Interview Questions

CORE JAVA

**1)What is Java?**

Java is an object-oriented programming language Which follows the concepts like Abstraction Polymorphism and Inheritance Encapsulation.it is platform independent.

Java has some features like simple, robust and secure, portable and multithreaded.

**2)What are the features of java?**

Java features also known as buzz words. Some important features of java are

* Simple
* Object-Oriented
* Portable
* Platform-Independent
* Secured
* Robust
* Multithreaded
* **Simple**: Java has syntax similar to Cpp which is considered as the simple programming language that anyone can learn very easily. Java don't have pointer and operator overloading concepts.
* **Object-Oriented**: Object-oriented programming (OOPs) is a methodology that simplifies software development and maintenance by providing some rules. and it also provides reusability, modularity and readability.so it makes programming comfortable.
* **Portable**: Java bytecode can be executed in any platform.
* **Platform-Independen**t: Java platform differs from most other platforms in the sense that it is a software based platform that runs on the top of other hardware-based platforms. It has two components namely Runtime Environment, API(Application Programming Interface).
* **Secured**: It is secured because JVM has

**Class loader**: Its adds security by separating the package for the classes of the local file form those that are imported from network sources.

**Bytecode Verifier**: checks the code fragments for illegal code that can violate access right to objects.

**Security Manager**: determines what resources a class can access such as reading and writing to the local disk.

**3)What are the Different types of applications you can build with JAVA ?**

There are mainly four kind of applications we can build using java

**Standalone applications, enterprise applications, web applications, mobile applications**

**Standalone Applications:** or desktop applications is kind of applications that we need to install on every machine such as media player antivirus.

AWT and Swing are used in java for creating standalone applications

**Enterprise Applications:** this kind of applications are distributed in nature, such as banking applications etc.it has the advantage of high level security, load balancing, clustering

In java EJB is used for creating enterprise applications.

**Web Applications**: these are the applications that runs on the server side and generates dynamic web pages.in java we use servlets, JSPs, JSF, Struts technologies to develop web applications

Mobile Applications: An application that is created for mobile devices. currently Android and java ME are used for creating mobile applications

**4)What is path and classpath?**

Path and Classpath both are environment variables

Path is used by the operating system to locate the JDK binaries like “java”or “javac” commands and other executables which are used to run java programms. where as classpath is used by the JDK or JVM or java classloaders to find where the .class or .jar files are located.

**5)  Difference between JDK, JRE and JVM ?**

**JVM  :** It is an abstract machine which loads, verifies and executes the code, provides the runtime environment for the byte code to be executed. It directly comes in contact with OS but the code within the JVM never interacts with it which makes it more secured.

**JRE :** It is the implementation of the JVM. It has got all the libraries + all the files used by JVM at run time. It provides the runtime environment.

**JDK :** It consists of JRE and  utilities tools like javac , java for compilation and execution of java and bytecode respectively. In addition it also consists of development tools like javadoc e.t.c.

**6)Explain is the Java memory Model?**

Java memory model mainly consists of 5 areas.

**Class Area**: It consists of the static variables.

**Heap:** all The objects created are stored in heap.

**Stack:** It holds the primitive variables, partial results and local variables.

**Program Counter:** It consists of the address of the current instruction being executed.

**Native Methods:** it consists of the methods of other programming languages like C.

**7)Different data types in Java?**

              Data types represents different values to be in the variable. In java there are two types of data types.

