Database

Table Login : in Database ie MySQL

Core Java

bean class -→ generally link table which contains property setter, getter and constructor.

dao class → it contains pure database logic code. Like insert, delete, update and retrieve etc. using JDBC.

service class → it contains pure business logic. Those logic can execute before calling dao method or after dao method or without conditions.

resource class → provide resource details like database connection

utility class → in this class we are written menu driven options.

Interact with console in

Core AWT (Abstract Window toolkit) / Swing / JavaFX which help to create GUI (Graphical User interface).

Then we can interact with browser

Then we can interact with angular

main class → entry of the application in core java

Generally dao and service layer not responsible to interact with input as well output device.

In DAO layer, service layer, resource layer don't use System.out.println() to display the message. Because System.out.println always display the message on console not on browser.

With Core Java with help of AWS/Swing/JavaFX we can create GUI application

JDBC to connect the database

MySQL and Oracle

Using these concept we can create standalone or desktop application.

The application running on one machine is known as standalone or desktop application.

It is like a single user. To run this application in our machine all required software must be present.

Web Application

In web application we need to browser and internet.

https://www.google.com

req(http/https)---------->

Client Server

<-----res(http/https)----------

Html

Css

JavaScript

Using angular we can develop only frontend side technologies

Not backend technologies.

Java

Core Java or JSE

Java Standard edition

JEE : Java Enterprise edition

Python

Asp.net

Php

Node JS

JEE : Java Enterprise Edition :

Servlet, JSP (Java Server Pages), EJB (Enterprise Java Bean).

EJB replace by Spring Framework (Course 3)

To run Servlet, JSP or EJB we need server. Servlet , JSP and EJB are server side technologies. Because in Servlet, JSP and EJB no main method.

We need to create application using servlet, jsp or ejb and deploy in server.

Once we deploy the application server only responsible to load the class, create the object, call life cycle method and destroy object.

Application point of view.

Server mainly divided into two types.

1. Web server : Apache tomcat
2. Application server : Web Logic, JBoss, WebSphere etc

Web Server is small server. Web server we use in development environment.

Application server are very big server. Application server we use in production environment.

http://localhost:42000

http://127.0.0.1:42000 if we run the application in local machine we can access that application using above url.

http://192.165.12.78:42000

or

http://www.akash.com

if we deploy the application in server ie web server or application server then we can access that application using above url

aws or azure or private cloud or public cloud help us to deploy the application in web server or application server.

Tomcat web server

Application server to deploy enterprise level application.

Web server : tomcat

Application server : web logic or web sphere or jboss etc

JEE : Java Enterprise Edition which help to create web application using java technologies.

Servlet : Servlet is normal java program which help to create dynamic web page on server side.

import javax.servlet.*;

servlet : servlet is a package part of javax (extensible)

import javax.servlet.Servlet;

Servlet : Servlet is an interface part of servlet package. Which contains 5 methods.

init	this method call only once
service	this method call again and again to give service to client
destroy	this method called at last to destroy the object.

life cycle methods. Those methods call automatically whenever client send the request to web application. If application develop using java ie servlet. Inside server ie tomcat contains web container. Web container is responsible to call those methods whenever any request receive from the client.

getServletInfo	
getServletConfig	abstract method (method without body)

1st option

class MyServlet implements Servlet {

we need to override all five methods mandatory.

}

2nd : GenericServlet : it is a type of abstract class internally implements Servlet interface and provided body for all abstract method except service method.

class MyServlet extends GenericServlet {

we need to provide body for only service method.

}

3rd option : HttpServlet : it is a type of abstract class which internally extends GenericServlet and provided body for service method as well as it provide some extra method in the form of doXXX like doGet, doPost, doPut, doDelete etc.

class MyServlet extends HttpSerlvet {

doGet or doPost or service

}

FTP

SMTP

POP

Etc

Web.xml file dd file (deployment descriptor file).

Once we run the application or deploy on server

Then using browser we can access this application

http://localhost:8080/ProjectName/UrlPattern