk-11\lib

step3:

----

Paste "lib" directory in environmental variables.

ex:

right click to This PC --> properties --> Advanced system settings -->

environmental variables --> user variables --> click to new button -->

variable name : CLASSPATH

variable value : C:\Program Files\Java\jdk-11\lib;

---> ok.

system variables --> click to new button -->

variable name : path

variable value : C:\Program Files\Java\jdk-11\bin; --> ok --> ok --> ok.

step4:

-----

Open the command prompt and check environmental variables done perfectly or not.

ex:

cmd> javap

cmd> java -version

Steps to develop first java application

========================================

step1:

-----

Make sure JDK 11 installed successfully.

step2:

-----

Make sure environmental setup done perfectly.

step3:

------

Open the notepad and develop Hello World program.

ex:

class Test

{

public static void main(String[] args)

{

System.out.println("Hello World");

}

}

step4:

-----

Create a "javaprog" folder in 'E' drive.

step5:

-----

Save above program with same name as class inside "javaprog" location.

step6:

------

Open the command prompt from "javaprog" location.

step7:

------

Compile above program by using below command.

ex:

javac Test.java

|

filename

step8:

------

Run above program by using below command.

ex:

java Test

|

classname

class 5:-

Internal Architecture of JVM

============================

Diagram: class5.1

Java program contains java code instructions. Once if we compile java code instructions converts to byte code instructions in .class file.

Now JVM will invoke one module called classloader or sub system to load all the byte code instructions from .class file .The work of classloader is to check these byte code instructions are proper or not.

If they are not proper then it will refuse the execution. If they are proper then it will allocates the memories.

We have five types of memories.

1) Method Area

------------

It contains code of a class, code of a variable and code of a method.

2) Heap

--------

Our object creations will store in heap area.

3) Java Stack

--------------

Java methods will store in method area.

To execute those methods we required some memory and that memory will be allocated in java stack.

4) PC Register

--------------

It is a program counter register which is used to track the address of an instructions.

5) Native Method Stack

-----------------------

Java methods will execute in method area.

Similarly native methods will execute in native method stack.

But to execute native methods we required a program called Native method interface.

Execution Engine

----------------

Execution engine contains interpreter and JIT compiler.

Whenever JVM loads byte code instructions from .class file, it will uses interpreter and JIT compiler.

Interpreter is used to execute our program line by line procedure.

JIT compiler is used to increase the execution speed of our program.

Finally , JVM converts byte code to machine code.

Interview Question

===================

Q) What is native method in java?

A method which is developed by using some other language is called native method.

Q) How many memories are there in java?

We have five memories in java.

1) Method Area

2) Heap

3) Java Stack

4) PC Register

5) Native Method Stack

Q) What is JIT compiler?

It is a part of a JVM which is used to increase the execution speed of our program.

Q) How many classloaders are there in java?

We have three classloaders in java.

1) Bootstrap classloader

2) Extension classloader

3) Application/System classloader

Identifiers

===========

A name in java is called identifier.

It can be variable name, method name, class name or label name.

ex:

class Test

{

public static void main(String[] args)

{

int x = 10;

System.out.println(x);

}

}

Here Test, main ,String , args, x , System are identifiers.

Rules to declare an identifiers

--------------------------------

Rule1:

------

Identifier will accept following characters.

ex:

A-Z

a-z

0-9

\_

$

Rule2:

------

If we take other characters then we will get compile time error.

ex:

int emp$al;

String stud\_Name;

double emp#Fee; //invalid

Rule3:

------

Identifier must and should starts with alphabet, underscore or dollar symbol but not

with digits.

ex:

int \_empId; //valid

int $alary;//valid

int a1234; //valid

int 1abcd; //invalid

Rule4:

------

Every identifier is a case sensitive.

ex:

int number;

int NUMBER;

int NuMbEr;

Rule5:

-----

We can't take reserved words as an identifier name.

ex:

int if; //invalid

int else; //invalid

Rule6:

-----

There is no length limit for an identifier but it is not recommanded to take more then

15 characters.

Clsss 6:-

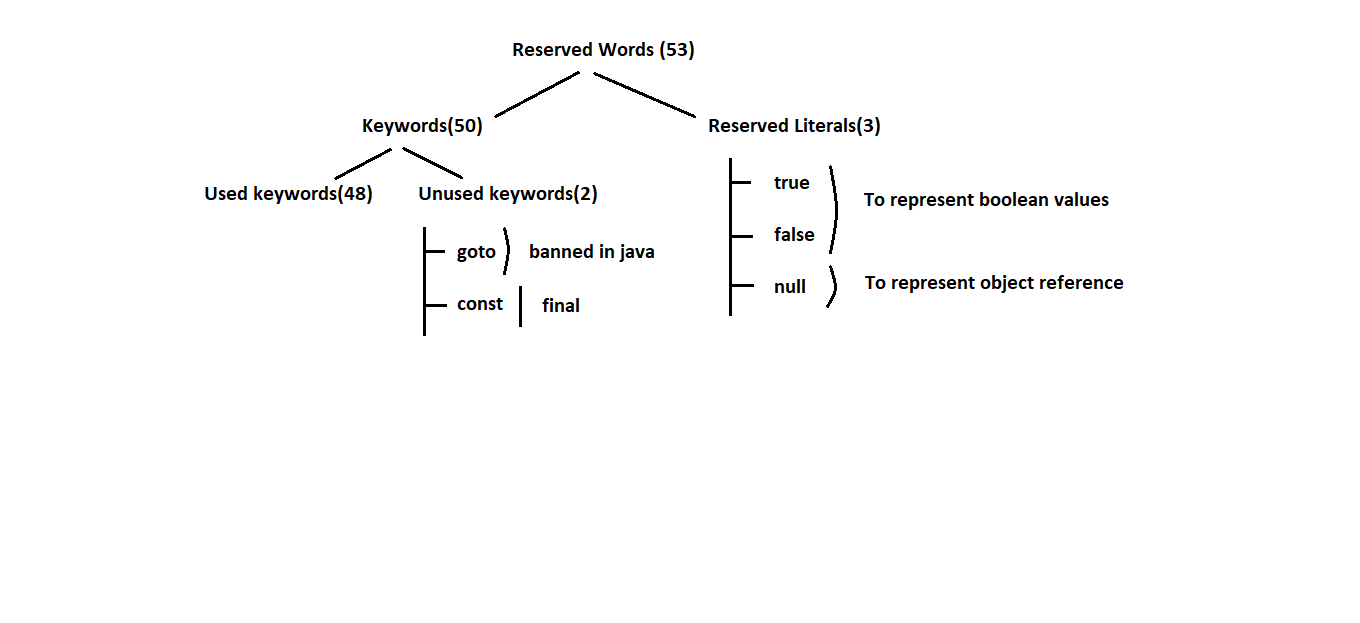
Reserved Words

===============

There are some identifiers which are reserved to associate some funcationality or meaning such type of identifiers are called reserved words.

Java supports 53 reserved words and it is divided into two types.

Diagram: class6.1



Used keywords with respect to class (7)

-------------------------------------

import

package

interface

enum

class

extends

implements

Used keywords with respect to object (4)

----------------------------------

new

instanceof

this

super

Used keywords with respect to datatypes (8)

-------------------------------

byte

short

int

long

float

double

boolean

char

Used keywords with respect to return type (1)

------------------------

void

Used keywords with respect to modifiers (11)

------------------------------

default

public

private

protected

static

final

abstract

synchronized

strictfp

transient

volatile

Used keywords with respect to flow control (9)

if

else

switch

case

break

continue

do

while

for

Used keywords with respect to exception handling (6)

--------------------

try

catch

throw

throws

finally

assert

Datatypes

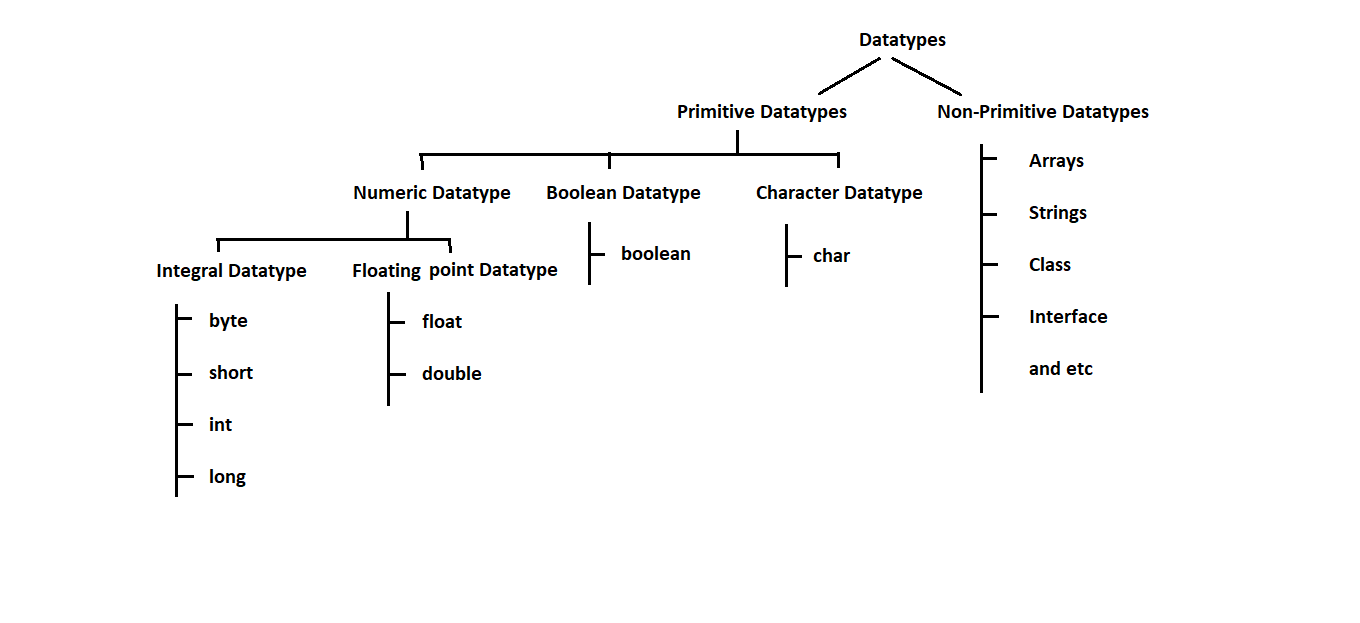
==========

Datatype describes what type of value we want to store inside a variable.

Datatype also tells how much memory has to be created for a variable.

In java, we have two types of datatypes

Diagram: class6.2



byte

-------

It is a smallest datatype in java.

Size : 1 byte (8 bits)

Range: -128 to 127 (-2^7 to 2^7-1)

ex:

1) byte b=10;

System.out.println(b); // 10

2) byte b=10.5;

System.out.println(b); // C.T.E

3) byte b="hi";

System.out.println(b); // C.T.E

short

------

It is a rarely used datatype in java.

Size : 2 bytes (16 bits)

Range : -32768 to 32767 (-2^15 to 2^15-1)

ex:

---

1) byte b=10;

short s=b;

System.out.println(s); // 10

2) short s=true;

System.out.println(s); // C.T.E

3) short s=10.56;

System.out.println(s); // C.T.E

int

-----

It is a mostly used datatype in java.

Size : 4 bytes (32 bits)

Range : -2147483648 to 2147483647 (-2^31 to 2^31-1)

ex:

1) int i=10.56;

System.out.println(i); // C.T.E

2) int i=true;

System.out.println(i); // C.T.E

3) int i="hi";

System.out.println(i); // C.T.E

4) int i='a';

System.out.println(i); // 97

Note:

-----

In java, every character has a universal unicode value.

ex:

a --> 97

A --> 65

Class 7:-

long

--------

If int datatype is not enough to hold large value then we need to use long datatype.

Size : 8 bytes (64 bits)

Range : -2^63 to 2^63-1

ex:

1) long l=10.56;

System.out.println(l); // C.T.E

2) long l="hi";

System.out.println(l); // C.T.E

3) long l=true;

System.out.println(l); // C.T.E

4) long l='a';

System.out.println(l); // 97

float double

------------- ----------------

If we need 4 to 6 decimal point of accuracy then If we need 14 to 16 decimal point of accuracy

we need to use float. then we need to use double.

Size: 4 bytes (32 bits) Size: 8 bytes (64 bits)

Range : -3.4e38 to 3.4e38 Range: -1.7e308 to 1.7e308

To declare a float value we need to suffix with To declare a double value we need to suffix

'f' or 'F'. with 'd' or 'D'.

ex: ex:

10.56f 10.56d

ex:

---

1) float f=10.56f;

System.out.println(f); // 10.56

2) float f=10;

System.out.println(f); // 10.0

3) float f='A';

System.out.println(f); // 65.0

4) float f="true";

System.out.println(f); // C.T.E

5) float f=false;

System.out.println(f); // C.T.E

ex:

---

1) double d=10.56d;

System.out.println(d); // 10.56

2) double d=10;

System.out.println(d); // 10.0

3) double d='A';

System.out.println(d); // 65.0

4) double d="true";

System.out.println(d); // C.T.E

5) double d=false;

System.out.println(d); // C.T.E

boolean

---------

It is used to represent boolean values either true or false.