                 1)  Primitive (Boolean, character, byte, short, int, long, double, float)

                 2)  Non-Primitive (String, Array)

**8)  Explain the design of Java class?**

              In java we design program using the template like

1. Package Name (domain Name. company Name. Project Name. functionality)
2. Import statements
3. Class Declaration (Access Modifier  classkeyword  ClassName)
4. Properties or data members (Access Modifier   data Types   variable Name)
5. Method Declaration (Access Modifier returnType methodName())
6. Comments

**9)Different types of operators?**

* Arithmetic Operator(\* / % + -)
* Assignment Operator(= += -= \*= /= %= &= ^= |= <<= >>= >>>=)
* Relational Operator(< > <= >= instanceof == !=)
* Bitwise Operator(& ^ |)
* Logical Operator(&& ||)
* Unary Operator(expr++ expr-- ++expr --expr +expr -expr ~ !)
* Ternary operator(? :)

10) **Top 5 operators ?**

These are the operators we use many times while developing applications

1) .(dot) operator

2) = operator

3) == operator

4) && and || operators

5) + operator

**11) Explain how each control statement works ?**

**(if-else, for, while, do-while, ternary, switch, break and continue)**

 These control statements can be classified into three groups: Decision making statements, repetition statements and branching statements.

**if:**

The Java if statement is used to test the condition. It checks boolean condition: true or false. There are various types of if statement in java.

* if statement
* if-else statement
* nested if statement
* if-else-if ladder

if(condition1){

//code to be executed if condition1 is true

}else if(condition2){

//code to be executed if condition2 is true

}

else if(condition3){

//code to be executed if condition3 is true

}

...

else{

//code to be executed if all the conditions are false

}

**Switch:** The Java switch statement executes one statement from multiple conditions.

**switch**(expression){

**case** value1:

 //code to be executed;

**break**;  //optional

**case** value2:

 //code to be executed;

**break**;  //optional

......

**default**:

code to be executed **if** all cases are not matched;

}

The java switch statement is fall-through. It means it executes all statement after first match if break statement is not used with switch cases.

**For Loop:** The Java for loop is used to iterate a part of the program several times. If the number of iteration is fixed, it is recommended to use for loop.

* Simple For Loop

**for**(initialization;condition;incr/decr){

//code to be executed

}

**For-each or Enhanced For Loop:** The for-each loop is used to traverse array or collection in java

It works on elements basis not index. It returns element one by one in the defined variable

**for**(Type var:array){

//code to be executed

}

* Labeled For Loop: We can have name of each for loop. To do so, we use label before the for loop. It is useful if we have nested for loop

labelname:

**for**(initialization;condition;incr/decr){

//code to be executed

}

If you use two semicolons ;; in the for loop, it will be infinitive for loop.

**for**(;;){

//code to be executed

}

**While Loop:**

The Java while loop is used to iterate a part of the program several times. If the number of iteration is not fixed, it is recommended to use while loop.

**while**(condition){

//code to be executed

}

If you pass **true** in the while loop, it will be infinitive while loop.

**while**(**true**){

//code to be executed

}

**Do-While Loop:**The Java do-while loop is used to iterate a part of the program several times. If the number of iteration is not fixed and you must have to execute the loop at least once, it is recommended to use do-while loop. The Java do-while loop is executed at least once because condition is checked after loop body.

**do**{

//code to be executed

}**while**(condition);

**Break Statement:** The Java break is used to break loop or switch statement. It breaks the current flow of the program at specified condition. In case of inner loop, it breaks only inner loop.

**Continue Statement:** The Java continue statement is used to continue loop. It continues the current flow of the program and skips the remaining code at specified condition. In case of inner loop, it continues only inner loop.

**Ternary Operator:**

**class** OperatorExample{

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

**int** a=2;

**int** b=5;

**int** min=(a<b)?a:b; //a<b is true so it gives a from a:b if a<b is false it gives b

System.out.println(min);

}} // Output:2

**12)What is class and object?**

Class: Class is a template which represents state and behaviour of an object. In java state is implemented through properties and behaviour is implemented through methods. class is collection if objects.

     Object: Object is instance of a class.

**13) What are different OOPS concepts?**

o   Abstraction

o   Polymorphism

o   Encapsulation

o   Inheritance

**14)What is inheritance in java?**

Inheritance is one of the object oriented programming concept where one object acquiring all the properties and behaviour of another object from one class to another class .the extends keyword is used to extend the one class to another. the class which extends another class is called subclass or child class And the extended class is called parent class or superclass. In order to extend one class to another there must be IS -A relationship between these classes.

