What is java?

Java is a cross platform object oriented programming language developed by Sun microsystems.

What is Object Oriented Programming?

Object Oriented Programming is programming paradigm based on the concept of classes and objects(which are the instances of class).In oop classes are used to bind attributes and behaviour together as a wrapper and objects are used as a way to access attributes and execute methods.

OOP consists of:

Inheritance,Abstraction,Encapsulation,Polymorphism,Class,Object.

Inheritance:

inheritance is a concept of inheritating properties from one to class another.

Abstraction:

Abstraction is the concept in which only the essential data is shown to the user.

Encapsulation:

Encapsulation is the concept of binding together code and data that it manipulates.

Polymorphism:

The word polymorphism means having many forms.This is further divided into 2 types compile time polymorphism and run time polymorphism.

Inheritance in Java?

Inheritance is a property of oop in which an object acquires all properties and behaviours of the parent object.

It increases reusablity of methods and variables as well as properties.

Inheritance is also referred as IS-A relationship or a parent-child relationship.

There are 2 main reasons to use inheritance:

1. for method over-riding

and

2. for code reusablity

inheritance is achieved using extends keyword.

Class child extends parent{

}

multiple inheritance is not supported in jav,rather it supports multilevel inheritance

multiple inheritance is not supported in java since it creates a complexity,for example, a child class inherits two parent classes and calls a method which is present in both parent classes, then it creates a complexity about which method to run.

Hence,multi-level inheritance.

Abstraction in Java?

Encapsulation In Java?

Polymorphism In Java?

Which is the latest version of java?

The latest version of java or JDK is 14 (written on 28 june 2020,might be outdated)

What is JDK,JRE,JVM?

Jdk is one of the 3 core technology packages which make java.(other two are JRE and JVM)

JDK stands for Java Development Kit

JRE stands for Java Runtime Enviornment

JVM stands for java Virtual Machine

Jvm is the java platform component that executes the program

jre is the on disk part of java that creates jvm

jdk allows developers to create java programs that can be run by jre and jvm.

Jdk is the package of tools used to develop java programs,whereas jre is the package to tools to run java program.

Jre is the part of jdk

Jdk consists of jre,Interpreter(java),compiler(javac),archiver(jar),documentation generator(javadoc),etc.

Jre is also written as java RTE(Runtime enviornment)

Jre consists of minimum requirements to run a java program.

Jre consists of jvm,core classes,supporting files.

Jre is the specification where working of jvm is specified(though the jdk provider might choose their own algorithms while executing)

Whenever a java program is run on command line,a instance of JVM is created.

What is JVM?

Jvm is a virtual machine that is used to execute programs written in java or in other languages and are converted to java bytecode.

What is a virtual machine?

A virtual machine is an isolated computer enviornment where all resources are packed together so that no external os interaction is needed (except starting of virtual machine/hypervisor)

What is a compiler,interpreter,jar and javadoc?

Compiler transforms high level language program into machine code at once and then run the program,whereas interpreter does the same but line by line and executes each.

Compiled code runs faster than interpreted code.

Jar stands for java archive and are saved with the extension .jar

jar is an aggregation of java classes,images,audios and other corresponding medias together.

Jar files are generally used to distribute java application software.

Javadoc tool is a document generator api for java programs,it creates a documentation in html format.For proper java documentation it is necessary to have proper comments in the program.

It is a part of jdk.

Explain jvm architecture?

Now when a java file is compiled it is converted in to a .class file,this .class file is given as input to jvm as it contains bytecode that can be executed.jvm architecture is made up of:

1.classloader subsystem:

classloader subsystem makes the dynamic class loading possible in java.It loads,links and initialises class when it is referred for the first time during run time.

Bootstrap classloader is used to load the highest priority classes inside rt.jar.

Extension classloader is used to load classes inside ext folder

Application classloader is used to load application level classes.

