Introduction to Servlets Quiz

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

Without making use of some kind of helper application, a web server serves *static content* – files, images, pdfs, videos, exactly as they are on the server machine. Responses cannot be customized based on input data passed in by the client or modified at the time they are delivered.

Servers that support servlets have a helper app as a container. Web container manages servlet that can handle http requests and return dynamic content

**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*

**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 web container.  
Web 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.

**4. Who creates request objects?**

Web container creates HttpServletRequest and HttpServletResponse objects

**5. What are the states in the servlet lifecycle?**

* Load servlet class
* Instantiate servlet
* init() called only once in the servlet's life.  
  Must complete before Container can call service().
* service() (called for each request, each request runs in a separate thread)
* destroy() (called only once)

**6. Who calls init and when?**

Web container calls init only once in the servlet's life before container can call service

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

doGet should be overrided

**8. In what sense are servlets multi-threaded?**

A Java servlet is typically multithreaded. That means, that multiple requests to the same servlet may be executed at the same time.

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

When we say that a program is multithreaded, we are not implying that the program runs two separate instances simultaneously. This means that more than one sequential flow of control runs through the same memory block. When multiple threads execute a single instance of a program and therefore share memory, multiple threads could possibly be attempting to read and write to the same place in memory.

If we have a multithreaded program, we will have multiple threads processing the same instance.