<http://theopentutorials.com/tutorials/java/strings/using-tostring-method/>

<http://www.javajee.com/interface-vs-abstract-class>

<http://www.javatechinfo.com/2013/01/tricky-sql-queries.html>

**What is the Servlet ?**  
  
Java Servlet is a simple, consistent mechanism for extending the functionality of a web server. Servlets are precompiled Java programs that are executed on the server side. It requires a Servlet container to run in  
  
**Why Use Servlets?**  
  
1. Work well in a Heterogeneous Environments

* OS and platform neutral
* Work with all major web servers (IIS, Apache,etc..)

2. Well defined Web Architecture framework

* Standard built in services such as:
  + Standard Approach to Authentication using declarative security vice programmatic security
  + Database connection pooling
  + Complete support for sessions via cookies and/or URL re-writing
    - Has automatic fallback to URL re-writing

3. Clean separation of Controller/Logic from Presentation Efficient, scales very well  
  
**What Does Servlet Do ?** 

* Receives client request (mostly in the form of HTTP request)
* Extract some information from the request
* Do content generation or business logic process (possibly by accessing database, invoking EJBs, etc)
* Create and send response to client (mostly in the form of HTTP response) or forward the request to another servlet or JSP page

**What is the difference between servlets and applets ?**   
**Servlet :**

* subclass of GenericServlet
* Runs in a server
* must be multi threaded or thread safe
* no direct user interface

**Applets :**

* subclass of Applet
* Runs in a browser
* generally single thread per appplet
* uses AWT for user interface

**What is Web Deployment Descriptor ?**

* /web-inf/web.xml
  + Part of the standard
  + Defines servlets used in the web application
  + Maps servlets to URLs
  + A servlet can map to many URLs
* Defines resources available to the web app
* Defines security constraints
* Defines other stuff like
  + Welcome file list
  + Session timeout
  + Error page mapping

**What are the features of the Servlets ?**

* Tailored to interact with Web Server
* Server and platform independent
* Efficient and scalable
* Container provides additional functionality (i.e., authentication, cookies, etc.)

**What is a servlet request ?**

Information that is sent from client to a server

* Who made the request
* What user-entered data is sent
* Which HTTP headers are sent

**What is a servlet response ?**

Information that is sent to client from a server

* Text (html, plain) or binary (image) data
* HTTP headers, cookies, etc

**Explain Servlet life cycle ?**

The life cycle of a servlet is controlled by the container in which the servlet has been deployed. When a request is mapped to a servlet, the container performs the following steps.  
  
If an instance of the servlet does not exist, the Web container  
1) Loads the servlet class.  
2) Creates an instance of the servlet class.  
Initializes the servlet instance by calling the init method. Initialization is covered in Initializing a Servlet.  
3) Invokes the service method, passing a request and response object. Service methods are discussed in the section Writing Service Methods.  
4)If the container needs to remove the servlet, it finalizes the servlet by calling the servlet's destroy method.

**When is the servlet instance created in the life cycle of servlet? What is the importance of configuring a servlet?**

An instance of servlet is created when the servlet is loaded for the first time in the container. Init() method is used to configure this servlet instance. This method is called only once in the life time of a servlet, hence it makes sense to write all those configuration details about a servlet which are required for the whole life of a servlet in this method.

**What is the GenericServlet class?**  
  
GenericServlet is an abstract class that implements the Servlet interface and the ServletConfig interface. In addition to the methods declared in these two interfaces, this class also provides simple versions of the lifecycle methods init and destroy, and implements the log method declared in the ServletContext interface.

**Can we use the constructor, instead of init(), to initialize servlet ?**

Yes. But you will not get the servlet specific things from constructor. The original reason for init() was that ancient versions of Java couldn t dynamically invoke constructors with arguments, so there was no way to give the constructor aServletConfig. That no longer applies, but servlet containers still will only callyour no-arg constructor. So you won t have access to a ServletConfig or ServletContext.

**What are the Scope Objects available ?**

Enables sharing information among collaborating web components via attributes maintained in Scope objects.Attributes are name/object pairs.Attributes maintained in the Scope objects are accessed with getAttribute() & setAttribute()  
  
Scope objects are four types :   
  
***Web context***: Accessible from Web components within a Web context  
***Session***: Accessible from Web components handling a request that belongs to the session  
***Request***: Accessible from Web components handling the request  
***Page***: Accessible from JSP page that creates the object

**What is session tracking and how do you track a user session in servlets ?**

Session tracking is a mechanism that servlets use to maintain state about a series requests from the same user across some period of time. The methods used for session tracking are:   
  
a) ***User Authentication*** : It occurs when a web server restricts access to some of its resources to  only those clients that log in using a recognized username and password   
  
b) ***Hidden form fields***: This fields are added to an HTML form that are not displayed in the client’s  browser. When the form containing the fields is submitted, the fields are sent back to the server  
  
c) ***URL rewriting***: Every URL that the user clicks on is dynamically modified or rewritten to  include extra information. The extra information can be in the form of extra path information,  added parameters or some custom, server-specific URL change.   
  
d) ***Cookies***:  A "cookie" is a small piece of information sent by a web server to store on a web browser so it can later be read back from that browser. This is useful for having the browser remember some specific information. .   
  
e) ***HttpSession***: HttpSession places a limit on the number of sessions that can exist in memory. This limit is  set in the session.maxresidents property   
  
**What is HTTP session in servlets ?**  
  
Session tracking in Servlets is done by using Interface HttpSession. It helps to identify a client throughout many page requests or visiting a website and to store details about that client.   
  