The classloaders will follow a delegation hierarchy algorithm while loading classes.

After loading the second step is linking.

A bytecode verifier verifies the bytecode.

Then memory is allocated for all static variables and a default value is given to them.

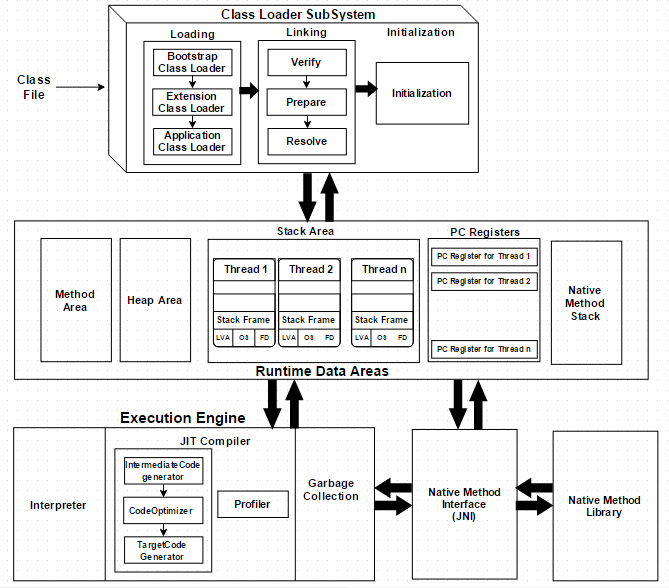
Finally all symbolic references are replaced with original references i.e. all symbols are replaced with their actual address.

In the last step,all static variables are initialised and the static block is executed.

2.Method area:

All the class level data is stored inside method area including the bytecodes of all methods,static variables.There is only one method area per jvm,it is a shared area.

3.Heap Area:



Explain Java memory management.

Explain Method area.

Explain Delegation hierarchy Algorithm?

What classes are inside ext folder?

What are application level classes in java?

How to take input in java?

How to use javadoc?

How are jar files created and executed ?

Which are different popular jdks?

How to run a java program?

1.save your java code in ,java extension file

2.compile the code with command ‘javac file\_name.java’

3.now a bytecode(.class file) will be created

4.run your code with command ‘java class\_name’

what is the name of java compiler?

Javac

what is the name of java interpreter?

Java

what are different data types in java?

what is Bytecode?

How to see bytecode?

What are the features of java?

Jvm has 3 main sections:

1.class loader subsystem

2.Runtime data area

3.Execution engine

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

What is dynamic class loading in java?

Dynamic class loading is a feature of java that let’s the user to create class objects during runtime.This makes it possible for the programmer to create class objects without actually knowing the class name.

Example:

import java.util.\*;

class a{

a(){

System.out.println("Constructor A");

}

}

class c{

c(){

System.out.println(“Constructor C”);

}

}

class b{

public static void main(String args[]) throws Exception{

Scanner s=new Scanner(System.in);

String c;

c=s.nextLine();

Class z=Class.forName(c);

Object o=z.newInstance();

}

}

o/p:

a

Constructor A

What is the difference between c++ and java?

What are the applications of java?

What is a class?

A class is prototype of an objects,It defines an object’s behaviour and it’s attributes.It acts as a wrapper for binding them together in the form of methods and variables.

Example:

every person is different , we have different color,different face,height,weight but in wider scope we all are human.So in this case class human will define what are the basic properties of human are and then it’s objects will define what the particular person is.

Syntax:

class Class\_Name{

}

What is a object?

Object is an instance of a class, a real time entity that follows the properties defined by the class.

Eg: human is class whereas I am an object.

Syntax:

Class\_Name ObjectName;

What is a method?

Method is a block of code that runs only when it is called,A method in java can have arguments/parameters,return type and modifier.

Syntax:

Modifier Return\_Type Method\_Name(Return\_Type Argument1\_Name,Return\_Type Argument2\_Name){

}

What is an constructor?