Using Inheritance concept we can achieve the code Reusability.

Private properties and methods and constructors of the super class can not be inherited in subclass.

And we can achieve inheritance through interfaces also.

**15)What is Abstraction?**

      Abstraction is one of the object-oriented concept of hiding implementation details and showing essential features or functionality of the object to the user.

There are two ways to achieve abstraction:

1. Abstract class.
2. Interface.

If we use abstract class we can achieve 0-100% abstraction where as if we use interface we can achieve 100% abstraction.

Abstraction mainly comes into picture for future flexibility when the developer is not sure of the implementation

**16)What is Encapsulation?**

Encapsulation is one of the object orientedconcept which enforces protecting variables, functions from outside of the class by making them private using private keyword and providing access only through getters and setters . using encapsulation, we can provide security.

**17)What is the difference between abstraction and encapsulation?**

Abstraction and Encapsulation are two important [Object oriented programming concept](http://javarevisited.blogspot.sg/2012/03/10-object-oriented-design-principles.html)s and they are completely different to each other.

**Abstraction** is a process of hiding the implementation details and showing only functionality to the user.

**Encapsulation in java** is a process of wrapping data and code(methods) which is acting on data together into a single unit.

First difference between Abstraction and Encapsulation is that, Abstraction is implemented in Java using [interface](http://javarevisited.blogspot.sg/2012/04/10-points-on-interface-in-java-with.html) and abstract class while Encapsulation is implemented using [private](http://javarevisited.blogspot.sg/2012/03/private-in-java-why-should-you-always.html) keyword.

Design principles "[programming for interface than implementation](http://javarevisited.blogspot.sg/2012/06/20-design-pattern-and-software-design.html)" is based on abstraction and "encapsulate whatever changes" is based upon Encapsulation.

**18) What is Abstract class?**

A class which is declared as abstract using abstract keyword is called abstract class.  It may or may not include abstract methods which means in abstract class you can have concrete methods as well along with abstract methods .the abstract class must be extended and abstract methods must be implemented and you cannot instantiate an abstract class.

Using abstract class, we can achieve 0 to 100% abstraction.

We can use Abstract class to make our software flexible enough to support future changes.

**19)What is interface?**

Interface is a mechanism to achieve abstraction. There can be only abstract methods. It is used to achieve abstraction and multiple inheritance in java.

**20)What is Polymorphism?**

Polymorphism is the concept of an object’s ability to take on many forms. There are two types of polymorphism:

     1.Compile time polymorphism

                                   2.Runtime polymorphism.

The polymorphism which occur during method overloading is called compile time polymorphism because the method calling will be resolved at compile time

The polymorphism which occur during method overriding is called runtime polymorphism because the method calling will be resolved at runtime depending upon the actual object and not the type of variable.

***21)Why you should write Polymorphic code?***

Simple to be flexible, to accommodate change and to take advantage of evolution on later stage of development. A static code is fixed when written, but a Polymorphic code can evolve.

**22)  What is method overloading and overriding?**

Method overloading is the **concept** of a class having multiple methods of same name but vary with type and number of arguments. It enhances the readability. It is performed within a class. method overloading is resolved at compile time

Method overriding is the **concept** of implementing a method in the sub class which is already provided in the super class. Both classes should compulsorily be in an IS-A relation.  The subclass method must have the same method name and the parameters. method overriding is resolved at runtime.  
  
 You cannot override private , static and final methods from a superclass.

**23) Can we prevent overriding a method without using the final modifier?**   
Yes, you can prevent the method overriding in Java without using the final modifier. In fact, there are several ways to accomplish it e.g. you can mark the method private or static, those cannot be overridden.

**24) What is the difference between the extends and implements keywords in java?**

Though both extends and implements keyword in Java is used to implement [Inheritance](http://www.java67.com/2016/03/top-21-java-inheritance-interview-Questions-Answer-Programming.html) concept of Object-Oriented programming, but there is difference between them. The extends keyword is mainly used to extend a class i.e. to create a subclass in Java, while implements keyword is used to implement an interface in Java. The extends keyword can also be used by an interface for extending another interface.