Size : (Not Applicable) (1-bit)

Range: (Not Applicable)

ex:

1) boolean b="true";

System.out.println(b); // C.T.E

2) boolean b=TRUE;

System.out.println(b); // C.T.E

3) boolean b=true;

System.out.println(b); // true

char

-------

It is a single character which is enclosed in a single quotation.

Size : 2 bytes (16 bits)

Range: 0 to 65535

ex:

1) char ch='a';

System.out.println(ch); //a

2) char ch="p";

System.out.println(ch); // C.T.E

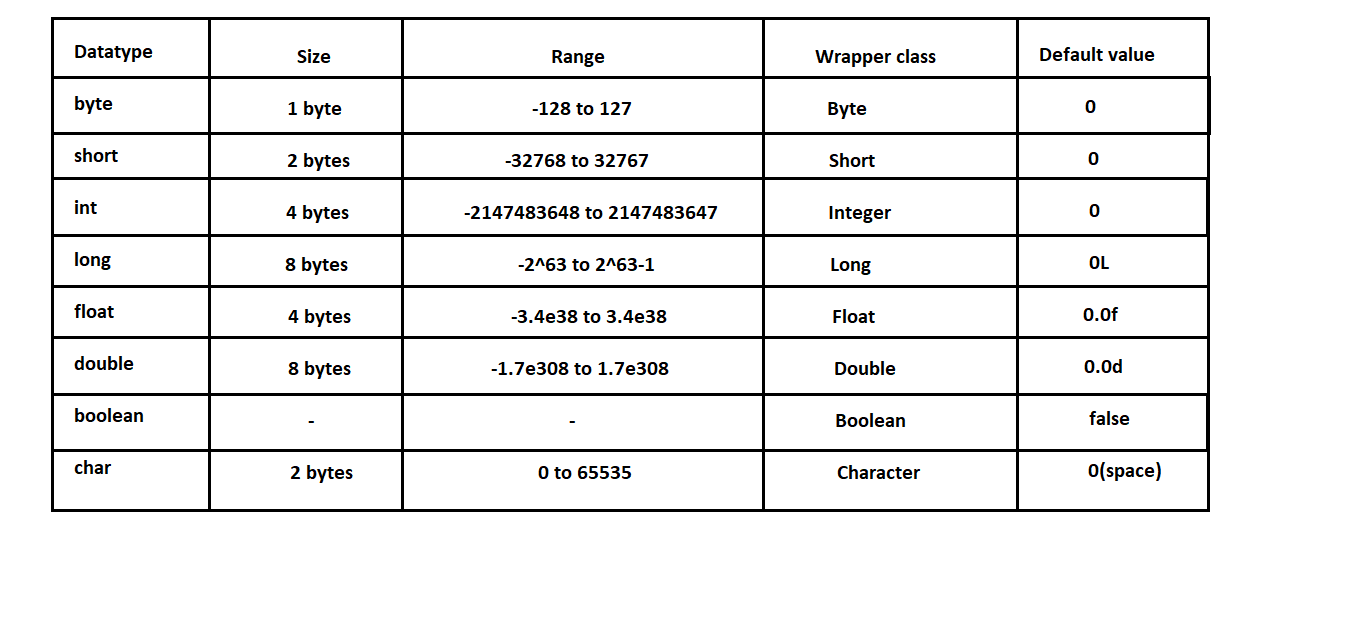
3) char ch='ab';

System.out.println(ch); // C.T.E

4) char ch=65;

System.out.println(ch); // A

Diagram: class7.1



Interview Question

===================

Q) Is java purely object oriented or not?

No, java will not consider as purely object oriented programming language because it does not

support many OOPS concepts like multiple inheritance, operator overloading and more ever we

depends upon primitive datatypes which are non-objects.

Q) Write a java program to display range of byte datatype?

byte : -128 to 127

class Test

{

public static void main(String[] args)

{

System.out.println(Byte.MIN\_VALUE);

System.out.println(Byte.MAX\_VALUE);

}

}

Q) Write a java program to display range of short datatype?

short : -32768 to 32767

ex:

class Test

{

public static void main(String[] args)

{

System.out.println(Short.MIN\_VALUE);

System.out.println(Short.MAX\_VALUE);

}

}

Class 8:-

Types of variables

==================

A name which is given to a memory location is called variable.

Purpose of variable is used to store the data.

In java, we have two types of variables.

1) Primitive variables

2) Reference variables

1) Primitive variables

-----------------------

Primitive variables are used to represent primitive values.

2) Reference variables

----------------------

Reference variables are used to represent object.

ex:

Student s=new Student();

|

reference variable

Based on the position and execution these variables are divided into three types.

1) Instance variables / Non-static variables

2) Static variables / Global variables

3) Local variables / Temperory variables / Automatic variables

1) Instance variables

----------------------

A value of a variable which is varied (changes) from object to object is called instance variable.

Instance variable will be created at the time of object creation and it will destroy at the time of

object destruction. Hence scope of instance variable is same as scope of an object.

Instance variable will store in heap area as a part of an object.

Instance variable must and should declare immediately after the class but not inside methods,

blocks and constructors.

Instance variable we can access directly from instance area but we can't access directly from

static area.

To access instance variable from static area we need to create object reference.

ex:1

-----

class Test

{

//instance variable

int i=10;

public static void main(String[] args)

{

System.out.println(i); // C.T.E

}

}

ex:2

----

class Test

{

//instance variable

int i=10;

public static void main(String[] args)

{

Test t=new Test();

System.out.println(t.i); // 10

}

}

Note:

----

If we won't initialize any value to instance variable then JVM will initialized default values.

ex:3

-----

class Test

{

//instance variable

boolean b;

public static void main(String[] args)

{

Test t=new Test();

System.out.println(t.b);// false

}

}

ex:4

----

class Test

{

//instance variable

int i=10;

public static void main(String[] args)

{

Test t1=new Test();

Test t2=new Test();

System.out.println(t1.i);//10

System.out.println(t2.i);//10

t1.i=100;

System.out.println(t1.i);//100

System.out.println(t2.i);//10

}

}

ex:5

-----

class Test

{

public static void main(String[] args)

{

//calling

Test t=new Test();

t.m1();

}

//non-static method

public void m1()

{

System.out.println("instance-method");

}

}

If we write the logic in a seperate instance method then we can reuse the bussiness logic.

ex:6

----

class Test

{

public static void main(String[] args)

{

Test t=new Test();

t.m1();

t.m1();

t.m1();

}

//non-static method

public void m1()

{

System.out.println("instance-method");

}

}

ex:7

-----

class Test

{

public static void main(String[] args)

{

Test t=new Test();

t.sum();

}

// non static method

public void sum()

{

int a=50;

int b=100;

int c=a+b;

System.out.println(c);

}

}

o/p : 150

class 9:-

2) static variables

--------------------

A value of a variable which is not varied(changes) from object to object is called static variable.

A static variable will be created at the time of classloading and it will destroy at the time of

classunloading . Hence scope of static variable is same as scope of a .class file.

Static variables will store in method area.

Static variable must and should declare immediately after the class using static keyword but not inside methods, blocks and constructors.

Static variable we can access directly from static area as well as instance area.

Static variable we can access by using object reference and class name.

ex:1

-----

class Test

{

//static variable

static int i=10;

public static void main(String[] args)

{

System.out.println(i); //10

Test t=new Test();

System.out.println(t.i); //10

System.out.println(Test.i);//10

}

}

Note:

-----

If we won't initialize any value to static variable JVM will initialized default values.

ex:2

-----

class Test

{

//static variable

static String s;

public static void main(String[] args)

{

System.out.println(s); // null

Test t=new Test();

System.out.println(t.s); // null

System.out.println(Test.s); // null

}

}

ex:3

-----

class Test

{

//static variable

static int i=10;

public static void main(String[] args)

{

Test t1=new Test();

Test t2=new Test();

System.out.println(t1.i);//10

System.out.println(t2.i);//10

t1.i=100;

System.out.println(t1.i); //100

System.out.println(t2.i); //100

}

}

ex:4

-----

class Test

{

public static void main(String[] args)

{

m1();

Test t=new Test();

t.m1();

Test.m1();

}

//static method

public static void m1()

{

System.out.println("static-method");

}

}

3) Local variables

--------------------

To meet temperory requirements a programmer will declare some variables inside methods, blocks

and constructors such type of variables are called local variables.

Local variable will be created at the time of execution block and it will destroy when execution

block is executed.Hence scope of local variable is same as scope of a execution block where it is

declared.

Local variables will store in Java stack.

ex:1

-----

class Test

{

public static void main(String[] args)

{

//local variable

int i=10;

System.out.println(i); //10

}

}

Note:

-------

If we won't initialize any value to local variable then JVM will not initialized default values.

ex:2

----

class Test

{

public static void main(String[] args)

{

//local variable

int i;

System.out.println(i); //

}

}

o/p:

C.T.E : variable i might not have been initialized

Interview Question

==================

Q) Jack and John both are best friends in a town.While going to school they saw one begger.Jack and john both decided to help that needy person.Jack gave Rs.50 from his pocket money and John gave Rs.100 from his bag.Write a java console to find out total contribution for a poor guy?

class Test

{

public static void main(String[] args)

{

sum();

}

//static method

public static void sum()

{

int a=50;

int b=100;

int c=a+b;

System.out.println(c);

}

}

Class 10:-

Main method

============

Our program contains main method or not.

Either it is properly declared or not.

It is not a responsibility of a compiler to check. It is a liability of a JVM to look for main method always at runtime.

If JVM won't find main method then it will throw one runtime error called main method not found.

JVM always look for main method with following signature.

signature:

--------

public static void main(String[] args)

If we perform any changes in above signature then JVM Will throw one runtime error called main method not found.

Q) Explain main method in java?

public

------

JVM wants to call main method from anywhere.

static

------

JVM wants to call main method without using object reference.

void

-----

Main method does not return anything to JVM.

main

----

It is a identifier given to a main method.

String[] args

-----------

It is a command line argument.

We can perform following changes in main method.

1) Order of modifiers is not important. Incase of public static we can declare static public also.

ex:

static public void main(String[] args)

2) We can change String[] in following acceptable formats.

ex:

public static void main(String[] args)

public static void main(String []args)

public static void main(String args[])

3) We can change String[] with var-arg parameter.

ex:

public static void main(String... args)

4) We can replace args with any java valid identifier.

5) Main method will accept following modifiers.

ex:

synchronized

strictfp

final

Command line arguments

======================

Arguments which are passing through command prompt such type of arguments are called command line

arguments.

In command line arguments we need to pass input values at runtime command.

ex:

javac Test.java

java Test 101 raja M 1000.0

| | | |\_\_\_\_ args[3]

| | |\_\_\_\_\_\_\_\_ args[2]

| |\_\_\_\_\_\_\_\_\_\_\_\_\_ args[1]

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ args[0]

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println(args[0]);

System.out.println(args[1]);

System.out.println(args[2]);

System.out.println(args[3]);

}

}

System.out.println()

=====================

It is a output statement in java.