A constructor is similar to a method. Every class has atleast one constructor.A constructor is called when an instance of a class is created using the new keyword.As soon as an object is created it’s constructor is called,now this constructor initialises the instance variables and hence allocates the memory.If a class does not have a constructor a default constructor is provided by the compiler.

A constructor does not support return type or modifiers except access modifiers.Constructor has the same name as that of class.A constructor does not return any value.

Constructor is only called if new is used(class\_name Object\_Name=new Class\_Name();),

it on be called if new is not used.(class\_name object\_name;)

Syntax:

Access\_Modifier Class\_Name(Return\_Type Argument\_Name){

}

Difference between Java constructor and java method?

Constructor:

no return type,no modifiers(except access modifiers),atleast one constructor is always present in a class,does not return a value,

Methods:

return type,modifiers,a class can exist without a method,returns a value

what are modifiers?

Modifiers are the keywords that either control access or control functionality

they can be categorized as:

access modifier: that control access

eg:private,public,default,protected

non-access modifier: controls functionality

eg:final,abstract,static,transient,synchronized,volatile,native

Explain each access modifier?

Access modifiers can be divided in 2 categories:

1.that are applicable on classes : public,default

2.that are applicable on functions : private,public,protected,default

private : can be accessed only within the class

public: can be accessed from everywhere(within class,outside class,within package,outside package)

default: can be accessed everyehere within package

protected: can be accessed everywhere within package and can be accessed outside package using a child class.

Why are private constructors are used?

If we make a constructor private,then we cannot create it’s instance(using new) outside the class.

Can a class be private or protected?

Yes a class can be private or protected if and only if it is a nested class.

What is a nested class?

Class defination inside class is called as nested class.

Syntax:

class a{

class b{

}

}

nested classes can further be divided into static and non-static nested classes.Non static nested classes are called as inner classes.

Syntax:

class a{

static class d{

void ff(){

System.out.println("fffff");

}

}

public static void main(String args[]){

a.d d2=new a.d();

d2.ff();

}

}

nested class can use more access modifiers than outer classes such as public,private,protected and default.

class a{

static protected class d{

void ff(){

System.out.println("fffff");

}

}

public static void main(String args[]){

a.d d2=new a.d();

d2.ff();

}

}

Compelling reasons for using nested classes include the following:

* It is a way of logically grouping classes that are only used in one place: If a class is useful to only one other class, then it is logical to embed it in that class and keep the two together. Nesting such "helper classes" makes their package more streamlined.
* It increases encapsulation: Consider two top-level classes, A and B, where B needs access to members of A that would otherwise be declared private. By hiding class B within class A, A's members can be declared private and B can access them. In addition, B itself can be hidden from the outside world.
* It can lead to more readable and maintainable code: Nesting small classes within top-level classes places the code closer to where it is used.

Explain each non access modifier?

static keyword is used to create variables that can exist without creating an object of the class.

Only one copy of static variable exists no matter how many instances you use it for.Ststic variables are also known as class variables.Making class variablers static means we can access them using their class name only(without making an object).static methods are also known as class methods.

Example:

class a{

static int a1;

public static void main(String args[]){

gg d=new gg();

hh b=new hh();

System.out.println(a.a1);

}

}

class gg{

gg(){

a.a1=1;

}

}

class hh{

hh(){

a.a1=2;

}

}

o/p:

2

static keyword can be used on variables,methods,blocks,nested classes.The variables that must be common to all instances must declared static.

Example:

class a{

static int a1;

public static void main(String args[]){

gg d=new gg();

hh b=new hh();

a gi=new a();

gi.a1=3;

System.out.println(a.a1);

}

}

class gg{

gg(){

a.a1=1;

}

}

class hh{

hh(){

a.a1=2;

}

}

o/p:

3

A static variable gets memory assigned only once during class loading.Use static variables is memory efficient.

