

## Assignment - 1

Q1 - What is Java beans, advantages and properties? Explain in detail.

A Java Bean is a Java class that should follow the following conventions.

- \* It should have no-arg constructor.

- \* It should be serializable.

- \* It may have a no. of properties which can be read or written.

- \* It may have a no. of "getter" and "setter" methods for the properties.

### Java Beans Properties

A Java Bean property is a named characteristic that can be accessed by the user of the object. JavaBeans properties are accessed through two methods in the Java Bean's implementation class:

1. get Property Name();

for ex, if the property name is FirstName, the method name would be getFirstName() to read that property. This method is called the accessor.

2. set Property Name();

for ex, if the property name is FirstName, the method name would be setFirstName() to write that property. This method is called the mutator.

### Advantages of Java Bean

The following are the advantages of Java Beans:

- \* The javabean properties and method can be exposed to

to another application.

\* It provides an easiest to reuse the software components.

### Simple Example of JavaBean class

```
package mypack;
public class Employee implements java.io.Serializable {
    private int id;
    private String name;
    public Employee() {}
    public void setId (int id) {
        this.id = id;
    }
    public int getId () {
        return id;
    }
    public int getId () {
        return id;
    }
    public void setName (String name) {
        this.name = name;
    }
    public String getName () {
        return name;
    }
}
```

Q2: Write the short notes on

i) BDK - It stands for Bean Developer Kit, available from the java soft site, is a simple example of a tool that enables you to create, configure and connect a set of Beans. There is also a set of sample Beans with their source code.

(ii) EJB - It stands for Enterprise JavaBeans. EJB was the robust kind of framework techniques which is made by sun micro system to make easy in the developing process of applications. It is a server side component that encapsulates business logic of an application. An EJB is a web container provides a runtime environment.

(iii) Java Beans API - Java Beans API provides the set of the interfaces and classes which are defined in JavaBeans. The same list of APIs that can be used while developing programs:-

Bean → This class is useful in obtaining information about the bean.

Event Handler → For supporting event listener creation, this class is useful.

Expression → Call and method are bound together to return the result.

Method Description → Method of beans can be described.

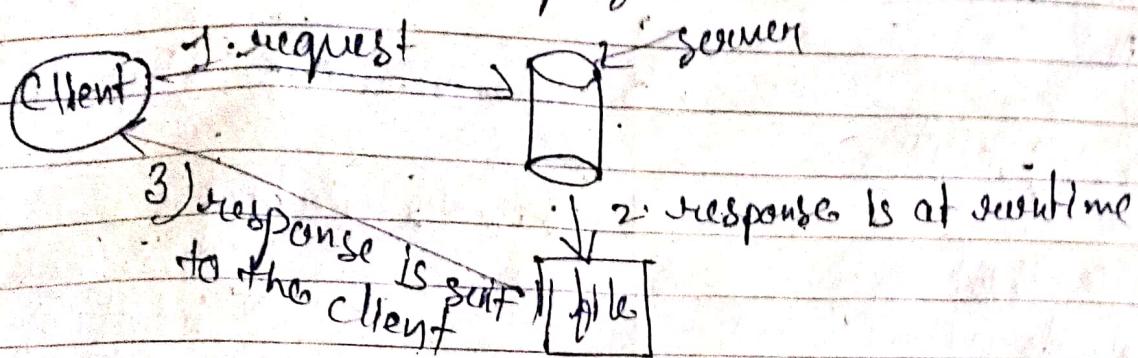
#### (iv) Servlet

\* Servlet technology is used to create a web organization. servlet technology is robust and scalable because of java language, there are many interfaces and classes in the servlet API such as servlet, GenericServlet, HTTPServlet, ServletRequest, ServletResponse, etc.

\* Servlet is an interface that must be implemented before creating any servlet

\* Servlet is a class that extends the capabilities of the

servers and responds to the incoming request. It is a web component that is deployed on the server.



Q3 What is the architecture of EJB? Explain in detail.

### Architecture of EJB

The EJB architecture consists of three main components: enterprise beans, the EJB container, and the Java application server. EJBs run inside an EJB container and the EJB container runs inside a Java application server.

There are two types of EJB. -- session:- beans and message driven beans:

- Session beans → Session beans are invoked by the client and make enterprise functionality such as transactions and resource management available to the client programmatically.
- Message-driven beans → It also encapsulates and deliver enterprise functionality, but they are asynchronous and event driven. Message driven beans listen and respond to events and cannot be invoked by the client.  
Once used to provide persistence in the EJB system, entity beans have been supported by the Java Persistence API.