Whenever we want to display any data or user defined statements then we need to use output stmt.

syntax:

-----

static variable

|

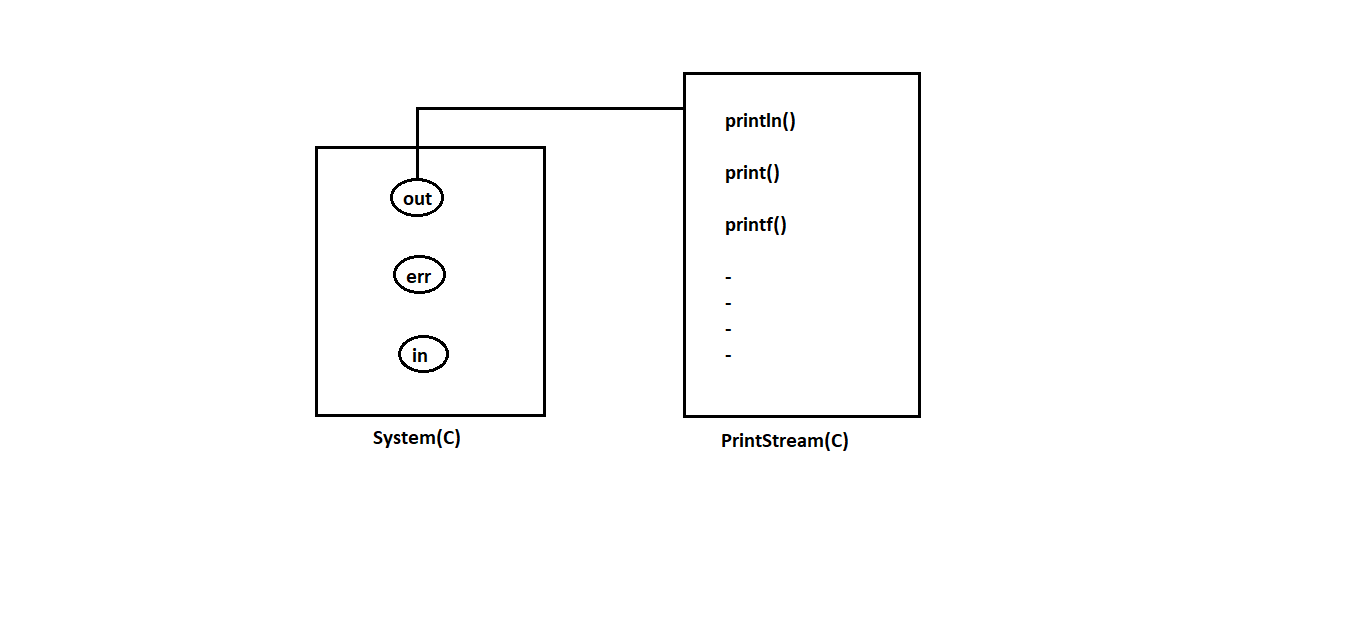
System.out.println();

| |

predefined predefined method

final class

Diagram: class10.1



Class 11:-

ex:

----

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

System.out.print("stmt2");

System.out.printf("stmt3");

}

}

Various ways to display the data in java

----------------------------------------

1)

System.out.println("Hello World");

2)

int i=10;

System.out.println(i);

System.out.println("The value is ="+i);

3)

int i=10,j=20;

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

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

4) int i=1,j=2,k=3;

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

Fully Qualified Name

====================

Fully qualified name means we need to declare a class or interface along with package name.

It is used to increase the readability of our code.

ex:

---

class Test

{

public static void main(String[] args)

{

java.util.Date d =new java.util.Date();

System.out.println(d);

}

}

Import statements

=================

Whenever we use import statement we should not use fully qualified name.

Using shortname also we can achieve.

In java, we have three import statements.

1) Explicit class import

2) Implicit class import

3) Static import

1) Explicit class import

-------------------------

This type of import statement is highly recommanded to use because it will improve readability of our code.

ex:

---

import java.time.LocalDate;

import java.time.LocalTime;

class Test

{

public static void main(String[] args)

{

LocalDate date=LocalDate.now();

System.out.println(date);

LocalTime time=LocalTime.now();

System.out.println(time);

}

}

2) Implicit class import

--------------------

This type of import statement is not recommanded to use because it will reduce the readability of our code.

ex:

---

import java.time.\*;

class Test

{

public static void main(String[] args)

{

LocalDate date=LocalDate.now();

System.out.println(date);

LocalTime time=LocalTime.now();

System.out.println(time);

}

}

static important

----------------

Using static import we can call static members (static variables and static methods) directly.

Often use of static import makes our program complex and unreadable.

ex:

---

import static java.lang.System.\*;

class Test

{

public static void main(String[] args)

{

out.println("stmt1");

out.println("stmt2");

out.println("stmt3");

}

}

ex:

---

import static java.lang.System.\*;

class Test

{

public static void main(String[] args)

{

out.println("stmt1");

exit(0);

out.println("stmt2");

}

}

Editplus Free Download

======================

Download link : https://www.editplus.com/download.html

Java Basic Programs

====================

Q) Write a java program to perform sum of two numbers?

import java.util.Scanner;

class Example1

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();

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

int b=sc.nextInt();

int c = a + b;

System.out.println("sum of two numbers is ="+c);

}

}

Q) Write a java program to perform sum of two numbers without using third variable?

import java.util.Scanner;

class Example2

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();

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

int b=sc.nextInt();

System.out.println("sum of two numbers is ="+(a+b));

}

}

Assignment

===========

Q) Write a java program to find out of area of a rectangle?

Class 12:-

Q) Write a java program to display square of a given number?

input:

5

output:

25

ex:

import java.util.Scanner;

class Example3

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); //5

//logic

int square=n\*n;

System.out.println("square of a given number is ="+square);

}

}

Q) Write a java program to display cube of a given number?

input:

5

output:

125

ex:

---

import java.util.Scanner;

class Example4

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); //5

//logic

int cube=n\*n\*n;

System.out.println("cube of a given number is ="+cube);

}

}

Q) Write a java program to perform area of a circle?

input:

r = 5

output:

78.5

ex:

---

import java.util.Scanner;

class Example5

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the radius :");

int r=sc.nextInt();

//logic

float area=3.14f\*r\*r;

System.out.println("area of a circle is ="+area);

}

}

Q) Write a java program to perform perimeter of a circle?

import java.util.Scanner;

class Example6

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the radius :");

int r=sc.nextInt();

//logic

float perimeter=2\*3.14f\*r;

System.out.println("perimeter of a circle is ="+perimeter);

}

}

Q) Write a java program to perform swapping of two numbers?

output:

Before swapping a = 10 and b = 20

After swapping a = 20 and b = 10

ex:

import java.util.Scanner;

class Example7

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt(); //10

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

int b=sc.nextInt(); //20

System.out.println("Before swapping a = "+a+" and b = "+b);

//logic

int temp=a;

a=b;

b=temp;

System.out.println("After swapping a = "+a+" and b = "+b);

}

}

Q) Write a java program to perform swapping of two numbers without using third variable?

import java.util.Scanner;

class Example8

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt(); //10

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

int b=sc.nextInt(); //20

System.out.println("Before swapping a = "+a+" and b = "+b);

//logic

a = a + b;

b = a - b;

a = a - b;

System.out.println("After swapping a = "+a+" and b = "+b);

}

}

Q) Write a java program to find out 10% of TDS from given salary?

import java.util.Scanner;

class Example9

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the salary :");

int salary=sc.nextInt();

float tax=(float)salary\*10/100;

System.out.println("10 percent of TDS is ="+tax);

}

}

Q) Write a java program to convert CGPA to percentage?

import java.util.Scanner;

class Example10

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the CGPA :");

float cgpa=sc.nextFloat();

float percent=cgpa\*9.5f;

System.out.println("CGPA to percentage is ="+percent);

}

}

Assignment

==========

Q) Write a java program to accept six marks of a student then find out total and average?

Typecasting in Java

====================

The process of converting from one datatype to another datatype is called typecasting.

In java, typecasting can be done in two ways.

1) Implicit typecasting

2) Explicit typecasting

1) Implicit typecasting

-----------------------

If we want to store small value into a bigger variable then we need to use implicit typecasting.

A compiler is responsible to perform implicit typecasting.

There is no possibility to loss the information.

It is also known as Widening or Upcasting.

We can perform implicit typecasting as follow.

ex:

byte --> short

-->

int --> long --> float --> double

-->

char

ex:

---

class Test

{

public static void main(String[] args)

{

byte b=10;

int i=b;

System.out.println(i);//10

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

char ch='a';

long l=ch;

System.out.println(l);//97

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

double d=i;

System.out.println(d);//10.0

}

}

1) Explicit typecasting

-----------------------

If we want to store big value into a small variable then we need to use explicit typecasting.

A programmer is responsible to perform implicit typecasting.

There is a possibility to loss the information.

It is also known as Norrowing or Downcasting.

We can perform explicity typecasting as follow.

ex:

byte <-- short

<--

int <-- long <-- float <-- double

<--

Char

class 13 :-

ex:

byte <-- short

<--

int <-- long <-- float <-- double

<--

char

ex:

---

class Test

{

public static void main(String[] args)

{

double d=10.5d;

int i=(int)d;

System.out.println(i); // 10

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=65;

char ch=(char)i;

System.out.println(ch); // A

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=130;

byte b=(byte)i;

System.out.println(b); // -126

}

}

Assignment

===========

Q) Write a java program to accept six marks of a student then find out total and average?

ex:

class Test

{

public static void main(String[] args)

{

int m1=89,m2=45,m3=65,m4=74,m5=49,m6=58;

int total=m1+m2+m3+m4+m5+m6;

float avg=(float)total/6;

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

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

}

}

Q) Write a java program to perform cube of a given number?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); //5

int cube=(int)Math.pow(n,3);

System.out.println("cube of a given number is ="+cube);

}

}

Types of blocks in java

=======================

A block is a set of statements which is enclosed in a curly braces i.e {}.

In java, we have three types of blocks.

1) Instance block

2) Static block

3) Local block

1) Instance block

----------------

Instance block is used to initialize the values to instance variables.

Instance block must and should declare immediately after the class but not inside methods and constructors.

Instance block will execute when create an object.

We can declare instance block as follow.

syntax:

//instance block

{

-

- //set of statements

-

}

ex:

----

class Test

{

//instance block

{

System.out.println("instance-block");

}

public static void main(String[] args)

{

System.out.println("main-method");

}

}

o/p:

main-method

ex:

---

class Test

{

//instance block

{

System.out.println("instance-block");

}

public static void main(String[] args)

{

System.out.println("main-method");

Test t=new Test();

}

}

o/p:

main-method

instance-block

ex:

---

class Test

{

//instance block

{

System.out.println("instance-block");

}

public static void main(String[] args)

{

Test t1=new Test();

System.out.println("main-method");

Test t2=new Test();

}

}

o/p:

instance-block

main-method

instance-block

ex:

---

class Test

{

//instance variable

int i;

//instance block

{

i=100;

}

public static void main(String[] args)

{

Test t=new Test();

System.out.println(t.i);//100

}

}

2) static block

----------------

Static block is used to initialize the values to static variables.

Static block must and should declare immediately after the class using static keyword but not inside methods and constructors.

Static block will execute at the time of classloading.

We can declare static block as follow.

syntax:

//static block

static

{

-

- //set of statements

-

}

ex:

---

class Test

{

//static block

static

{

System.out.println("static-block");

}

public static void main(String[] args)

{

System.out.println("main-method");

}

}

o/p:

static-block

main-method

ex:

----

class Test

{

//instance block

{

System.out.println("instance-block");

}

//static block

static

{

System.out.println("static-block");

}

public static void main(String[] args)

{

Test t=new Test();

System.out.println("main-method");

}

}

o/p:

static-block

instance-block

main-method

ex:

---

class Test

{

//static variable

static int i;

//static block

static

{

i=200;

}

public static void main(String[] args)

{

System.out.println(i); //200

}

}

Class 14:-

3) local block

================

Local block is used to initialize the local variables.

Local block must and should declare inside the methods.

Local block will execute just like a normal statement.

syntax:

------

//local block

{

-

- //set of statements

-

}

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

//local block

{

System.out.println("stmt2");

}

System.out.println("stmt3");

}

}

o/p:

stmt1

stmt2

stmt3

ex:

----

class Test

{

public static void main(String[] args)

{

//local variable

int i;

//local block

{

i=300;

}

System.out.println(i); // 300

}

}

Q) Can we execute java program without main method?

Yes, Till 1.6 version it is possible to execute java program without main method using static block. But from 1.7 version onwards it is not possible to execute java program without main method.

ex:

---

class Test

{

//static block

static

{

System.out.println("Hello World");

System.exit(0);

}

}

Operators

==========

Operator is a symbol which is used to perform some operations on operands.

ex:

c = a + b;

Here + and = are operators

Here a, b and c are operands.

It can be arithmetic operation, logical operation, bitwise operation, relational operation and etc.

We have following list of operators in java.

1) Assignment operators

2) Ternary operators

3) Logical operators

4) Bitwise operators

5) Arithmetic operators

6) Relational operators

7) Shift operators

8) Unary operators

1) Assignment operators

-----------------------

class Test

{

public static void main(String[] args)

{

int i=10;

i=20;

i=30;

System.out.println(i);//30

}

}

Note:

-----

Reinitialization is possible in java.

ex:

---

class Test

{

public static void main(String[] args)

{

final int i=10;

i=20;

i=30;

System.out.println(i);//C.T.E

}

}

Note:

-----

Final variables can't be modified.

ex:

----

class Test

{

public static void main(String[] args)

{

int i=1,2,3,4,5;

System.out.println(i); //C.T.E

}

}

ex:

---

class Test

{

//global variable

static int i=10;

public static void main(String[] args)

{

//local variable

int i=20;

System.out.println(i); // 20

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

i+=20; // i = i + 20;

System.out.println(i); // 30

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

i-=20; // i = i - 20;

System.out.println(i); // -10

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

i\*=20; // i = i \* 20;

System.out.println(i); // 200

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=5;

i/=2; // i = i / 2;

System.out.println(i); //2

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=5;

i/=20;

System.out.println(i); //0

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=5;

i%=2;

System.out.println(i); //1

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=5;

i%=20;

System.out.println(i); // 5

}

}

2) Ternary operator

--------------------

syntax:

------

(condition)?value1:value2;

ex:

---

class Test

{

public static void main(String[] args)

{

boolean b=(5>2)?true:false;

System.out.println(b);//true

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=(5>20)?1:0;

System.out.println(i);//0

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

String s=(true)?"Hi":"Bye";

System.out.println(s); // Hi

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

char ch=(false)?'t':'f';

System.out.println(ch); // f

}

}

Q) Write a java program to find out greatest of two numbers?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();

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

int b=sc.nextInt();

//logic

int max=(a>b)?a:b;

System.out.println("Greatest of two numbers is ="+max);

}

}

Q) Write a java program to find out greatest of three numbers?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();

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

int b=sc.nextInt();

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

int c=sc.nextInt();

//logic

int max=(a>b)?((a>c)?a:c):((b>c)?b:c);

System.out.println("Greatest of three numbers is ="+max);

}

}

Class 15:-

3) Logical operators

====================

Logical AND operator (&&)

--------------------------

Logical AND operator deals with boolean value either true or false.