HttpSession session = request.getSession();

* If user already has a session  the existing session is returned.
* If no session exists a new one is created and returned.
* If you want to know if this is a new session:
  + call the Session isNew() method.

**What is Cookies and what is the use of Cookies ?**  
  
A "cookie" is a small piece of information sent by a web server to store on a web browser so it can later be read back from that browser. This is useful for having the browser remember some specific information.  

* to create a **temporary session** where the site in some way "remembers in the short term" what the user was doing or had chosen between web page requests, e.g. remembering who the user is logged in as at the moment, or what they've ordered from an on-line "shopping cart";
* to **remember low-security information more permanently**: for example, to remember a user's search results preferences or who they are logged in as on their social bookmarking site;
* to compile **user statistics**, e.g. for advertising purposes or for improving the functionality of a site.

**What is use of setMaxAge() and getMaxAge() in Cookies ?**  
  
Gets/sets how much time (in seconds) should elapse before the cookie expires. If you don't set this, the cookie will last only for the current session (i.e. until the user quits the browser), and will not be stored on disk.   
  
How do servlets handle multiple simultaneous requests?   
The server has multiple threads that are available to handle requests. When a request comes in, it  is assigned to a thread, which calls a service method (for example: doGet(), doPost( ) and service( ) ) of  the servlet. For this reason, a single servlet object can have its service methods called by many threads at   
once.  
  
**When init() and Distroy() will be called ?**  
  
init() is called whenever the servlet is loaded for the first time into the webserver.it performs certain one time activities which are required during the lifetime of the servlet.It may be some initialisation of variables or a database connection.  
  
destroy() will be called whenever the servlet is removed from the webserver. Various resources which are held by the servlet will be released,database connections which were opened will be closed.Later the servlet is destroyed.  
  
**What is the difference in using request.getRequestDispatcher() and context.getRequestDispatcher() ?**  
  
**request.getRequestDispatcher(path)**: In order to create it we need to give the relative path of the resource  
  
**context.getRequestDispatcher(path)**: In order to create it we need to give the absolute path of the resource.  
  
**What is the difference between ServletContext and ServletConfig ?**  
   
**ServletConfig:**

* One ServletConfig Object is created per servlet
* It can be used to access ServletContext
* Parameters are configured in DD(deployment description)

**ServletContext:**

* One ServletContext will be created per web application.
* Can be used to access web app parameter.
* Can be used to get server Info.

**How do servlets handle multiple simultaneous requests ?**  
  
The server has multiple threads that are available to handle requests. When a request comes in, it is assigned to a thread, which calls a service method (for example: doGet(), doPost( ) and service( ) ) of the servlet. For this reason, a single servlet object can have its service methods called by many threads at once.  
  
**What is the use load on startup (load-on-startup tag) in Servlet ?**

* If the value is not specified or is a Number < 0 then it means "lazy loading"
* What " lazy loading " means is that servlet is NOT loaded by container on startup
* Servlet in this case is loaded on the first client request - so the first client can experience poor performance
* " Eager " loading means that the servlet is initialised on container startup
* If there are two servelts A & B with values 0 & 1 than it means that Servlet A ( having value = 0 ) will be loaded first
* So if there are more than one servlet this element specifies the order of loading - lower integer values ( including zero ) are loaded first
* If you specify this element but do not provide the value - even then the servlet will be the first servlet that gets loaded
* To ensure that servlet follows " lazy loading " - do not provide this entry at all in web.xml file OR provide a negative number

**If a servlet is not properly initialized, what exception may be thrown ?**  
  
During initialization or service of a request, the servlet instance can throw an UnavailableException or a ServletException.   
 **What is URL Rewriting ?**  
  
When the client does not accept cookies, you can use URL rewriting to handle session management. It takes the session ID and sticks it right onto the end of every URL that comes to in to this app. When the user clicks that enhanced link, the request goes to the container with that extra bit at the end, and the container simply strips off extra part of the URL and uses it to find the matching session.  
  
We add the session ID to the end of all URLs in the HTML we send in the respose.  
<a href=http://www.xyz.com/demo.do;jsessionid=0AYU88d7dd89dd8>click me</a>  
  
**What is Servlet chaining ?**  
  
Servlet chaining is a technique in which two or more servlets can cooperate in servicing a single request.In servlet chaining, one servlet’s output is piped to the next servlet’s input. This process continues until the last servlet is reached. Its output is then sent back to the client.  
  
**Is Servlets thread-safe ?**  
  
Servlets are not thread safe. If you want to make it Servlet as Thread safe, you can implement SingleThreadInterface.  
  
There are two different ways of making a servlet thread safe namely  
1. By implementing SingleThreadModel: By implementing a SingleThreadModel it will be possible to create a Thread safe servlet.There can only be one user at a given point of time.  
2. Synchornize the part of sensitive code: We can allow a single user at a given point of time by making that part of the code which is sensitive as synchronized.  
 **What is the difference between Server and Container ?**  
  
A server provides many services to the clients, A server may contain one or more containers such as ejb containers, servlet/jsp container. Here a container holds a set of objects.  
  
Container refers to the component that manages lifecycle of servelet/JSP or EJB. Server refers to the infrastructure which contains those containers.