1. **Read** [**Your First Cup: An Introduction to the Java EE Platform**](https://docs.oracle.com/javaee/7/firstcup/index.html) **— Focus on Chapter 2, “Understanding Java Platform, Enterprise Edition”**
   1. **Explain the *tiers* discussed in the article. On which does J2EE focus? On which to the Oracle database focus?**In a multi-tiered application, the functionality of the application is separated into isolated functional areas, called tiers. Typically, multi-tiered applications have a client tier, a middle tier, and a data tier (often called the enterprise information systems tier). The client tier consists of a client program that makes requests to the middle tier. The middle tier is divided into a web tier and a business tier, which handle client requests and process application data, storing it in a permanent datastore in the data tier.
   2. **Explain the nature of a J2EE *container*. What sort of containers are there?**J2EE/Java EE applications aren't self contained. In order to be executed, they need to be deployed in a container. In other words, the container provides an execution environment on top of the JVM.  
       Also, applications rely on several APIs like JPA, EJB, servlet, JMS, JNDI, etc. The role of the EE compliant container is to provide a standard implementation of all or some of these APIs. This means you can theoretically run your application on top of any container as long as it relies on standard APIs.  
       From a technical perspective, a container is just another Java SE application with a main() method. EE applications on the other hand are a collection of services/beans/servlets/etc. The container finds these components and runs them, providing API implementations, monitoring, scalability, reliability and so on.
2. **Read** [**The Java EE Tutorial**](https://docs.oracle.com/javaee/7/tutorial/index.html) **— This tutorial provides extensive material on J2EE; focus on the sections listed here.**
   1. **Sections 1.7–1.8, “Java EE 7 APIs” — Explain the purpose of these technologies (only): *JPA*, *JTA*, *JAX-RS*\*, *JDBC*, *JAXB*.**JPA: Java Persistence API  
      JTA: Java Transaction API  
      JAX-RS: The Java API for RESTful Web Services  
      JDBC: The Java Database Connectivity (JDBC) API lets you invoke SQL commands from Java programming language methods  
      JAXB: The Java Architecture for XML Binding (JAXB) provides a convenient way to bind an XML schema to a representation in Java language programs.
   2. **Section 5, “Packaging Applications” — Compare and contrast *JARs*, *WARs* & *EARs*.**JAR: EJB modules which contain enterprise java beans (class files) and EJB deployment descriptor are packed as JAR files with .jar extenstion  
      WAR: Web modules which contain Servlet class files, JSP Files, supporting files, GIF and HTML files are packaged as JAR file with .war (web archive) extension  
      EAR: All above files (.jar and .war) are packaged as JAR file with .ear (enterprise archive) extension and deployed into Application Server.
3. **Review these tutorials, on which the lab is based:** [**Developing a Java EE Application**](https://www.jetbrains.com/help/idea/2016.3/developing-a-java-ee-application.html) **&** [**Creating and Running Your First RESTful Web Service**](https://www.jetbrains.com/help/idea/2016.3/creating-and-running-your-first-restful-web-service-on-glassfish-application-server.html)**.**
   1. **Identify the IDE we’ll use for J2EE development.**
   2. **Identify the roles of the following technologies in the J2EE application: J2SE’s *JDK*; *GlassFish*.**
   3. **Explain the nature of a RESTful webservice.\***