Truth table

-----------

T T = T

T F = F

F T = F

F F = F

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println(true && true); //true

System.out.println(true && false); //false

System.out.println(false && true); // false

System.out.println(false && false); // false

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

boolean b= (5>2) && (6<10);

System.out.println(b);//true

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

boolean b= (5>2) && (6<1);

System.out.println(b);//false

}

}

Logical OR operator (||)

-------------------------

Logical OR operator deals with boolean values either true or false.

Truth table

----------

T T = T

T F = T

F T = T

F F = F

ex:

class Test

{

public static void main(String[] args)

{

System.out.println(true || true);//true

System.out.println(true || false);//true

System.out.println(false || true);//true

System.out.println(false || false);//false

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

boolean b=(6>10) || (5<2);

System.out.println(b); // false

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

boolean b=(6>1) || (5<2);

System.out.println(b); // true

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

boolean b=(7>2) && (5<10) || (6>10);

System.out.println(b); // true

}

}

Logical NOT operator (!)

-----------------------

ex:

---

class Test

{

public static void main(String[] args)

{

boolean b= !(5>2);

System.out.println(b); // false

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

boolean b= !(5>20);

System.out.println(b); // true

}

}

How to convert decimal to binary number

----------------------------------------

10 - decimal number

1010 - binary number

2|10

---- 0

2|5

---- 1

2|2

---- 0 ^

1 |

-----------------

1010

How to convert binary to decimal number

--------------------------------------

0101 - binary number

5 - decimal number

0101

<--

1\*1 + 0\*2 + 1\*4 + 0\*8

1 + 0 + 4 + 0 = 5

4) Bitwise operators

=====================

Bitwise AND operator (&)

--------------------------

Bitwise AND operator deals with binary numbers.

Truth table

-------------

T T = T

T F = F

F T = F

F F = F

ex:

---

class Test

{

public static void main(String[] args)

{

int a=10,b=5;

int c = a & b;

System.out.println(c); //0

}

}

/\*

10 - 1010

5 - 0101

-----------

& - 0000

\*/

ex:

---

class Test

{

public static void main(String[] args)

{

int a=10,b=15;

int c = a & b;

System.out.println(c); //10

}

}

/\*

10 - 1010

15 - 1111

---------

& - 1010 <---

0\*1 + 1\*2 + 0\*4 + 1\*8

0 + 2 + 0 + 8 = 10

\*/

Bitwise OR operator (|)

--------------------------

Bitwise OR operator deals with binary numbers.

Truth table

-------------

T T = T

T F = T

F T = T

F F = F

ex:

---

class Test

{

public static void main(String[] args)

{

int a=2,b=3;

int c = a | b;

System.out.println(c); // 3

}

}

/\*

2 - 0010

3 - 0011

---------

| - 0011 <--

1\*1 + 1\*2 + 0\*4 + 0\*8

1 + 2 + 0 + 0 = 3

\*/

ex:

---

class Test

{

public static void main(String[] args)

{

int a=10,b=5;

int c = a | b;

System.out.println(c); // 15

}

}

/\*

10 - 1010

5 - 0101

----------

| - 1111

\*/

Bitwise XOR operator (^)

--------------------------

Bitwise XOR operator deals with binary numbers.

Truth table

-------------

T T = F

T F = T

F T = T

F F = F

ex:

---

class Test

{

public static void main(String[] args)

{

int a=10,b=15;

int c = a ^ b;

System.out.println(c); // 5

}

}

/\*

10 - 1010

15 - 1111

----------

^ - 0101

\*/

Bitwise NOT operator (~)

-------------------------

ex:

---

class Test

{

public static void main(String[] args)

{

int i=~10;

System.out.println(i); // -11

}

}

ex:

----

class Test

{

public static void main(String[] args)

{

int i=~23;

System.out.println(i); // -24

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=~(-12);

System.out.println(i); // 11

}

}

5) Arithmetic operators

======================

% - modules

/ - division

\* - multiplication

+ - addition

- - subtraction

ex:

---

class Test

{

public static void main(String[] args)

{

int i= 6+7/2+6%3+7\*4+9/10+7%20+6-10;

System.out.println(i); //40

}

}

/\*

6 + 7/2 + 6%3 + 7\*4 + 9/10 + 7%20 + 6 - 10

6 + 3 + 0 + 28 + 0 + 7 + 6 - 10

50 - 10

40

\*/

6) Relational operators

------------------------

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println(10 > 20); //false

System.out.println(10 >= 20); //false

System.out.println(10 < 20); //true

System.out.println(10 <= 10); //true

System.out.println(10 == 10);//true

System.out.println(10 == 20);//false

System.out.println(10 != 20); //true

System.out.println(10 != 10); //false

}

}

Class 16:-

7) Shift operators

===================

Right shift operator (>>)

--------------------------

10 >> 1 = 10/2

10 >> 2 = 10/4

10 >> 3 = 10/8

10 >> 4 = 10/16

ex:

---

class Test

{

public static void main(String[] args)

{

int i= 10 >> 3; // 10/8

System.out.println(i); //1

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i= 100 >> 6; //100/ 64

System.out.println(i); //1

}

}

Left shift operator (<<)

--------------------------

10 << 1 = 10\*2

10 << 2 = 10\*4

10 << 3 = 10\*8

10 << 4 = 10\*16

ex:

---

class Test

{

public static void main(String[] args)

{

int i= 10 << 3; // 10\*8 =80

System.out.println(i); // 80

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i= 100 << 2; // 100 \* 2\*2

System.out.println(i); // 400

}

}

8)Unary operators

=================

Increment/Decrement operators (++/--)

-----------------------------------

We have two types of increment operators

1) Post increment

ex:

i++;

2) Pre increment

ex:

++i;

We have two types of decrement operators.

1) Post decrement

ex:

i--;

2) Pre decrement

ex:

--i;

POST Increment/Decrement

------------------------

Rule1: First Take

Rule2: Then Change

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

i++;

System.out.println(i); // 11

}

}

ex:

----

class Test

{

public static void main(String[] args)

{

int i=10;

System.out.println(i++); //10

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

int j= i++ + i++; //10 + 11

System.out.println(i+" "+j); //12 21

}

}

ex:

----

class Test

{

public static void main(String[] args)

{

int i=10;

int j=i-- + i--; //10 + 9

System.out.println(i+" "+j); // 8 19

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

int j=i++ + i-- + i++; //10 + 11 + 10

System.out.println(i+" "+j); // 11 31

}

}

Pre increment/decrement

------------------------

Rule1 : First Change

Rule2 : Then Take

ex:

----

class Test

{

public static void main(String[] args)

{

int i=10;

++i;

System.out.println(i); //11

}

}

ex:

----

class Test

{

public static void main(String[] args)

{

int i=10;

System.out.println(++i); // 11

}

}

ex:

----

class Test

{

public static void main(String[] args)

{

int i=10;

int j=++i + ++i; //11 + 12

System.out.println(i+" "+j); //12 23

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

int j=--i + --i; // 9 + 8

System.out.println(i+" "+j); // 8 17

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

System.out.println(i++ + ++i); //10 + 12 = 22

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=100;

100++;

System.out.println(i);//C.T.E

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

int i=10;

System.out.println(++(i++));//C.T.E

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

byte b=127;

b++;

System.out.println(b);//-128

}

}

Control Statements

==================

Control statement enables the programmer to control flow of a program.

Control statement allows the programmer to make decisions, to jump from one section of code to another section and to execute the code repeatedly.

In java, we have four control statements.

1) Decision Making statement

2) Selection statement

3) Iteration statement

4) Jump statement

1) Decision Making statement

----------------------------

Decision making statement is used to declare conditions in our program.

Decision making statement is possible by using following ways.

i) if stmt

ii) if else stmt

iii) if else if ladder

iv) nested if stmt

i) if stmt

-----------

It is used to execute the source code only if our condition is true.

syntax:

-------

if(condition)

{

-

- //code to be execute

-

}

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

if(5>2)

{

System.out.println("stmt2");

}

System.out.println("stmt3");

}

}

o/p:

stmt1

stmt2

stmt3

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

if(!(5>2))

{

System.out.println("stmt2");

}

System.out.println("stmt3");

}

}

o/p:

stmt1

stmt3

ex:

---

class Test

{

public static void main(String[] args)

{

if(!(5>2))

System.out.println("stmt1");

System.out.println("stmt2");

System.out.println("stmt3");

}

}

o/p:

stmt2

stmt3

Class 17:-

Q) Write a java program to find out greatest of two numbers?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();

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

int b=sc.nextInt();

if(a>b)

System.out.println(a+" is greatest");

if(b>a)

System.out.println(b+" is greatest");

}

}

Q) Write a java program to find out greatest of three numbers?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();

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

int b=sc.nextInt();

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

int c=sc.nextInt();

if((a>b) && (a>c))

System.out.println(a+" is greatest");

if((b>a) && (b>c))

System.out.println(b+" is greatest");

if((c>a) && (c>b))

System.out.println(c+" is greatest");

}

}

ii) if else stmt

=================

It will execute the source code either our condition is true or false.

syntax:

------

if(condition)

{

- //code to be execute if cond is true

}

else

{

- //code to be execute if cond is false

}

ex:

----

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

if(5>10)

{

System.out.println("stmt2");

}

else

{

System.out.println("stmt3");

}

System.out.println("stmt4");

}

}

o/p:

stmt1

stmt3

stmt4

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

if(5>1)

{

System.out.println("stmt2");

}

else

{

System.out.println("stmt3");

}

System.out.println("stmt4");

}

}

o/p:

stmt1

stmt2

stmt4

Q) Write a java program to check given age is eligible to vote or not?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int age=sc.nextInt();

if(age>=18)

System.out.println("U r eligible to vote");

else

System.out.println("U r not eligible to vote");

}

}

Q) Write a java program to check given number is even or odd?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

if(n%2==0)

System.out.println("It is even number");

else

System.out.println("It is odd number");

}

}

Q) Write a java program to check given number is odd or not?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

if(n%2!=0)

System.out.println("It is odd number");

else

System.out.println("It is not odd number");

}

}

Q) Write a java program to check given year is a leap year or not?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the year :");

int year=sc.nextInt();

if(year%4==0 && year%100!=0 || year%400==0)

System.out.println("It is a leap year");

else

System.out.println("It is not a leap year");

}

}

Q) Write a java program to check given number is positive or negative?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

if(n==0)

{

System.out.println("It is not a positive or negative");

System.exit(0);

}

if(n>0)

System.out.println("It is positive number");

else

System.out.println("It is negative number");

}

}

Class 18:-

iii) if else if ladder

======================

It will execute the source code based on multiple conditions.

syntax:

------

if(cond1)

{

- //code to be execute if cond1 is true

}

else if(cond2)

{

- //code to be execute if cond2 is true

}

else if(cond3)

{

- //code to be execute if cond3 is true

}

else

{

- //code to be execute if all conditions are false

}

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the option :");

int option=sc.nextInt();

if(option==100)

System.out.println("It is a police number");

else if(option==103)

System.out.println("It is enquiry number");

else if(option==108)

System.out.println("It is emergency number");

else

System.out.println("Invalid option");

}

}

Q) Write a java program to check given alphabet is a vowel or not?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the alphabet :");

char ch=sc.next().charAt(0);

if(ch=='a'|| ch=='A')

System.out.println("It is a vowel");

else if(ch=='e' || ch=='E')

System.out.println("It is a vowel");

else if(ch=='i' || ch=='I')

System.out.println("It is a vowel");

else if(ch=='o' || ch=='O')

System.out.println("It is a vowel");

else if(ch=='u'|| ch=='U')

System.out.println("It is a vowel");

else

System.out.println("It is not a vowel");

}

}

Q) Write a java program to check given alphabet is a uppercase letter , lowercase letter, digit or a special symbol?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the alphabet :");

char ch=sc.next().charAt(0);

if(ch>='A' && ch<='Z')

System.out.println("It is uppercase letter");

else if(ch>='a' && ch<='z')

System.out.println("It is lowercase letter");

else if(ch>='0' && ch<='9')

System.out.println("It is digit");

else

System.out.println("It is special symbol");

}

}

Assignment

==========

Q) Write a java program to accept six marks of a student then find out total ,average and grade?

i) If average is greater then equals to 70 then A grade.