**25)What is the difference between the class and interface?**

Class is a template which represents state and behaviour of an object .where as interface is also a special type of class which have abstract methods and variables which are static and final.

A class can not extend more than one class at a time .and a class can implement more than one interface because an interface can extend more than one interface.

You can create an object for a class but you can not create an object for an interface(because it is completely abstract)

Usually we represent a class names with a noun and interface names with an adjective.

26) **What is static in JAVA ?**

Static is a keyword in java which you can apply towards variables, methods and blocks and inner classes

If you apply towards variables the variable becomes class level variable and if you apply towards methods we need not to create the instance of the class to call the method and if we apply towards blocks these blocks will be executed before the constructor.

this and super keywords can not be used in static methods.

**27)What is final in java?**

final is a keyword in java which you can apply towards variables, methods and classes

If we apply towards variables the value can not be changed. and if we apply towards methods we can not override the method in the child class. if you apply towards classes we can not extend the class.

28) **What is final, finally, finalize ?**

**final:** final is a keyword in java which you can apply towards variables, methods and classes

If we apply towards variables the value can not be changed. and if we apply towards methods we can not override the method in the child class. if you apply towards classes we can not extend the class.

**finally:** finally is another Java keyword which is used in Exception handling concept along with try, catch, [throw and throws](http://javarevisited.blogspot.sg/2012/02/difference-between-throw-and-throws-in.html).

We write finally block after the try or catch blocks. finally block have an advantage that no matter what the code inside the block will be executed whether the exception occurred or not, handled or not.so we write important code like close system resource e.g. [InputStream](http://javarevisited.blogspot.sg/2012/08/convert-inputstream-to-string-java-example-tutorial.html) or OutputStream and closing  network connection, database connection

**finalize:** finalize() is a special method in Java which is called by [Garbage Collector](http://javarevisited.blogspot.sg/2011/04/garbage-collection-in-java.html) before object is eligible for garbage collection. This is the last chance for an object to perform any cleanup activity.

**29) What is static and dynamic binding ?**

Association of method definition to the method call is known as binding.

**Static binding:**

The binding which can be resolved at compile time by compiler is known as static or early binding. All the static, private and final methods have always been bonded at [compile-time](http://beginnersbook.com/2013/04/runtime-compile-time-polymorphism/) .because Compiler knows that all such methods cannot be overridden and will always be accessed by object of that particular class. Hence compiler doesn’t have any difficulty to determine object of class.

Dynamic binding:The binding which can be resolved at run time by JVM is known as dynamic or late binding.Overriding is a perfect example of dynamic binding as in overriding both parent and child classes have same method. Thus while calling the overridden method, the compiler gets confused between parent and child class method. this will be resolved by JVM at runtime.

**30)** **Why do we create abstract classes in application development ?**

A key challenge while writing software Java Programs is not just to cater today's requirement but also to ensure that nurture requirement can be handled without any architectural or design change in your code. In short, your software must be flexible enough to support future changes.

Example: Though there are lot of [difference between abstract class and interface](http://java67.blogspot.sg/2012/09/what-is-difference-between-interface-abstract-class-java.html), key thing to remember is that they both provides abstraction. Let's take an example, you need to design a program, which can produce reports for employees e.g. how many hours they worked and how much they are paid every month. Let's assume that currently your organization only has permanent employees, which are paid monthly. You know that and you write code based upon that, after sometime your company started recruiting contract employees, which are paid at hourly rate rather than monthly salary. Now, if you need to rewrite your program to support this, they your program is not flexible enough. On the other hand, if you just needs to write some code to plug this new type of employee into system, they your program is very much flexible and maintainable. If you would have known about [abstract class](http://javarevisited.blogspot.sg/2013/04/10-abstract-class-and-interface-interview-question-java-answers.html), you would have made Employee an abstract class and methods like salary() abstract, because that is what varies between different types of employees. Now introducing a new type of Employee would be cake walk, all you need to do is to create another subclass of Employee to represent Contract Employee, and their salary method return salary based upon number of hours they had worked multiplied by their hourly rate. So, you get the idea right, we code at certain level of abstraction, which allows us to accommodate new changes in our system.

