1. What is the difference between a web server and a web container?
   Web server refers to an execution infrastructure that handles HTTP requests and responses; a servlet container refers to a component that handles the lifecycle for servlets (gets the response and provides the response to the web server); an application server refers to a framework for handling Web.
2. What is a servlet?
   It's is a Java program that extends the capabilities of a server. Servlet is a technology used to create web application. Also it is a class that extend the capabilities of the servers and respond to the incoming request. It can respond to any type of requests.
3. How do web servers and web containers interact with servlets?
   When user sends request to the web server then web server delivers the client request to the web container if the dynamic response. Web container has the servlet deployed and calls the service method of the servlet in a new thread. The container send the request and response to the servlet.
4. Who creates request objects?
   Web container.
5. What are the states in the servlet lifecycle?
   There are three states of a servlet: new, ready and end. The servlet is in new state if servlet instance is created. After invoking the init() method, Servlet comes in the ready state. In the ready state, servlet performs all the tasks. When the web container invokes the destroy() method, it shifts to the end state.
6. Who calls init and when?
   A servlet's init method is called by the server after the server constructs the servlet's instance.
7. Which of init, service, and doGet should you override?
   Init and doGet overridden when it needs for initialization.
8. In what sense are servlets multi-threaded?
   In order to accomplish client's request web container creates multi threads to a single servlet.
9. What are the implications of this for servlet instance variables?
   Instance variables are shared to all requests. Servlet's instance variables are not thread-safe.