1. How J2EE differ from EJBs.
    - J2EE is a platform for developing distributed enterprise applications. J2EE is a set of specifications needed to create applications in the enterprise environment with Java technology. J2EE is a set of specifications covering a very large range. Representative components include Servlets, JSPs, EJBs, RMI, JNDI, and JDBC.
    - EJBs is one of the components of J2EE which is a architecture for distributed computing that provide 3-Tier. With EJB, you can create Java classes that are distributed, transactional, and secure with little effort.

2. Difference between Web Server and Application Server.
    - **Web server** can be either a computer program or a computer running a program that is responsible for accepting HTTP requests from clients, serving back HTTP responses along with optional data contents, which usually are web pages such as HTML documents and linked objects on it.
    - **Application server** is the kind of software engine that will deliver various applications to another device. It is the kind of computer found in an office or university network that allows everyone in the network to run software of the same machine.

3. What is serialization in JAVA?
    - Serialization is the conversion of the state of an object into a byte stream; deserialization does the opposite. Stated differently, serialization is the conversion of a Java object into a static stream (sequence) of bytes which can then be saved to a database or transferred over a network.
    - The serialization process is instance-independent, i.e. objects can be serialized on one platform and deserialized on another. Classes that are eligible for serialization need to implement a special marker interface Serializable.

4. Difference between GET and POST request.
    - GET requests a representation of the specified resource. Note that GET should not be used for operations that cause side-effects, such as using it for taking actions in web applications. One reason for this is that GET may be used arbitrarily by robots or crawlers, which should not need to consider the side effects that a request should cause.
    - POST submits data to be processed (e.g., from an HTML form) to the identified resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both.