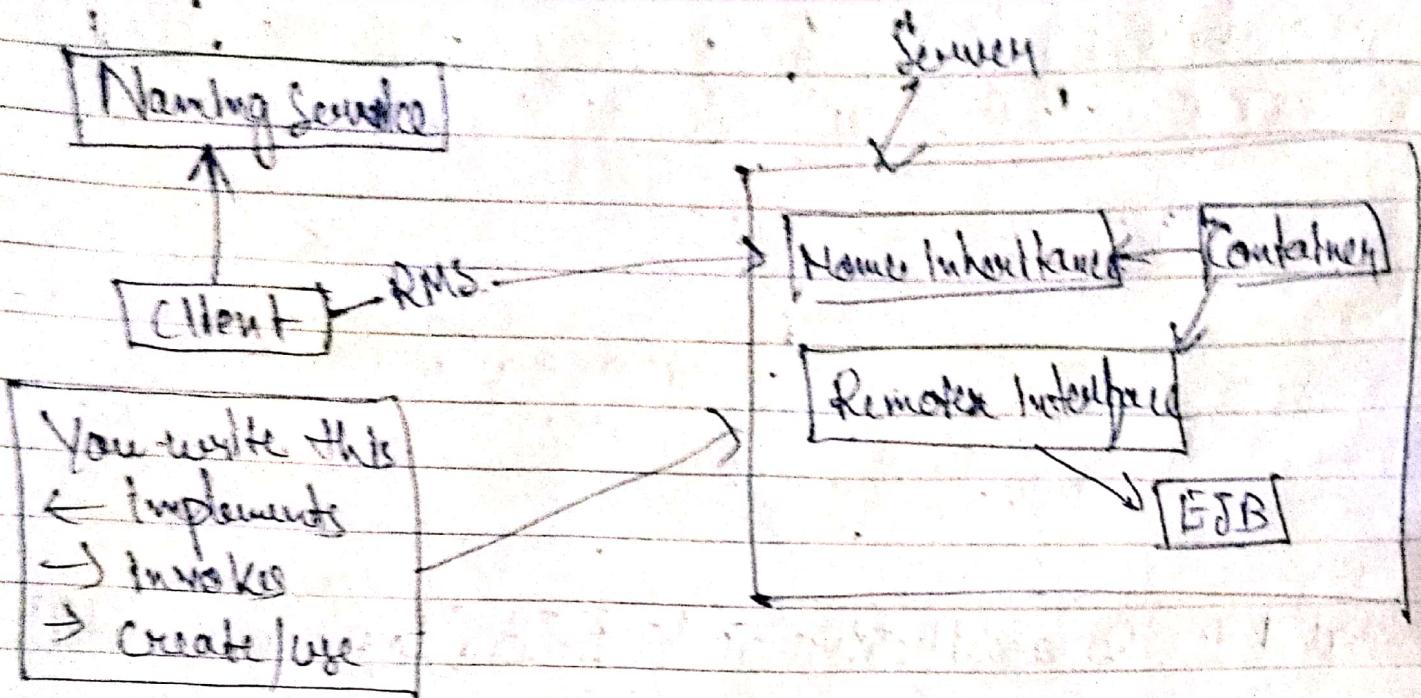


Fig:- EJB Architecture

Q4 Explain life cycle of Servlet, JSDK and deploying a servlet.

Servlet Life Cycle → This web container has the life cycle of a servlet instance. There are five steps of the life cycle of the servlet:

Step 1 → We load the servlet class and the class loader is responsible to load a servlet class.

Step 2 → The web container creates the instance of servlet after loading the servlet class.

Step 3 → The web container calls the service init method only once after creating the servlet instance. The init method is used to initialize the servlet.

Step 4 → The web container calls the service method each time when request comes the servlet is executed.

Step 5 → The web container calls the destroy method before removing the servlet instance from the services.

### JSDK

- \* It stands for Java Servlet Development Kit.
- \* The JSDK allows developer to build servlet or applet that run on server to Servlet API that consistent across Web server. The current kit includes the Java Servlet & a servlet runner for testing servlets.
- \* JSDK also contains a web server and servlet engine to test your creations.

### Deploying Servlet

The deployment of a servlet is an XML file from which container gets the information about the servlet to be invoked.

```
<Web-app> <Servlet>
  <Servlet-name> ----- </Servlet-name>
  <Servlet-class> ----- </Servlet-class>
  <Servlet-mapping>
    <Servlet-name> </Servlet-name>
    <url-pattern> . . . </url-pattern>
  </Servlet-mapping>
</Web-app>
```

Q5. Explain Servlet API & Servlet package with HttpServlet package.

Servlet API → Servlet API to create servlets :

There are two packages that you must remember while using API that is:

- javax.servlet
- javax.servlet.http

Some important class and interfaces of javax.servlet

Interfaces	Classes
Servlet	ServletInputStream
ServletContext	ServletOutputStream
ServletConfig	ServletRequestWrapper
ServletRequest	ServletRequestEvent
ServletResponse	ServletContextEvent
Filter	ServletException

Some important classes and interfaces of javax.servlet.http

Classes	Interfaces
HttpServlet	HttpServletRequest
HttpServletResponse	HttpSessionAttributeListener
HttpSession	HttpSessionListener
Cookie	HttpSessionEvent

Servlet package with http

public abstract class GenericServlet extends java.lang.Object  
implements Servlet, ServletConfig, java.io.Serializable

- GenericServlet defines a generic, protocol implements servlet
- GenericServlet uses service() method to handle request
- GenericServlet gives a blue print and makes writing servlet easier.