ii) If average is greater then equals to 50 then B grade.

iii) If average is greater then equals to 35 then C gade.

iv) If average is less then 35 then failed.

iv) nested if stmt

====================

If stmt contains another if stmt is called nested if stmt.

syntax:

------

if(condition)

{

if(Condition)

{

-

- //code to be execute

-

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

if(5>2)

{

System.out.println("stmt2");

if(true)

{

System.out.println("stmt3");

}

System.out.println("stmt4");

}

System.out.println("stmt5");

}

}

o/p:

stmt1

stmt2

stmt3

stmt4

stmt5

ex:

-----

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

if(5>2)

{

System.out.println("stmt2");

if(false)

{

System.out.println("stmt3");

}

System.out.println("stmt4");

}

System.out.println("stmt5");

}

}

o/p:

stmt1

stmt2

stmt4

stmt5

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

if(5>20)

{

System.out.println("stmt2");

if(true)

{

System.out.println("stmt3");

}

System.out.println("stmt4");

}

System.out.println("stmt5");

}

}

Q) Write a java program to check given number is positive or negative using nested if stmt?

import java.util.\*;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

if(n!=0)

{

if(n>0)

{

System.out.println("It is positive number ");

System.exit(0);

}

System.out.println("It is negative number");

}

}

}

Orchasp interview Question

==========================

Write a java program to calculate costs based on user input. The program

should prompt users to enter the total weight of items(in kilograms) and the shipping destination (domestic or international). for domestic orders, the program should charge Rs.500 for weights upto 5 kg and Rs.100 per additional kg. for international orders, it should charge Rs.1000 for weights upto 5 kg , Rs.200 per additional kg , and a Rs.500 surcharge for weights exceeding 10 kg. print calculated shipping cost.

input:

total weight of items : 11

Shipping Destination : domestic

output:

1100

Explanation

------------

weight <=5 = (Rs.500)

weiht <=11 = (Rs.500 + 600)

ex:

import java.util.\*;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter total weight of item :");

int weight=sc.nextInt(); // 11

System.out.println("Enter shipping destination :");

String destination=sc.next(); // domestic

if(destination.equals("domestic"))

{

if(weight<=5)

{

System.out.println(500);

}

if(weight>5)

{

System.out.println(500 + (weight-5)\*100);

}

}

else if(destination.equals("international"))

{

if(weight<=5)

{

System.out.println(1000);

}

else if(weight>5 && weight<=10)

{

System.out.println(1000 + (weight-5) \* 200);

}

else if(weight>10)

{

System.out.println(1000 + (weight-5) \* 200 + 500);

}

}

}

}

Class 19:-

2) Selection statement

=======================

switch case

============

It will execute the source code based on multiple conditions.

It is similar to if else if ladder.

syntax:

-------

switch(condition)

{

case value1: //code to be execute

break stmt;

case value2: //code to be execute

break stmt;

-

-

default: //code to be execute if all cases are false.

}

ex:

----

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the option :");

int option=sc.nextInt();

switch(option)

{

case 100: System.out.println("It is a police number");

break;

case 103: System.out.println("It is a enquiry number");

break;

case 108: System.out.println("It is a emergency number");

break;

default : System.out.println("Invalid option");

}

}

}

Declaration of break stmt in switch case is optional.If we won't declare break statement then from where our condition is satisfied from there all cases will be executed that state is called fall through state of switch case.

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the option :");

int option=sc.nextInt();

switch(option)

{

case 100: System.out.println("It is a police number");

//break;

case 103: System.out.println("It is a enquiry number");

//break;

case 108: System.out.println("It is a emergency number");

//break;

default : System.out.println("Invalid option");

}

}

}

Q) Write a java program to check given alphabet is a vowel or consonent?

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the alphabet :");

char ch=sc.next().charAt(0);

switch(ch)

{

case 'a': System.out.println("It is a vowel"); break;

case 'e': System.out.println("It is a vowel"); break;

case 'i': System.out.println("It is a vowel"); break;

case 'o': System.out.println("It is a vowel"); break;

case 'u': System.out.println("It is a vowel"); break;

default : System.out.println("It is a consonent");

}

}

}

Note:

----

The allowed data type for switch case are byte, short, int, char and String.

3) Iteration statement

======================

Iteration statement is used to execute the code repeatedly.

Iteration statement is possible by using loops.

We have four types of loops.

i) do while loop

ii) while loop

iii) for loop

iv) for each loop

i) do while loop

=================

It will execute the source code how long our condition is true.

syntax:

------

do

{

-

- //code to be execute

-

}while(condition);

ex:

---

class Test

{

public static void main(String[] args)

{

int i=1;

do

{

System.out.print(i+" "); // infinite 1

}

while (i<=10);

}

}

In do while loop, our code will execute atleast for one time either our condition is true or false.

ex:

class Test

{

public static void main(String[] args)

{

int i=11;

do

{

System.out.print(i+" "); // 11

}

while (i<=10);

}

}

Q) Write a java program to display 10 natural numbers?

class Test

{

public static void main(String[] args)

{

int i=1;

do

{

System.out.print(i+" "); // 1 2 3 4 5 6 7 8 9 10

i++;

}

while (i<=10);

}

}

Q) Write a java program to display 10 natural numbers in descending order?

class Test

{

public static void main(String[] args)

{

int i=10;

do

{

System.out.print(i+" "); // 10 9 8 7 6 5 4 3 2 1

i--;

}

while (i>=1);

}

}

Q) Write a java program to perform sum of 10 natural numbers?

class Test

{

public static void main(String[] args)

{

int i=1,sum=0;

do

{

sum = sum + i;

i++;

}

while (i<=10);

System.out.println(sum);

}

}

Q) Write a java program to find out factorial of a given number?

input:

n = 5

output:

120 (5\*4\*3\*2\*1)

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); // 5

int i=n,fact=1;

do

{

fact = fact \* i;

i--;

}

while (i>=1);

System.out.println(fact);

}

}

Q) Write a java program to display multiplication table of a given number?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); // 5

int i=1;

do

{

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

i++;

}

while (i<=10);

}

}

Class 20:-

ii) while loop

==============

It will execute the source code how long our condition is true.

syntax:

-------

while(condition)

{

-

- //code to be execute

-

}

ex:

----

class Test

{

public static void main(String[] args)

{

int i=1;

while(i<=10)

{

System.out.print(i+" "); // infinite 1

}

}

}

ex:

----

class Test

{

public static void main(String[] args)

{

int i=11;

while(i<=10)

{

System.out.print(i+" "); //nothing

}

}

}

Q) Write a java program to display 10 natural numbers?

ex:

---

class Test

{

public static void main(String[] args)

{

int i=1;

while(i<=10)

{

System.out.print(i+" "); //1 2 3 4 5 6 7 8 9 10

i++;

}

}

}

Q) Write a java program to perform sum of 10 natural numbers?

class Test

{

public static void main(String[] args)

{

int i=1,sum=0;

while(i<=10)

{

sum+=i;

i++;

}

System.out.println(sum);

}

}

Q) Write a java program to display factorial of a given number?

input:

n=5

output:

120

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

int i=n,fact=1;

while(i>=1)

{

fact\*=i;

i--;

}

System.out.println(fact);

}

}

Q) Write a java program to display multiplication table of a given number?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

int i=1;

while(i<=10)

{

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

i++;

}

}

}

Q) Write a java program to perform sum of digits of a given number?

input:

123

output:

6

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); //123

int rem,sum=0;

while(n>0)

{

rem=n%10;

sum=sum+rem;

n=n/10;

}

System.out.println(sum);

}

}

Q) Write a java program to find out given number is Armstrong or not?

input:

153

output:

It is armstrong number (1\*1\*1+5\*5\*5+3\*3\*3) (1+125+27)(153)

ex:

----

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); //153

int temp=n;

int rem,sum=0;

while(n>0)

{

rem=n%10;

sum=sum+rem\*rem\*rem;

n=n/10;

}

if(sum==temp)

System.out.println("It is armstrong number");

else

System.out.println("It is not armstrong number");

}

}

Q) Write a java program to display reverse of a given number?

input:

123

output:

321

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); //123

int rem,rev=0;

while(n>0)

{

rem=n%10;

rev=rev\*10+rem;

n=n/10;

}

System.out.println(rev);

}

}

Q) Write a java program to check given number is palindrome or not?

input:

121

output:

It is a palindrome number

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); //123

int temp=n;

int rem,rev=0;

while(n>0)

{

rem=n%10;

rev=rev\*10+rem;

n=n/10;

}

if(temp==rev)

System.out.println("It is palindrome number");

else

System.out.println("It is not palindrome number");

}

}

Q) Write a java program to display reverse of a given number?

input:

123

output:

ThreeTwoOne

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); //123

while(n>0)

{

switch(n%10)

{

case 0: System.out.print("Zero"); break;

case 1: System.out.print("One"); break;

case 2: System.out.print("Two"); break;

case 3: System.out.print("Three"); break;

case 4: System.out.print("Four"); break;

case 5: System.out.print("Five"); break;

case 6: System.out.print("Six"); break;

case 7: System.out.print("Seven"); break;

case 8: System.out.print("Eight"); break;

case 9: System.out.print("Nine"); break;

}

n=n/10;

}

}

}

iii) for loop

=============

It will execute the source code how long our condition is true.

syntax:

-------

for(initialization;condition;incrementation/decrementation)

{

-

- //code to be execute

-

}

**Class 21:-**

If number of iterations are known by the user then we need to use for loop.

If number of iterations are not known by the user then we need to use while loop.

If number of iterations are not known by the user but code must execute alteast for one time then we need to use do while loop.

ex:

---

class Test

{

public static void main(String[] args)

{

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

{

System.out.print(i+" "); //1 2 3 4 5 6 7 8 9 10

}

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

for(;;)

{

System.out.print("Hello "); // infinite Hello

}

}

}

ex:

---

class Test

{

public static void main(String[] args)

{

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

{

System.out.print("Hello ");

i--;

}

}

}

Q) Write a java program to check given number is prime or not?

input:

5

output:

It is a prime number

ex:

----

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); // 5

boolean flag=true;

for(int i=2;i<=n/2;i++)

{

if(n%i==0)

{

flag=false;

break;

}

}

if(flag==true)

System.out.println("It is a prime number ");

else

System.out.println("It is not a prime number");

}

}

Q) Write a java program to display prime numbers from 1 to 100?

class Test

{

public static void main(String[] args)

{

for(int n=2;n<=100;n++)

{

boolean flag=true;

for(int i=2;i<=n/2;i++)

{

if(n%i==0)

{

flag=false;

break;

}

}

if(flag==true)

System.out.print(n+" ");

}

}

}

Q) Write a java program to check given number is perfect or not?

input:

6

output:

It is a perfect number

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); //6

int sum=0;

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

{

if(n%i==0)

{

sum+=i;

}

}

if(n==sum)

System.out.println("It is perfect number ");

else

System.out.println("It is not perfect number");

}

}

Q) Write java program to display GCD (Greatest Common Divisor) of two numbers?

input:

12 18

output:

6

ex:

class Test

{

public static void main(String[] args)

{

int a=12,b=18,gcd=0;

for(int i=1;i<=a && i<=b;i++)

{

if(a%i==0 && b%i==0)

{

gcd=i;

}

}

System.out.println("GCD of two numbers is ="+gcd);

}

}

Q) Write a java program to find out fibonacci series of a given number?

fibonacci series : 0 1 1 2 3 5 8

ex:

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); // 6

int a=0,b=1,c;

System.out.print(a+" "+b+" ");

for(int i=2;i<=n;i++)

{

c = a+b;

System.out.print(c+" ");

a=b;

b=c;

}

}

}

Assignment program

-------------------

Q) Write a java program to check given number is even or odd without using modules operator?

ex:

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); // 6

if((n&1) == 0)

System.out.println("It is even number");

else

System.out.println("It is odd number");

}

}

**Class 22:-**

Q) What will be the output of below program?

ex:

class Test

{

public static void main(String[] args)

{

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

{

if(i%2==0)

{

System.out.print(i+" "); //2 4 6 8 10

}

}

}

}

Q) What will be the output of below program?

ex:

class Test

{

public static void main(String[] args)

{

int cnt=0;

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

{

if(i%2==0)

{

cnt++;

}

}

System.out.println(cnt); // 5

}

}

Loop Pattern

===================

1)

1 1 1 1

2 2 2 2

3 3 3 3

4 4 4 4

class Test

{

public static void main(String[] args)

{

for(int i=1;i<=4;i++)

{

for(int j=1;j<=4;j++)

{

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

}

//new line

System.out.println();

}

}

}

2)

1 2 3 4

1 2 3 4

1 2 3 4

1 2 3 4

class Test

{

public static void main(String[] args)

{

for(int i=1;i<=4;i++)

{

for(int j=1;j<=4;j++)

{

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

}

//new line

System.out.println();

}

}

}

3)

\* \* \* \*

\* \* \* \*

\* \* \* \*

\* \* \* \*

ex:

class Test

{

public static void main(String[] args)

