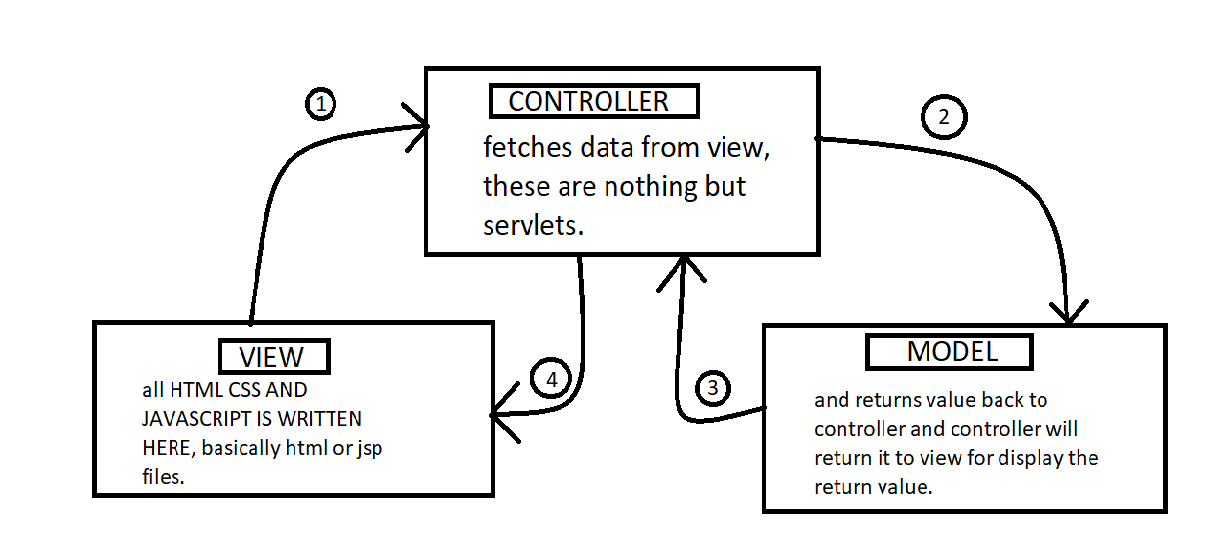
Advance Java Interview Questions

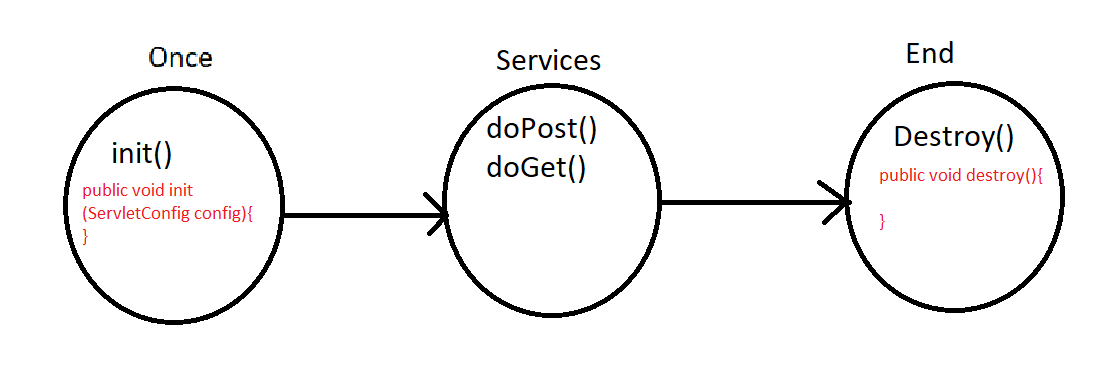
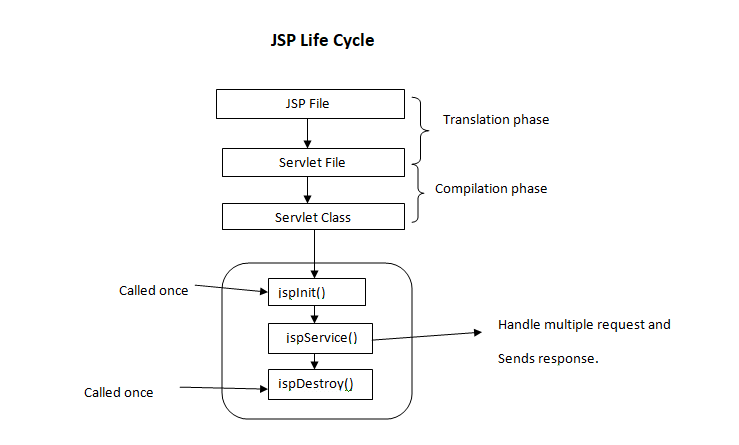
1. **What is MVC Architecture?**