#### Java static property is shared to all objects.

Only static methods/static data members can be directly called inside a static method(else you have to create an object)

example:

class a{

void ff(){}

public static void main(String args[]){

ff();

}

}

o/p:

stat.java:4: error: non-static method ff() cannot be referenced from a static context

ff();

^

1 error

example:

class a{

int a;

void ff(){}

public static void main(String args[]){

a=10;

}

}

o/p:

stat.java:5: error: non-static variable a cannot be referenced from a static context

a=10;

^

1 error

this and super cannot be used in static context.

Example:

class a{

static int a;

void ff(){}

public static void main(String args[]){

this.a=10;

}

}

o/p:

stat.java:5: error: non-static variable this cannot be referenced from a static context

this.a=10;

^

1 error

example:

class a{

a(){

super();

}

static int a;

void ff(){}

public static void main(String args[]){

super();

}

}

o/p:

stat.java:8: error: call to super must be first statement in constructor

super();

^

1 error

Java Static Block is executed before the main class at the time of class loading.It is used to initialise static dat members

example:

class a{

a(){

System.out.println("Constructor");

}

static{

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

}

public static void main(String args[]){

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

}

}

o/p:

static block

main method

final keyword can be used on variables,methods and classes.Java final keyword restricts value change,restricts method overriding,restricts inheritance.final variable must be initialised a value at the time of declaration.

Example:

class a{

final int g;

public static void main(String args[]){

}

}

o/p:

stat.java:2: error: variable g not initialized in the default constructor

final int g;

^

1 error

example:

class a{

final static int g=10;

public static void main(String args[]){

g=20;

}

}

o/p:

stat.java:4: error: cannot assign a value to final variable g

g=20;

^

1 error

if a method is made final it cannot be over ridden.

Example:

class a{

final void hh(){

int k;

}

}

class b extends a{

void hh(){

int j;

}

public static void main(String args[]){

}

}

o/p:

stat.java:7: error: hh() in b cannot override hh() in a

void hh(){

^

overridden method is final

1 error

If you make a class final,you cannot extend it.

final class a{

}

class b extends a{

public static void main(String args[]){

}

}

o/p:

stat.java:3: error: cannot inherit from final a

class b extends a{

^

1 error

A final method can be inherited but not over ridden.

Example:

class a{

final static void kk(){

System.out.println("final method");

}

}

class b extends a{

public static void main(String args[]){

kk();

}

}

o/p:

final method

you can declare a blank final variable but can initialise it only in constructor.(and that too only once).

Example:

class a{

final int j;

a(){

j=100;

}

public static void main(String args[]){

a i=new a();

System.out.println(i.j);

}

}

o/p:

100

Explain what is package?

Packages are used to keep related classes together, they are like folders.Packages are used to avoid name conflicts.They help in making the classes maintainable.

There are 2 types of packages:

1 built in packages

2 user defined packages

importing a package:

import java.packageName.\*; or import java.packageName.\*;

write a class inside a package and creating the package:

package PackageName;

class ClassName{

........

}

then save the code and compile it using javac -d . NameOfFile.java

[Don’t forget the dot in javac -d command]

to run the file java PackageName.ClassName

ways to access classes from different package classes :

1.import PackageName.\*;

2.import PackageName.ClassName;

3.PackageName.ClassName s=new PackageName.ClassName();

What is the standard defination of package name?

Domain.company.package

What is classpath?

if you import a package sub packages will not be imported.

What is an interface?

Name all built in java packages?

What is the difference between folder and file directory?

Explain what is inheritance?

How to access class outside package?

What are non static nested classes or inner classes?

What is this in java?

What is super in java?

Explain is java main method?

Can we execute the program without main() method?

It is possible if you are using jdk 1.6 or less,from 1,7 it became compulsary to use the main method.

In jdk 1.6 or less it was done using the static block

Difference between System.exit() and return?

Explain all different data types in java?