{

for(int i=1;i<=4;i++)

{

for(int j=1;j<=4;j++)

{

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

}

//new line

System.out.println();

}

}

}

4)

4 4 4 4

3 3 3 3

2 2 2 2

1 1 1 1

ex:

class Test

{

public static void main(String[] args)

{

for(int i=4;i>=1;i--)

{

for(int j=1;j<=4;j++)

{

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

}

//new line

System.out.println();

}

}

}

5)

\* \* \* \*

\* \*

\* \*

\* \* \* \*

ex:

class Test

{

public static void main(String[] args)

{

for(int i=1;i<=4;i++)

{

for(int j=1;j<=4;j++)

{

if(i==1 || i==4 || j==1 || j==4)

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

else

System.out.print(" ");

}

//new line

System.out.println();

}

}

}

6)

\* - - -

- \* - -

- - \* -

- - - \*

class Test

{

public static void main(String[] args)

{

for(int i=1;i<=4;i++)

{

for(int j=1;j<=4;j++)

{

if(i==j)

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

else

System.out.print("- ");

}

//new line

System.out.println();

}

}

}

7)

\* - - - \*

- \* - \* -

- - \* - -

- \* - \* -

\* - - - \*

class Test

{

public static void main(String[] args)

{

for(int i=1;i<=5;i++)

{

for(int j=1;j<=5;j++)

{

if(i==j || i+j==6)

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

else

System.out.print("- ");

}

//new line

System.out.println();

}

}

}

Left Side Loop Patterns

=======================

1)

1

2 2

3 3 3

4 4 4 4

class Test

{

public static void main(String[] args)

{

//rows

for(int i=1;i<=4;i++)

{

//cols

for(int j=1;j<=i;j++)

{

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

}

//new line

System.out.println();

}

}

}

2)

4 4 4 4

3 3 3

2 2

1

class Test

{

public static void main(String[] args)

{

//rows

for(int i=4;i>=1;i--)

{

//cols

for(int j=1;j<=i;j++)

{

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

}

//new line

System.out.println();

}

}

}

3)

\*

\* \*

\* \* \*

\* \* \* \*

class Test

{

public static void main(String[] args)

{

//rows

for(int i=1;i<=4;i++)

{

//cols

for(int j=1;j<=i;j++)

{

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

}

//new line

System.out.println();

}

}

}

4)

1

2 3

4 5 6

7 8 9 0

ex:

class Test

{

public static void main(String[] args)

{

int k=1;

//rows

for(int i=1;i<=4;i++)

{

//cols

for(int j=1;j<=i;j++)

{

if(k<=9)

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

else

System.out.print("0 ");

}

//new line

System.out.println();

}

}

}

5)

2

4 6

8 10 12

14 16 18 20

class Test

{

public static void main(String[] args)

{

int k=2;

//rows

for(int i=1;i<=4;i++)

{

//cols

for(int j=1;j<=i;j++)

{

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

k+=2;

}

//new line

System.out.println();

}

}

}

6)

1

3 5

7 9 11

13 15 17 19

ex:

---

class Test

{

public static void main(String[] args)

{

int k=1;

//rows

for(int i=1;i<=4;i++)

{

//cols

for(int j=1;j<=i;j++)

{

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

k+=2;

}

//new line

System.out.println();

}

}

}

7)

2

3 5

7 11 13

17 19 23 29

ex:

---

class Test

{

public static void main(String[] args)

{

int n=2;

//rows

for(int i=1;i<=4;i++)

{

//cols

for(int j=1;j<=i;j++)

{

while(true)

{

boolean flag=true;

for(int k=2;k<=n/2;k++)

{

if(n%k==0)

{

flag=false;

break;

}

}

if(flag==true)

{

System.out.print(n+" ");

n++;

break;

}

else

{

n++;

}

}

}

//new line

System.out.println();

}

}

}

8)

1

2 1

1 2 3

4 3 2 1

ex:

class Test

{

public static void main(String[] args)

{

//rows

for(int i=1;i<=4;i++)

{

if(i%2!=0)

{

for(int j=1;j<=i;j++)

{

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

}

//new line

System.out.println();

}

else

{

for(int j=i;j>=1;j--)

{

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

}

//new line

System.out.println();

}

}

}

}

**Class 23:-**

Right side loop patterns

========================

1)

1

2 2

3 3 3

4 4 4 4

ex:

---

class Test

{

public static void main(String[] args)

{

//rows

for(int i=1;i<=4;i++)

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

//elements

for(int j=1;j<=i;j++)

{

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

}

//new line

System.out.println();

}

}

}

2)

4 4 4 4

3 3 3

2 2

1

class Test

{

public static void main(String[] args)

{

//rows

for(int i=4;i>=1;i--)

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

//elements

for(int j=1;j<=i;j++)

{

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

}

//new line

System.out.println();

}

}

}

3)

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

class Test

{

public static void main(String[] args)

{

//rows

for(int i=1;i<=4;i++)

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

//elements

for(int j=1;j<=i;j++)

{

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

}

//new line

System.out.println();

}

//rows

for(int i=3;i>=1;i--)

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

//elements

for(int j=1;j<=i;j++)

{

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

}

//new line

System.out.println();

}

}

}

Pyramid loop patterns

===================

1)

1

1 2 1

1 2 3 2 1

1 2 3 4 3 2 1

ex:

---

class Test

{

public static void main(String[] args)

{

//rows

for(int i=1;i<=4;i++)

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

//left side elements

for(int j=1;j<=i;j++)

{

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

}

//right side elements

for(int j=i-1;j>=1;j--)

{

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

}

//new line

System.out.println();

}

}

}

2)

1 2 3 4 3 2 1

1 2 3 2 1

1 2 1

1

ex:

class Test

{

public static void main(String[] args)

{

//rows

for(int i=4;i>=1;i--)

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

//left side elements

for(int j=1;j<=i;j++)

{

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

}

//right side elements

for(int j=i-1;j>=1;j--)

{

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

}

//new line

System.out.println();

}

}

}

3)

\*

\* \* \*

\* \* \* \* \*

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

\* \* \* \* \*

\* \* \*

\*

ex:

---

class Test

{

public static void main(String[] args)

{

//rows

for(int i=1;i<=4;i++)

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

//left side elements

for(int j=1;j<=i;j++)

{

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

}

//right side elements

for(int j=i-1;j>=1;j--)

{

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

}

//new line

System.out.println();

}

//rows

for(int i=3;i>=1;i--)

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

//left side elements

for(int j=1;j<=i;j++)

{

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

}

//right side elements

for(int j=i-1;j>=1;j--)

{

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

}

//new line

System.out.println();

}

}

}

4) Write a java program to display pascal triangle?

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

ex:

----

class Test

{

public static void main(String[] args)

{

//rows

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

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

int number=1;

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

{

System.out.print(number+" ");

number = (number \* (i-k))/(k+1);

}

//new line

System.out.println();

}

}

}

Q) Write a java program to display the given loop pattern?

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

ex:

class Test

{

public static void main(String[] args)

{

//rows

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

{

//space

for(int j=4;j>i;j--)

{

System.out.print(" ");

}

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

{

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

}

//new line

System.out.println();

}

}

}

Q) Write a java program to display below loop pattern?

1 1

1 2 2 1

1 2 3 3 2 1

1 2 3 4 4 3 2 1

ex:

class Test

{

public static void main(String[] args)

{

int rows=4;

//rows

for(int i=1;i<=rows;i++)

{

//left side elements

for(int j=1;j<=i;j++)

{

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

}

//space

for(int j=1;j<=(rows-i)\*2;j++)

{

System.out.print(" ");

}

//right side elements

for(int j=i;j>=1;j--)

{

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

}

//new line

System.out.println();

}

}

}

Assignment

===========

Q) write a java program to display below loop pattern?

\*

\*

\* \* \* \* \*

\*

\*

class Test

{

public static void main(String[] args)

{

for(int i=1;i<=5;i++)

{

for(int j=1;j<=5;j++)

{

if(i==3 || j==3)

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

else

System.out.print(" ");

}

//new line

System.out.println();

}

}

}

**Class 24:-**

4) Jump Statement

==================

Jump statement is used to jump from one section of code to another section.

We have two types of jump statements.

i) break statement

ii) continue statement

i) break statement

-------------------

It is used to break the execution of loops and switch case.

For conditional statement we can use if condition.

ex:

----

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

break;

System.out.println("stmt2");

}

}

o/p:

C.T.E : break outside switch or loop

ex:

----

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

if(true)

{

break;

}

System.out.println("stmt2");

}

}

o/p:

C.T.E : break outside switch or loop

ex:

---

class Test

{

public static void main(String[] args)

{

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

{

if(i==5)

{

break;

}

System.out.print(i+" "); //1 2 3 4

}

}

}

ii) continue statement

-------------------

It is used to continue the execution of loops.

For conditional statements we can use if condition.

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

continue;

System.out.println("stmt2");

}

}

o/p:

C.T.E : continue outside of loop

ex:

----

class Test

{

public static void main(String[] args)

{

System.out.println("stmt1");

if(true)

{

continue;

}

System.out.println("stmt2");

}

}

o/p:

C.T.E : continue outside of loop

ex:

---

class Test

{

public static void main(String[] args)

{

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

{

if(i==5)

{

continue;

}

System.out.print(i+" "); //1 2 3 4 6 7 8 9 10

}

}

}

Various ways to declare the methods in java

============================================

1) No return type with No argument method

2) No return type with Argument method

3) With return type with No argument method

4) With return type with Argument method

1) No return type with No argument method

------------------------------------------

If we don't have arguments then we need to ask input values inside callie method.

Q) Write a java program to perform sum of two numbers with no returntype with no argument method?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

//caller method

sum();

sum();

}

//callie method

public static void sum()

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();

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

int b=sc.nextInt();

//logic

int c=a+b;

System.out.println("sum of two numbers is ="+c);

}

}

Q) Write a java program to display sum of 10 natural numbers using no returntype with no argument method?

class Test

{

public static void main(String[] args)

{

//caller method

sum();

}

//callie method

public static void sum()

{

int sum=0;

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

{

sum+=i;

}

System.out.println("sum of 10 natural numbers is ="+sum);

}

}

2) No return type with Argument method

-------------------------------------

If we have arguments then we need to ask input values inside main method.

Here number of arguments depends upon number of inputs.

Q) Write a java program to perform sum of two numbers using no returntype with argument method?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();//10

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

int b=sc.nextInt();//20

//caller method

sum(a,b);

}

//callie method

public static void sum(int a,int b)

{

int c=a+b;

System.out.println("sum of two numbers is ="+c);

}

}

Q) Write a java program to perform cube of a given number using no returntype with argument method?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

//caller method

cube(n);

}

//callie method

public static void cube(int n)

{

//int result=(int)Math.pow(n,3);

int result = n\*n\*n;

System.out.println("cube of a given number is ="+result);

}

}

Assignment

==========

Q) Write a java program to check given number is prime or not?

**Class 25:-**

3) With returntype with No argument method

===========================================

Returntype is completely depends upon output datatype.

Q) Write a java program to perform sum of two numbers using with returntype with no argument method?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

//caller method

int k=sum();

System.out.println("sum of two numbers is ="+k);

}

//callie method

public static int sum()

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();

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

int b=sc.nextInt();

//logic

int c = a+b;

return c;

}

}

Q) Write a java program to check given number is palindrome or not by using with return type with no argument method?

Input:

121

output:

It is a palindrome number

approach1

---------

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

//caller method

String k=palindrome();

System.out.println(k);

}

//callie method

public static String palindrome()

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

int temp=n;

int rem,rev=0;

while(n>0)

{

rem=n%10;

rev=rev\*10+rem;

n=n/10;

}

if(temp==rev)

return "It is a palindrome number";

else

return "It is not a palindrome number";

}

}

approach2

----------

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

//caller method

boolean k=palindrome();

if(k)

System.out.println("It is a palindrome string");

else

System.out.println("It is not a palindrome string");

}

//callie method

public static boolean palindrome()

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

int temp=n;

int rem,rev=0;

while(n>0)

{

rem=n%10;

rev=rev\*10+rem;

n=n/10;

}

if(temp==rev)

return true;

else

return false;

}

}

approach3

---------

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

//caller method

int k=palindrome();

if(k==1)

System.out.println("It is a palindrome string");

else

System.out.println("It is not a palindrome string");

}

//callie method

public static int palindrome()

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

int temp=n;

int rem,rev=0;

while(n>0)

{

rem=n%10;

rev=rev\*10+rem;

n=n/10;

}

if(temp==rev)

return 1;

else

return 0;

}

}

4) With returntype with Argument method

---------------------------------------

Q) Write a java program to perform sum of two numbers using with returntype with argument method?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();

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

int b=sc.nextInt();

//caller method

System.out.println("sum of two number is ="+sum(a,b));

}

//callie method

public static int sum(int a,int b)

{

int c = a + b;

return c;

}

}

Q) write a java program to check given number is palindrome or not using with returntype with argument method?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();

//caller method

if(palindrome(n))

System.out.println("It is palindrome number");

else

System.out.println("It is not palindrome number");

}

//callie method

public static boolean palindrome(int n)

{

int temp=n;

int rem,rev=0;

while(n>0)

{

rem=n%10;

rev=rev\*10+rem;

n=n/10;

}

if(temp==rev)

return true;

else

return false;

}

}

Assignment

===========

Q) Write a java program to check given number is prime or not using 3 and 4th approach?