**31) why do we create interfaces in application development ?** it must have three distinct characteristics: reusability, maintainability, and extensibility.

**https://dzone.com/articles/programming-when-use**

**32)** **What are different types of access modifiers ?**

1. Default – No keyword required
2. Private
3. Protected
4. Public

**Default**: When no access modifier is specified for a class , method or data member – It is said to be having the **default** access modifier by default.

* The data members, class or methods which are not declared using any access modifiers i.e. having default access modifier are accessible **only within the same package**.

**Private**: The private access modifier is specified using the keyword **private**.

* The methods or data members declared as private are accessible only **within the class** in which they are declared.
* Any other **class of same package will not be able to access** these members.

**protected**: The protected access modifier is specified using the keyword **protected**.

* The methods or data members declared as protected are **accessible within same package or in sub classes in different package through inheritance .**

**public**: The public access modifier is specified using the keyword **public**.

* The public access modifier has the **widest scope** among all other access modifiers.
* Classes, methods or data members which are declared as public are **accessible from every where** in the program. There is no restriction on the scope of a public data members.

**33)What is call by value?**

Call by value is a mechanism where we pass the parameters to a method by calling that method.java supports only call by value.

Usually we pass two kinds of parameters primitive types or Object references

When we pass primitive type as parameters, value held in the variable that is passed as an argument is copied into the parameters that are defined in the method header. That is why changes made to the variable within the method had no effect on the variable that was passed.

when an object reference is passed to a method, the method gets a value of the object reference, and both the actual and the formal parameters refer to the same object, therefore within from the method the state of an object parameter can be changed.

34) **What is Association, Aggregation and composition?** a relationship between two classes through their objects is referred as an association, here the relationship is HAS-A realation. an association is known as composition when one object *owns*other or when object higly depends on another, while an association is known as aggregation when one object uses another object.

Another example of **Aggregation** is Student in School class, when School closed, Student still exist and then can join another School or so.

example of **Composition** is Car and it's part e.g. engines, wheels etc. Individual parts of the car can not function when a car is destroyed.

**35)What is String in Java?**

String is an object that represents sequence of characters , java provides String class to create and manipulate strings. String objects are immutable.

There are two ways to create String object:

1. By string literal
2. By new keyword

String objects are stored in Special memory area known as String constant pool

**36)** **What is String constant pool ?**

 String Pool is a special area with **pool of strings** in [Java Heap Memory](http://www.journaldev.com/4098/java-heap-space-vs-stack-memory)**.** String Pool is possible only because [String is immutable in Java](http://www.journaldev.com/802/string-immutable-final-java)

When we use String literals to create a String, it first looks for String with same value in the String pool, if found it just returns the reference else it creates a new String in the pool and then returns the reference.

However using new operator, we force String class to create a new String object in heap space. We can use intern() method to put it into the pool or refer to other String object from string pool having same value.

37**) Why String is immutable ?**

Most of the data is represented in the form of strings in our application development .so there is a hign chance that two Strings may point to the same value. If a string changes, it will affect the other string which is being referenced to the same pointer and there are chances that it might become unreferenced which gets removed by the automatic garbage collector. This is the reason why strings are made immutable.

http://www.java67.com/2014/01/why-string-class-has-made-immutable-or-final-java.html

**38)** **How do you make a class immutable ?**

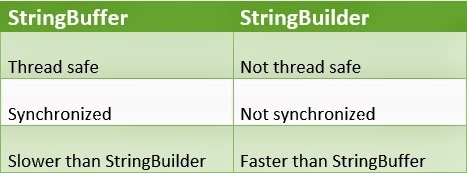
By declaring a class final.

By declaring the properties of the class as final.

