Introduction to Servlets Quiz

1. What is the difference between a web server and a web container?

Web Server or HTTP Server: A server which is capable of handling HTTP requests, sent by a client and respond back with a HTTP response.

Web Container: is used to manage the components like Servlets, JSP. It is a part of the web server.

2. What is a servlet?

A servlet is server-side java code that can handle http requests and return dynamic content.

Servlets are managed by a servlet engine or container (Web Container or Servlet Container). Each request that comes in results in the spawning of a new thread that runs a servlet (eliminating the cost of creating a new process every time).

3. How do web servers and web containers interact with servlets?

When a request comes to the web server, if the server sees the request is for a servlet, it passes the request data to the servlet container.

The servlet container locates the servlet, creates request and response objects and passes them to the servlet, and returns to the web server the response stream that the servlet produces.

The web server sends the response back to the client browser to be rendered.

4. Who creates request objects?

The servlet container locates the servlet, creates request and response objects and passes them to the servlet.

5. What are the states in the servlet lifecycle?

There are load servlet class, create servlet instance, initialization, service and destroy.

6. Who calls init and when?

The servlet container calls init servlet once and after instantiate servlet.

7. Which of init, service, and doGet should you override?

I should override doGet method.

8. In what sense are servlets multi-threaded?

When container called servlet service for each request, each request runs in a separate thread.

9. What are the implications of this for servlet instance variables?

There is only one instance for a servlet and many user/many threads/many requests may access the instance concurrently. i.e., many threads run on the single instance and it is obvious instance variables are shared by concurrent threads.