Recursion

==============

A method which call itself for many number of times is called recursion.

Recursion is similar to loopings.

Whenever we use recursion we should not use loops.

Q) Write a java program to display 10 natural numbers without using loops?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

//caller method

display(1);

}

//callie method

public static void display(int i)

{

if(i<=10)

{

System.out.print(i+" "); //1 2 3 4 5 6 7 8 9 10

display(i+1);

}

}

}

Q) Write a java program to perform sum of two numbers without using arthimetic operator ?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int a=sc.nextInt();//5

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

int b=sc.nextInt();//10

//caller method

System.out.println(sum(a,b));

}

//callie method

public static int sum(int a,int b)

{

if(a==0)

return b;

return sum(--a,++b);

}

}

Q) Write a java program to display factorial of a given number using recursion?

input:

5

output:

120

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt();//5

//caller method

System.out.println(factorial(n));

}

//callie method

public static int factorial(int n)

{

if(n<0)

return -1;

if(n==0)

return 1;

return n\*factorial(n-1);

}

}

**Class 26:-**

Q) Write a java program to find out N-th element of fibonacci series?

fibonacci sequence : 0 1 1 2 3 5 8

Input:

4

output:

2

ex:

---

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

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

int n=sc.nextInt(); // 4

//caller method

System.out.println(fib(n));

}

//callie method

public static int fib(int n)

{

if(n==0 || n==1)

return 0;

if(n==2)

return 1;

return fib(n-1)+fib(n-2);

}

}

Arrays

=======

Array is a collection of homogeneous data elements.

The main advantages of arrays are

1) We can represent multiple elements using single variable name.

ex:

int[] arr={10,20,30};

2) Performance point of view arrays are recommanded to use.

The main disadvantages of arrays are

1) It is fixed in the size.Once if we create an array there is no chance of increase or decreasing

the size of an array.

2) To use array concept in advanced we should know what is the size of array which is always not

possible.

In java, arrays are divided into three types .

1) Single Dimensional Array

2) Double Dimensional Array

3) Multi Dimensional Array

Array Declaration

-----------------

At the time of array declaration we should not specify array size.

Arrays

|----------------------------------|-------------------------------------|

Single Dimensional Array Double Dimensional Array Multi Dimensional Array

int[] arr; int[][] arr; int[][][] arr;

int []arr; int [][]arr; int [][][]arr;

int arr[]; int ar[][]; int arr[][][];

int[] []arr; int[][] []arr;

int[] arr[]; int[][] arr[];

int []arr[]; int[] [][]ar;

int[] arr[][];

int[] []arr[];

int [][]arr[];

int []arr[][];

Array creation

--------------

In java, every array consider as an object.Hence we will use new operator to create an array.

Diagram: class26.1

Rules to construct an array:

----------------------------

Rule1:

-----

At the time of array creation compulsary we need to specify array size.

ex:

int[] arr=new int[3];

int[] arr=new int[]; // C.T.E array dimension missing

Rule2:

-----

It is legal to have an array size with zero.

ex:

int[] arr=new int[0];

System.out.println(arr.length);

Rule3:

-----

We can't take negative numbers as an array size otherwise we will get

NegativeArraySizeException.

ex:

int[] arr=new int[-3]; //R.E NegativeArraySizeException

Rule4:

-----

The allowed datatype for an array size is byte,short,int and char.

If we take other datatyps then we will get compile time error.

ex:

int[] arr=new int['a']; //97

byte b=10;

int[] arr=new int[b]; // 10

int[] arr=new int[10.5]; //invalid

Rule5:

-----

The maximum length we can give for an array size is maximum length of integer.

ex:

int[] arr=new int[2147483647];

Array initialization

-------------------

Whenever we create an array , every array element will be initialized with default vlaues.

IF we are not happing with default values then we can change with customized values.

Diagram: class26.2

Array Declaration , Creation and Initialization using single line

------------------------------------------------------------------

int[] arr;

arr=new int[3];

arr[0]=10;

arr[1]=20;

arr[2]=30; ===> int[] arr={10,20,30};

===> char[] carr={'a','b','c'};

===> String[] sarr={"hi","hello","bye"};

Q) What is the difference between length and length() method?

length

------

It is a final variable which is applicable for arrays.

It will return size of an array.

ex:

class Test

{

public static void main(String[] args)

{

int[] arr=new int[3];

System.out.println(arr.length); //3

}

}

length()

--------

It is a final method which is applicable for String objects.

It will return number of characters present in String.

ex:

class Test

{

public static void main(String[] args)

{

String str="ihub";

System.out.println(str.length()); //4

}

}

Single Dimensional Array Programs

----------------------------------

Q) Write a java program to insert some elements in array and display them?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the array size :");

int size=sc.nextInt(); //4

int[] arr=new int[size];

//insert elements

for(int i=0;i<arr.length;i++)

{

System.out.println("Enter the element :");

arr[i]=sc.nextInt();

}

//display elements

for(int i=0;i<arr.length;i++)

{

System.out.print(arr[i]+" ");

}

}

}

iv) for each loop

==================

It is used to iterate the elements from array.

ex:

--

class Test

{

public static void main(String[] args)

{

int[] arr={10,20,30};

//for each loop

for(int i:arr)

{

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

}

}

}

**Class 27:-**

Q) Write a java program to display sum of array elements?

input:

5 8 1 3 9 2

output:

28

ex:

class Test

{

public static void main(String[] args)

{

int[] arr={5,8,1,3,9,2};

int sum=0;

//for each loop

for(int i:arr)

{

sum +=i;

}

System.out.println(sum);

}

}

Q) Write a java program to display even element from given positive integer array?

input:

5 8 1 3 9 2

output:

8 2

ex:

class Test

{

public static void main(String[] args)

{

int[] arr={5,8,1,3,9,2};

//for each loop

for(int i:arr)

{

if(i%2==0)

{

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

}

}

}

}

Q) Write a java program to display number of odd elements present in array?

input:

5 8 1 3 9 2

output:

4

ex:

class Test

{

public static void main(String[] args)

{

int[] arr={5,8,1,3,9,2};

int cnt=0;

//for each loop

for(int i:arr)

{

if(i%2!=0)

{

cnt++;

}

}

System.out.println(cnt);

}

}

Q) Write a java program to display array elements in reverse order?

input:

5 8 1 3 9 2

output:

2 9 3 1 8 5

ex:

class Test

{

public static void main(String[] args)

{

int[] arr={5,8,1,3,9,2};

//reverse logic

for(int i=arr.length-1;i>=0;i--)

{

System.out.print(arr[i]+" ");

}

}

}

Q) Write a java program to display prime elements from given array?

input:

6 2 9 7 4 3 10 5

output:

2 7 3 5

ex:

class Test

{

public static void main(String[] args)

{

int[] arr={6,2,9,7,4,3,10,5};

//for each loop

for(int n:arr)

{

boolean flag=true;

for(int i=2;i<=n/2;i++)

{

if(n%i==0)

{

flag=false;

break;

}

}

if(flag==true)

System.out.print(n+" ");

}

}

}

Q) Write a java program to display array elements in sorting order?

input:

6 2 9 7 4 3

output:

2 3 4 6 7 9

ex:

import java.util.Arrays;

class Test

{

public static void main(String[] args)

{

int[] arr={6,2,9,7,4,3};

Arrays.sort(arr); // 2 3 4 6 7 9

//for each loop

for(int i:arr)

{

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

}

}

}

Q) Write a java program to display array elements in sorting order without using sort() method?

input:

6 2 9 7 4 3

output:

2 3 4 6 7 9

ex:

---

class Test

{

public static void main(String[] args)

{

int[] arr={6,2,9,7,4,3};

//ascending logic

for(int i=0;i<arr.length;i++)

{

for(int j=0;j<arr.length;j++)

{

if(arr[i]<arr[j])

{

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

//for each loop

for(int i:arr)

{

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

}

}

}

Q) Write a java program to display array elements in descending order without using sort() method?

input:

6 2 9 7 4 3

output:

9 7 6 4 3 2

class Test

{

public static void main(String[] args)

{

int[] arr={6,2,9,7,4,3};

//ascending logic

for(int i=0;i<arr.length;i++)

{

for(int j=0;j<arr.length;j++)

{

if(arr[i]>arr[j])

{

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

//for each loop

for(int i:arr)

{

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

}

}

}

Q) Write a java program to display least element from given array?

input:

6 2 9 7 4 3

output:

2

ex:

import java.util.Arrays;

class Test

{

public static void main(String[] args)

{

int[] arr={6,2,9,7,4,3};

Arrays.sort(arr);

System.out.println(arr[0]);

}

}

Q) Write a java program to display highest element from given array?

input:

6 2 9 7 4 3

output:

9

ex:

import java.util.Arrays;

class Test

{

public static void main(String[] args)

{

int[] arr={6,2,9,7,4,3};

Arrays.sort(arr);

System.out.println(arr[arr.length-1]);

}

}

Q) Write a java program to display least element from given array without using sort() method?

input:

6 2 9 7 4 3

output:

2

class Test

{

public static void main(String[] args)

{

int[] arr={6,2,9,7,4,3};

int small=arr[0];

//for each loop

for(int i:arr)

{

if(i<small)

{

small=i;

}

}

System.out.println(small);

}

}

Q) Write a java program to display highest element from given array without using sort() method?

input:

6 2 9 7 4 3

output:

9

class Test

{

public static void main(String[] args)

{

int[] arr={6,2,9,7,4,3};

int big=arr[0];

//for each loop

for(int i:arr)

{

if(i>big)

{

big=i;

}

}

System.out.println(big);

}

}

Assignment

==========

Q) Write a java program to check given number is palindrome or not using recursion?

Q) Write a java program to check given number is reverse or not using recursion?

**Class 28:-**

Q) Write a java program to display three highest elements from given array?

input:

6 9 2 4 7 5 8 1

output:

9 8 7

ex:

class Test

{

public static void main(String[] args)

{

int[] arr={6,9,2,4,7,5,8,1};

int firstElement=Integer.MIN\_VALUE;

int secondElement=Integer.MIN\_VALUE;

int thirdElement=Integer.MIN\_VALUE;

//for each loop

for(int i:arr)

{

if(i>firstElement)

{

thirdElement=secondElement;

secondElement=firstElement;

firstElement=i;

}

else if(i>secondElement)

{

thirdElement=secondElement;

secondElement=i;

}

else if(i>thirdElement)

{

thirdElement=i;

}

}

System.out.println(firstElement+" "+secondElement+" "+thirdElement);

}

}

Q) Write a java program to display duplicate elements from given array?

input:

1 6 4 9 2 7 2 9 5 4

output:

4 9 2

ex:

class Test

{

public static void main(String[] args)

{

int[] arr={1,6,4,9,2,7,2,9,5,4};

//duplicate elements

for(int i=0;i<arr.length;i++)

{

for(int j=i+1;j<arr.length;j++)

{

if(arr[i]==arr[j])

{

System.out.print(arr[i]+" ");

}

}

}

}

}

Q) Write a java program to display unique elements from given array?

input:

1 6 4 9 2 7 2 9 5 4

output:

1 6 7 5

ex:

class Test

{

public static void main(String[] args)

{

int[] arr={1,6,4,9,2,7,2,9,5,4};

//unique elements

for(int i=0;i<arr.length;i++)

{

int cnt=0;

for(int j=0;j<arr.length;j++)

{

if(arr[i]==arr[j])

{

cnt++;

}

}

if(cnt==1)

System.out.print(arr[i]+" ");

}

}

}

Q) Write a java program to display most repeating element from given array?

input:

1 2 6 4 9 2 7 2 9 5 4 5 2 9 4

output:

2 repeating for 4 times

class Test

{

public static void main(String[] args)

{

int[] arr={1,2,6,4,9,2,7,2,9,5,4,5,2,9,4};

int element=0;

int maxCount=0;

for(int i=0;i<arr.length;i++)

{

int cnt=0;

for(int j=0;j<arr.length;j++)

{

if(arr[i]==arr[j])

{

cnt++;

}

}

if(maxCount<cnt)

{

maxCount=cnt;

element=arr[i];

}

}

System.out.println(element+" repeating for "+maxCount+" times");

}

}

Q)Write a java program to find out leader elements from given array?

input:

3 5 34 8 12 1 7

output:

7 12 34

ex:

---

class Test

{

public static void main(String[] args)

{

int[] arr={3,5,34,8,12,1,7};

int max=arr[arr.length-1];

System.out.print(max+" ");

//reading reverse

for(int i=arr.length-2;i>=0;i--)

{

if(arr[i]>max)

{

max=arr[i];

System.out.print(max+" ");

}

}

}

}

Assignments

============

Q) Write a java program to display below loop pattern?