By not providing the setters.

<http://javarevisited.blogspot.com/2013/03/how-to-create-immutable-class-object-java-example-tutorial.html>

**39) StingBuilder vs StringBuffer ?**



**40)** difference between **== vs equals**

== operator is used to used compare primitives along with the objects .equals method is used to check the equality of objects

If we use == operator to compare objects it checks the reference s of two Strings whether they are pointing to the same String or not based on this it returns true or false

If we use equals method it checks the content of String it returns true if the content is equal otherwise false.

**41)Why character array is better than String for storing password in java?**

Since **Strings are immutable in Java** if you store password as plain text it will be available in memory until Garbage collector clears it and since String are used in String pool for reusability there is pretty high chance that it will be remain in memory for long duration, which pose a security threat. Since any one who has access to memory dump can find the password in clear text and that's another reason you should always used an encrypted password than plain text. Since Strings are immutable there is no way contents of Strings can be changed because [any change will produce new String](http://javarevisited.blogspot.com/2011/07/string-vs-stringbuffer-vs-stringbuilder.html), while if you char[] you can still set all his element as blank or zero. So **Storing password in character array clearly mitigates security risk of stealing password**.

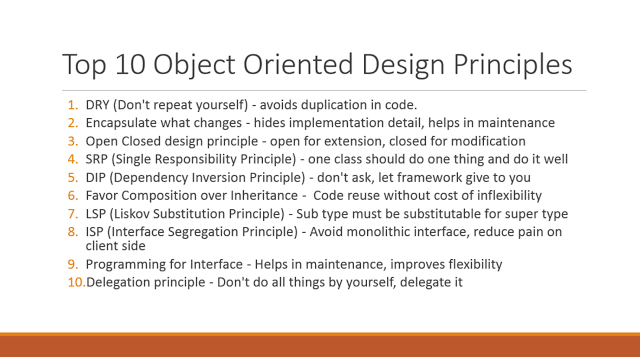
2) **Java itself recommends** using getPassword() method of JPasswordField which returns a char[] and deprecated getText() method which returns password in clear text stating security reason. Its good to follow advice from Java team and adhering to standard rather than going against it.

**42) Can you explain what does substring do?**

substring method is used to get parts of String in Java. It’s defined in java.lang.String class, and it's an [overloaded method](http://javarevisited.blogspot.com/2011/12/method-overloading-vs-method-overriding.html). One version of substring method takes just beginIndex, and returns part of String started from beginIndex till end, while other takes two parameters, beginIndex and endIndex, and returns part  of String starting from beginIndex to endIndex-1. every time you call  substring() method in Java,  it will return a new String because [String is immutable in Java](http://javarevisited.blogspot.com/2010/10/why-string-is-immutable-in-java.html).

**43)** **Top 5 methods in String class ?**

equals(), substring(), split(), chatAt(), indexOf(), equalsIgnore()

17) **Object Oriented Design Principles** ?  
Read more: <http://javarevisited.blogspot.com/2012/03/10-object-oriented-design-principles.html#ixzz4c1c2Cm6e>  
  
  
 **Servlets And JSP:**

**1)What is a servlet?**

Servlet is a server side technology which process the http request and generate the dynamic web pages .it is also an API that provides many interfaces and classes. In MVC design pattern servlets are considered as controller.

**2)What is the Difference between static and dynamic website or webpage?**

Static website is a set of webpages which is easy to create and it renders the same content in webpages for every request so here the content is prebuilt. The content is only changes when someone publishes or updates the file. Mostly static websites developed using HTML code.