* 
* VIEW:
  + Here we write all the front-end work such as CSS, HTML, JS, either in html file or in jsp file.
  + Everything which we see in front-end is Written here.
* CONTROLLER:
  + It is a mediator between view and Model.
  + It basically takes all the input from the view and sends it to model & once model completes its work and if it returns value to controller it sends that back to view.
  + Servlets are called as controllers.
* MODEL:
  + Here we write all the business logic such as connecting to database, all crud operations and closing of connection etc.
  + We always use RTP i.e., here we create interface and write all methods which we want to implement and override in other class and we do upcasting in servlets.

1. **What are servlets?**

* Servlet Technology is technically used to create web applications.
* Servlet is an API which provides many interfaces and classes such as http servlet, servlet request and servlet response etc.
* Servlet is an class which extends the capability of server and will respond to any type of incoming requests.
* It is a subclass of HTTP servlets, it is used to perform Back-end coding of the application.

1. **Explain Servlet life Cycle?**

* 
* For the first time when we start the server tomcat init() method will run, post that any number of times we can call doGet() and doPost() method, finally when the destroy method is called servlet life cycle comes to an end.
* Note:
  + Code to connect to DB we write itr in init(){} method.
  + Code to close Connection to DB we write it in destroy(){} method.
  + All other code of business logic is written inside doPost() and doGet() methods.
* What is Servlet Execution Engine?
* **It is also called as sevlet container.**
* **Servlet container**, also known as **Servlet engine** is an integrated set of objects that provide a run time environment for Java Servlet components.
* In simple words, it is a system that manages Java Servlet components on top of the Web server to handle the Web client requests.
* What is request Dispatcher?
* The RequestDispatcher interface provides the facility of dispatching the request to another resource it may be html, servlet or jsp. This interface can also be used to include the content of another resource also.
* There are 2 methods in request Dispatcher they are:
  + public void forward(ServletRequest request,ServletResponse response)throws ServletException,java.io.IOException:Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.
  + public void include(ServletRequest request,ServletResponse response)throws ServletException,java.io.IOException:Includes the content of a resource (servlet, JSP page, or HTML file) in the response.
* Explain Inter servlet Communication?
* ISC: When one servlet is calling another servlet using request dispacther is called as inter servlet communication.
  + Request.setAttribute & .get Attribute will work only when request dispatcher concept is used.
  + Call back methods🡪 main method, doGet method, doPost method etc. that’s the reason why servlet wont have any main methods as they consists of doGet and doPost methods.
  + .setAttribute are just used for short time that is temporary memory, whereas to store values for longer time we use session variable concepts.
  + If request dispatcher is called from post method of first servlet then it will call 2nd servlet post method only.
  + Session should be used in ecommerce app where logic of add cart etc where items should be stored for longer time.
  + As long as history/cookies are not deleted, session will not be deleted.
  + Session: once the data is stored in session variable then that data can be remembered across the application.
* What is the purpose of forward () and include () methods in request Dispatcher?
* public void forward(ServletRequest request,ServletResponse response)throws ServletException,java.io.IOException:Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.
* public void include(ServletRequest request,ServletResponse response)throws ServletException,java.io.IOException:Includes the content of a resource (servlet, JSP page, or HTML file) in the response.
* Explain JSP life Cycle?
* 
* A Java Server Page life cycle is defined as the process that started with its creation which later translated to a servlet and afterward servlet lifecycle comes into play. This is how the process goes on until its destruction.
* Write JDBC code to connect With Database?
* Refer Notes.
* What are cookies?
* Cookies are small blocks of [data](https://en.wikipedia.org/wiki/Data_(computing)) created by a [web server](https://en.wikipedia.org/wiki/Web_server) while a [user](https://en.wikipedia.org/wiki/User_(computing)) is [browsing](https://en.wikipedia.org/wiki/Browsing) a [website](https://en.wikipedia.org/wiki/Website) and placed on the user's computer or other device by the user’s [web browser](https://en.wikipedia.org/wiki/Web_browser). Cookies are placed on the device used to access a website, and more than one cookie may be placed on a user’s device during a session.
* Cookies serve useful and sometimes essential functions on the [web](https://en.wikipedia.org/wiki/World_Wide_Web). They enable web servers to store [stateful](https://en.wikipedia.org/wiki/Program_state) information (such as items added in the shopping cart in an [online store](https://en.wikipedia.org/wiki/Online_shopping)) on the user’s device or to track the user's browsing activity (including clicking particular buttons, [logging in](https://en.wikipedia.org/wiki/Access_control), or recording which [pages were visited in the past](https://en.wikipedia.org/wiki/Web_browsing_history)).
* What are session Variables?
* A session variable is a special type of variable whose value is maintained across subsequent web pages. With session variables, user-specific data can be preserved from page to page delivering customized content as the user interacts with the web application.
* For example while creating login and logout webapp we use session variables to store data to check if session is active or not if active all feauters work like crud operations. but if session is inactive then if we try to use any feauter it will lead to login page.
* Difference between Application Server and Web Server?
* Application-servlets:
* If you want to build Dynamic web-Applications using technologies such as servlets, jsp, Hibernate, Spring, Spring-Boot etc. in this case to run these technologies we use Application-Servlets such as Tomcat, J-Boss, Glass-Fish.
* Web-Servlets:
* To build static App we use Technologies like HTML, CSS, JS & to run this we use Webservers like IIS, Apache.
* Difference between doGet() and doPost() methods?
* Differences:

|  |  |
| --- | --- |
| GET | POST |
|  |  |
| * Should be used when we want to get data from database. | * Should be used when we submit data using forms to the database. |
| * Data is Exposed in the URL. | * Data is not Exposed in the URL. |
| * URL data is exposed in browser history | * URL data is not exposed in browser history |
| * When you refresh page you will not get security alert!. | * When you refresh page you will get security alert!. |

* What code we can write in init () method?
* Something which needs to be run before evrything we write here, such as connecting to Databases etc.
* What are JSP’s?
* It stands for Java Server Pages.
* It is a server side technology.
* It is used for creating web application.
* It is used to create dynamic web content.
* In this JSP tags are used to insert JAVA code into HTML pages.
* It is an advanced version of Servlet Technology.
* It is a Web based technology helps us to create dynamic and platform independent web pages.
* In this, Java code can be inserted in HTML/ XML pages or both.
* JSP is first converted into servlet by JSP container before processing the client’s request.
* We can write java code here using JSTL tags.
* We can write code to connect to database directly from here only.
* What are JSTL tags?

1. Script-Let Tag<% %>:

* Advantages:
  + 1)Implicit objects can be created.
  + 2)request
  + 3)response
  + 4)session
  + 5)out Object etc.
* Disadvantages:
  + 1)cannot create variable with access specifiers.
  + 2)Cannot build any methods in it.

1. Declaration Tag<%! %>:

* Advantages:
  + 1)can create variable with access specifiers.
  + 2)Can build any methods in it.
* Disadvantages:
  + 1)Implicit objects cannot be created.
  + 2)request cannot be created.
  + 3)response cannot be created.
  + 4)session cannot be created.
  + 5)out Object cannot be created etc.

1. Expression Tag<%= %>:

* It is generally like out.println() implicit jsp objects.
* We can write only one line of code in it.
* We don’t have to use any semicolon inside.

1. Directive Tag<%@ %>:

* 2 Types:
  + Page Directive Tag<%@page import=”” %>:
    - Basically, to write all imports like, file class, package imports, scanner class imports all these should be written in page directive tags.
  + Include Directive Tag<%@ include file=” any .jsp or .html or any txt files or etc.” %>:
    - Basically, it is used to import or include other or same extension files into jsp.
* JSP implicit Objects?
* Out🡪 out.println();
* Request🡪HttpServletRequest.
* Responnse🡪 HttpServletResponse.
* Session🡪HttpSession etc.