4 4 4 4 4 4 4

4 3 3 3 3 3 4

4 3 2 2 2 3 4

4 3 2 1 2 3 4

4 3 2 2 2 3 4

4 3 3 3 3 3 4

4 4 4 4 4 4 4

class Test

{

public static void main(String[] args)

{

int n=4;

int size=7;

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

{

for(int j=0;j<size;j++)

{

int value = n - Math.min(Math.min(i,j),Math.min(size-i-1,size-j-1));

System.out.print(value+" ");

}

//new line

System.out.println();

}

}

}

Q) Write a java program to display reverse of a given number using recursion?

class Test

{

public static void main(String[] args)

{

int n=123;

System.out.println(reverse(n));

}

public static int reverse(int n)

{

return reverseNumber(n,0);

}

public static int reverseNumber(int n,int reverse)

{

if(n==0)

{

return reverse;

}

int rem=n%10;

reverse = (reverse \* 10) + rem;

return reverseNumber(n/10,reverse);

}

}

**Class 29:-**

Q) Write a java program to find out missing element from given array?

input:

7 1 6 2 3 5

output:

4

ex:

---

class Test

{

public static void main(String[] args)

{

int[] arr={7,1,6,2,3,5};

int sum\_of\_ele=arr.length+1;

int sum=(sum\_of\_ele\*(sum\_of\_ele + 1))/2;

for(int i:arr)

{

sum=sum-i;

}

System.out.println(sum);

}

}

Q) Write a java program to segregate array elements?

input:

1 0 1 0 0 1 1 0 1 0

output:

0 0 0 0 0 1 1 1 1 1

ex:

---

class Test

{

public static void main(String[] args)

{

int[] arr={1,0,1,0,0,1,1,0,1,0};

int[] newArr=new int[arr.length];

//for each loop

int j=0;

for(int i:arr)

{

if(i==0)

{

newArr[j++] = i;

}

}

//inserting 1

while(j<arr.length)

{

newArr[j++]=1;

}

//display the elments

for(int i:newArr)

{

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

}

}

}

Q) Write a java program to merge two arrays and display them in a sorting order?

input:

6 9 7 8 10

5 1 3 4 2

output:

1 2 3 4 5 6 7 8 9 10

ex:

--

import java.util.Arrays;

class Test

{

public static void main(String[] args)

{

int[] arr1={6,9,7,8,10};

int[] arr2={5,1,3,4,2};

int size1=arr1.length;

int size2=arr2.length;

arr1=Arrays.copyOf(arr1,size1+size2);

int j=0;

for(int i=size1;i<arr1.length;i++)

{

arr1[i]=arr2[j++];

}

//sorting

Arrays.sort(arr1);

//display the array elements

for(int i:arr1)

{

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

}

}

}

Q) Write a java program to remove first occurance of a given element from given array?

input:

arr = 6 4 2 9 1 2 3 7 2

element = 2

output:

6 4 9 1 2 3 7 2

ex:

----

import java.util.Arrays;

class Test

{

public static void main(String[] args)

{

int[] arr={6,4,2,9,1,2,3,7,2};

int[] newArr=new int[arr.length-1];

int element=2;

int cnt=0,j=0;

for(int i:arr)

{

if(i==element && cnt==0)

{

cnt=1;

continue;

}

newArr[j++]=i;

}

//display the elements

for(int i:newArr)

{

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

}

}

}

Q) Write a java program to insert given element on a given index of an array?

input:

arr = 7 3 9 1 4 2

element = 100

index = 2

output:

7 3 100 9 1 4 2

ex:

---

import java.util.Arrays;

class Test

{

public static void main(String[] args)

{

int[] arr ={7,3,9,1,4,2};

int element = 100;

int index = 2;

arr = Arrays.copyOf(arr,arr.length+1);

//reading reverse

for(int i=arr.length-1;i>=index;i--)

{

arr[i]=arr[i-1];

}

arr[index]=element;

for(int i:arr)

{

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

}

}

}

Q) Write a java program to display pair of elements equals to sum ?

input:

0 1 2 3 5 6 7 11 12 14 20

output:

0 + 5 = 5

2 + 3 = 5

ex:

---

class Test

{

public static void main(String[] args)

{

int[] arr={0,1,2,3,5,6,7,11,12,14,20};

for(int i=0;i<arr.length;i++)

{

for(int j=i+1;j<arr.length;j++)

{

if(arr[i] + arr[j] == 5)

{

System.out.println(arr[i]+" + "+arr[j]);

}

}

}

}

}

Q) Write a java program to display triplet of arrays elements equals to sum ?

input:

arr = 0 1 2 3 5 6 7 11 12 14 20

sum = 10

output:

0 3 7

1 2 7

1 3 6

2 3 5

ex:

---

class Test

{

public static void main(String[] args)

{

int[] arr={0,1,2,3,5,6,7,11,12,14,20};

int sum=10;

for(int i=0;i<arr.length;i++)

{

for(int j=i+1;j<arr.length;j++)

{

for(int k=j+1;k<arr.length;k++)

{

if(arr[i]+arr[j]+arr[k]==sum)

{

System.out.println(arr[i]+" "+arr[j]+" "+arr[k]);

}

}

}

}

}

}

**Class 30:-**

Q) Write a java program to identify and print all elements in an array that are greater

than both their immediate predecessors and successors, considering the first and

last elements as having only one neighbor?

Input:

1 3 20 4 75 0 90

Output:

20 75 90

ex:

---

class Test

{

public static void main(String[] args)

{

int[] arr={1,3,20,4,75,0,90};

//first element

if(arr[0]>arr[1])

{

System.out.print(arr[0]+" ");

}

//middle elements

for(int i=1;i<=arr.length-2;i++)

{

if(arr[i]>arr[i-1] && arr[i]>arr[i+1])

{

System.out.print(arr[i]+" ");

}

}

//last element

if(arr[arr.length-1]>arr[arr.length-2])

{

System.out.println(arr[arr.length-1]+" ");

}

}

}

Q) Write a java program to determine the smallest number of coins needed to total 86 rupees.

Use the denominations provided in the array {1,2,5,10}?

Output:

1 coin(s) of 1 rupee(s)

1 coin(s) of 5 rupee(s)

8 coin(s) of 10 rupee(s)

ex:

---

class Test

{

public static void main(String[] args)

{

int[] denominations={1,2,5,10};

int amount=86;

//caller method

int[] result=minimumCoins(denominations,amount);

for(int i=0;i<result.length;i++)

{

if(result[i]>0)

{

System.out.println(result[i]+" coin(s) of "+denominations[i]+" rupee(s)");

}

}

}

//callie method

public static int[] minimumCoins(int[] denominations,int amount)

{

int[] coinsCount=new int[denominations.length];

//reading reverse

for(int i=denominations.length-1;i>=0;i--)

{

coinsCount[i]=amount/denominations[i];

amount=amount%denominations[i];

}

return coinsCount;

}

}

Double Dimensional Array

========================

Double dimensional array is a combination of rows and columns.

Double dimensional array is implemented based on array or arrays approach but not matrix form.

The main objective of double dimensional array is memory utilization.

Double dimensional array is used to develop business oriented application, gaming applications, matrix type of applications and etc.

We can declare dimensional array as follow.

ex:

columns

|

int[][] arr=new int[3][3];

|

rows

Here we can store 9 elements.

Q) Write a java program to display array elements in matrix form?

import java.util.Scanner;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the rows :");

int rows=sc.nextInt(); //3

System.out.println("Enter the columns :");

int cols=sc.nextInt(); //3

int[][] arr=new int[rows][cols];

//insert the elements

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

{

for(int j=0;j<cols;j++)

{

System.out.println("Enter the element :");

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

}

}

//display the elements

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

{

for(int j=0;j<cols;j++)

{

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

}

//new line

System.out.println();

}

}

}

Q) Write a java program to display square of a matrix?

input:

1 2 3

4 5 6

7 8 9

ex:

class Test

{

public static void main(String[] args)

{

int[][] arr={

{1,2,3},

{4,5,6},

{7,8,9}

};

int rows=arr.length;

int cols=arr[0].length;

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

{

for(int j=0;j<cols;j++)

{

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

}

//new line

System.out.println();

}

}

}

Q) Write a java program to perform sum of diagonal elements?

class Test

{

public static void main(String[] args)

{

int[][] arr={

{1,2,3},

{4,5,6},

{7,8,9}

};

int rows=arr.length;

int cols=arr[0].length;

int sum=0;

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

{

for(int j=0;j<cols;j++)

{

if(i==j)

{

sum += arr[i][j];

}

}

}

System.out.println(sum);

}

}

Q) Write a java program to perform sum of upper triangle elements?

class Test

{

public static void main(String[] args)

{

int[][] arr={

{1,2,3},

{4,5,6},

{7,8,9}

};

int rows=arr.length;

int cols=arr[0].length;

int sum=0;

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

{

for(int j=0;j<cols;j++)

{

if(i<j)

{

sum += arr[i][j];

}

}

}

System.out.println(sum);

}

}

Q) Write a java program to perform sum of lower triangle elements?

class Test

{

public static void main(String[] args)

{

int[][] arr={

{1,2,3},

{4,5,6},

{7,8,9}

};

int rows=arr.length;

int cols=arr[0].length;

int sum=0;

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

{

for(int j=0;j<cols;j++)

{

if(i>j)

{

sum += arr[i][j];

}

}

}

System.out.println(sum);

}

}

**Class 31:-**

Q) Write a java program to display array elements in spiral form?

input:

1 2 3

4 5 6

7 8 9

output:

1 2 3 6 9 8 7 4 5

ex:

---

class Test

{

public static void main(String[] args)

{

int[][] matrix={

{1,2,3},

{4,5,6},

{7,8,9}

};

int rows=matrix.length;

int cols=matrix[0].length;

int left=0;

int right=rows-1;

int top=0;

int bottom=cols-1;

while(true)

{

if(left>right)

{

break;

}

for(int i=left;i<=right;i++)

{

System.out.print(matrix[top][i]+" ");

}

top++;

if(top>bottom)

{

break;

}

for(int i=top;i<=bottom;i++)

{

System.out.print(matrix[i][right]+" ");

}

right--;

if(left>right)

{

break;

}

for(int i=right;i>=left;i--)

{

System.out.print(matrix[bottom][i]+" ");

}

bottom--;

if(top>bottom)

{

break;

}

for(int i=bottom;i>=top;i--)

{

System.out.print(matrix[i][left]+" ");

}

left++;

}

}

}

Anonymous Array

=================

Sometimes we will declare an array without name such type of nameless array is called anonymous array.

The main objective of anonymous array is just for instance use.

We can declare anonymous array as follow.

ex:

new int[]{10,20,30};

new int[][]{{1,2,3},{4,5,6}};

ex:

---

class Test

{

public static void main(String[] args)

{

System.out.println(sumOfArrayElements(new int[]{6,9,1,3,7,2}));

}

public static int sumOfArrayElements(int[] arr)

{

int sum=0;

for(int i:arr)

{

sum+=i;

}

return sum;

}

}

Interview Program

-----------------

Q) Write a java program to perform sum of array elements?

input:

6 9 1 3 7 2

output:

28

class Test

{

public static void main(String[] args)

{

//code Here

}

public static int sumOfArrayElements(int[] arr)

{

//Code Here

return sum;

}

}

solution

----------

class Test

{

public static void main(String[] args)

{

int[] arr={6,9,1,3,7,2};

System.out.println(sumOfArrayElements(arr));

}

public static int sumOfArrayElements(int[] arr)

{

int sum=0;

for(int i:arr)

{

sum+=i;

}

return sum;

}

}

OOPS

=====

OOPS stands for Object Oriented Programming System/Structure.

OOPS allows us to deals with real world entities using programming language.

We have following important features present in oops.

ex:

class

object

Abstraction

Encapsulation

Inheritance

Polymorphism

class

==========

A class is a blue print of an object.

A class is a logical entity.

A class is a collection of objects.

We can declare a class as follow.

syntax:

optional

|

modifier class class\_name <extends> parent\_classname

<implements> interface\_name

{

-

- //set of objects

-

}

A class will accept following modifiers.

ex:

default

public

abstract

final

Q) What is difference between default class and public class?

default class public class

--------------- -------------

To declare default class we should not To declare public class we should use

use any modifier. public modifier.

ex: ex:

class A public class A

{ {

} }

If we declare any class as default then we If we declare any class as public then we can

can access that class within the package. access that within the package and outside the

package.

Q) What is difference between abstract and final class?

abstract class final class

------------------ ----------

To declare abstract class we should use To declare final class we should use

abstract modifier. final modifier.

ex: ex:

abstract class A final class A

{ {

} }

Object creation is not possible. Object creation is possible.

Child creation is possible. Child creation is not possible.

c