Where as coming to dynamic website is also a set of webpages whose content changes dynamically for every new request that means the content is generated At the time of page loading.  It accesses content from a database or Content Management System (CMS). In java, We use Servlets, JSP, JSF,HTML to generate dynamic websites

**3)What are the different HTTP request methods?**

GET, POST, PUT, HEAD, TRACE, DELETE, OPTIONS

**4)Differences between GET and POST request methods?**

|  |  |
| --- | --- |
| **GET** | **POST** |
| 1) In case of Get request, only **limited amount of data**can be sent because data is sent in header. length of the URL is limited | In case of post request, **large amount of data**can be sent because data is sent in body. |
| 2) Get request is **not secured**because data is exposed in URL bar. | Post request is **secured**because data is not exposed in URL bar. |
| 3) Get request **can be bookmarked.** | Post request **cannot be bookmarked.** |
| 4) Get request is **idempotent**. It means second request will be ignored until response of first request is delivered | Post request is **non-idempotent.** |
| 5) Get request is **more efficient**and used more than Post. | Post request is **less efficient**and used less than get. |

# 5) [What is the difference between application server and web server?](http://stackoverflow.com/questions/936197/what-is-the-difference-between-application-server-and-web-server)

# In Java  Application server and web server both are used to host Java web application.

# main difference between web server and application server is support of EJB.

1**. Application Server** supports **distributed transaction and EJB**. While Web Server only supports Servlets and JSP.

2. Application Server can contain web server in them. most of App server e.g. JBoss or WebLogic ,Glassfish, WebSphere has Servlet and JSP container.

3. Though its not limited to Application Server but they used to provide services like **Connection pooling**, **Transaction management**, messaging, clustering, load balancing and persistence. Now Apache tomcat also provides connection pooling.

4. In terms of l*ogical difference between web server and application server*. web server is supposed to provide http protocol level service while application server provides support to web service and expose business level service e.g. EJB.

5. Application server are more heavy than web server in terms of resource utilization.  
6. A Web server exclusively handles HTTP requests, whereas an application server serves business logic to application programs through any number of protocols.

# <http://javarevisited.blogspot.com/2012/05/5-difference-between-application-server.html#ixzz4cMLXbqXZ>

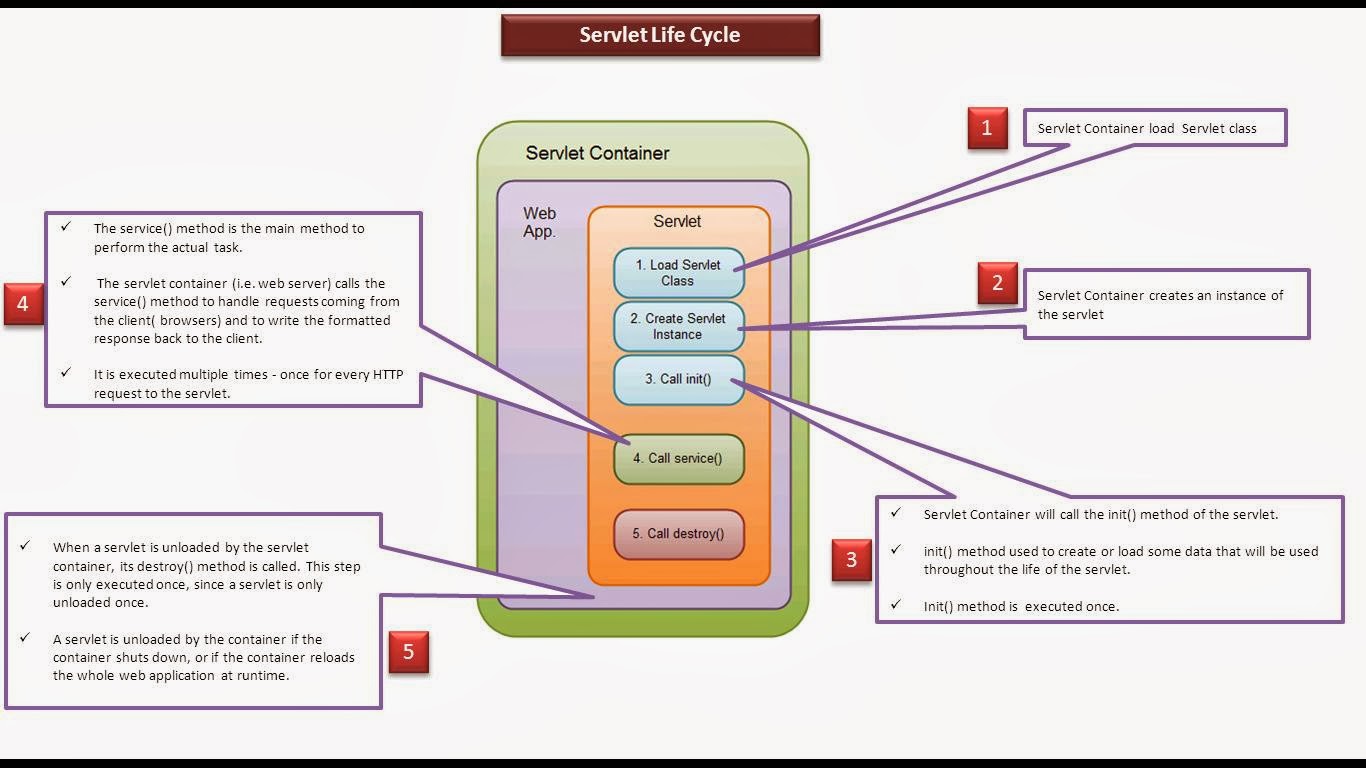
# http://www.javaworld.com/article/2077354/learn-java/app-server-web-server-what-s-the-difference.html

6)Explain about lifecycle of a servlet?

The web container maintains the life cycle of a servlet instance.

The lifecycle mainly consists of following steps

1. Servlet class is loaded. (only once when first request comes in lifecyle)
2. Servlet instance is created. (created only once in the servlet life cycle)
3. init method is invoked. (invoked only once in servlet life cycle)
4. service method is invoked. (invoked for every request comes by container)
5. destroy method is invoked. (invoked only once in lifecycle)



**7)What is web.xml or deployment descriptor?**

Web.xml or deployment descriptor is configuration file which is used by the web container to process the HTTP requests.

So web.xml consists of many elements like welcome file lists and servlets mappings and URL mappings, Initialization parameters and context parameters, load on start-ups, session timeout, Filters.

**8)What is RequestDispatcher in servlet?**

The RequestDispatcher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interface can also be used to include the content of another resource also. It is one of the way of servlet collaboration.

RequestDispatcher interface have two methods

1. **public void forward(ServletRequest request, ServletResponse response)throws ServletException, java.io.IOException:** Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.
2. **public void include(ServletRequest request, ServletResponse response)throws ServletException,java.io.IOException:** Includes the content of a resource (servlet, JSP page, or HTML file) in the response.

**9)What is sendRedirect in servlets?**

The **sendRedirect()** method of **HttpServletResponse** interface can be used to redirect response to another resource, it may be servlet, jsp or html file.

It accepts relative as well as absolute URL.

**public** **void** sendRedirect(String URL)**throws** IOException;

response.sendRedirect("http://www.google.com");

**10)What is ServletConfig?**

ServletConfig is an interface . An object of ServletConfig is created by the web container for each servlet. This object can be used to get configuration information from web.xml file.

If the configuration information is modified from the web.xml file, we don't need to change the servlet. So it is easier to manage the web application if any specific content is modified from time to time.

**getServletConfig() method** of Servlet interface returns the object of ServletConfig.

**11)What is ServletContext?**

ServletContext is an interface. An object of ServletContext is created by the web container at time of deploying the project. This object can be used to get configuration information from web.xml file. There is only one ServletContext object per web application.The ServletContext object can be used to set, get or remove attribute from the web.xml file.

If any information is shared to many servlet, it is better to provide it from the web.xml file using the **<context-param>** element.

**getServletContext() method** of ServletConfig interface returns the object of ServletContext.

**getServletContext() method** of GenericServlet class returns the object of ServletContext.

**12)What is Attribute in Servlet?**

An **attribute in servlet** is an object that can be set, get or removed from different Scopes like request scope, session scope, application scope

Using attributes, we can pass informations from one servlet to another servlet