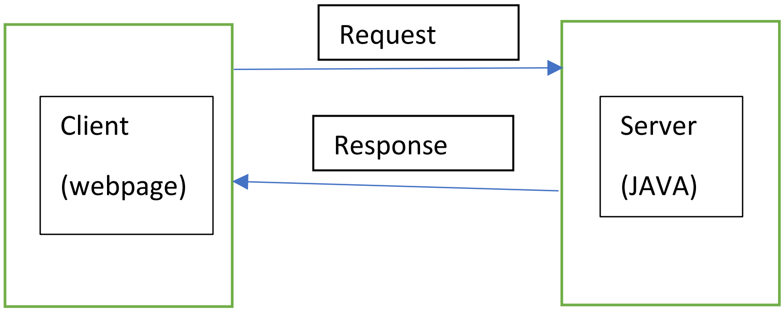
Advanced Java

1. **What is J2EE?**
    - J2EE stands for java enterprise edition.
    - J2EE is a extension for core java i.e. jse
    - JEE represents group of different technologies. e.g. JDBC, servlet, jsp
    - J2EE can be used to develop enterprise and web application.
    - J2EE is an implementation of core java.

2. **Explain the features of J2EE?**
    - J2EE is a platform independent because it is an implementation of JAVA.
    - Using J2EE we can develop portable web application.
    - It provides large number of libraries developed enterprise application.

Web application
Enterprise Application



Website- Static Pages (No only frontend, no Backend)

Web Application- Static & Dynamic

Web Portal-

3. Diff betn web application & Enterprise Application
    1. Web Application
        1. Web application is a simple client and server application.
        2. Client application sends the request to server application and server application provides the appropriate response to the client application.
        3. In case of web applications, money transactions are not involved.
    2. Enterprise Applications
        1. These are business applications.
        2. Enterprise applications are also known as Money making applications.
        3. In case of enterprise applications money transactions can be done through payment vendors and payment gateways.

#Conclusion:
J2EE is capable to develop web as well as enterprise applications.

Syllabus:
Section 1- (J2EE)
JDBC

1. Database Operations.
2. Database Transaction.
3. Database Applications.
4. DAO/DTO Design Pattern.

Section 2 (J2EE)

Servlet

1.Web Transactions.

2.Web Server Configurations.

3. Static and Dynamic Web Applications.

4. Sessions and Cookies.

Section 3- (JSP)

1.JSP Elements

2.JSP Life Cycle

3.MVC Design Pattern

Framework

1. Hibernate

1.Hibernate Architecture.

2. Hibernate CRUD

3. Hibernate HQL and HCQL

4. Hibernate Advanced mapping

5. Hibernate Inheritance

2. Spring

1. Spring IoC
2. Spring DI
3. Spring Bean Factory

4. Spring MVC
5. Spring Boot
6. Spring Security
7. Spring Data JPA

Language: Installable Software

Technology: Collections of multiple libraries

Framework: Collection of different Technologies

Explain the difference betn Language Vs Technologies vs Framework?

Language

1. Language is a collection of basic libraries or packages.
2. By using language, we can develop basic applications e.g. JAVA

Technology

1. It is a collection of multiple libraries or packages
2. We can develop mid-level applications with the help of technologies e.g. J2EE

Framework

1. It is a collection of different technologies.
2. It provides advanced libraries and packages to develop high scale applications.

These are the pre requisite

1. Core Java
   Inheritance, Overriding, Overloading, Upcasting, Downcasting, Interface, Abstraction, Exception Handling, Collection Framework
2. SQL
   SQL Statements, SQL operators, SQL Clauses, SQL Joins
3. WT
   HTML, CSS, Javascript

(10/02/23)>

Q. What is Properties file?

-> 1. It is a special type of file which stores the data in the form of key -value pair.

2. This file is mainly used to store configuration details. E.g. database username-pass, server port number

3. Properties file is a platform independent which means we can access the data from this file by using any operating system involved.

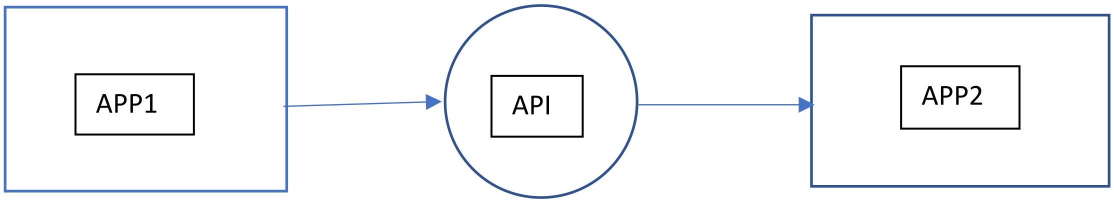
4. To perform the operations on properties file, java language has provided properties class which is declared inside the java.util package.

API

API-

To provide communication link betwn two applications, in other words one application can communicate with another application through API.

API acts as a middleware between two applications.



>What is the need of JDBC?

1. The Data available in terms of variables is a temporary data which means this data will not be available for long time.

2. To store the data for long time we have two options

    1. File Storage

    2. Database storage

3. File storage can be used to store un structured data whereas database storage can be used to store structured data.

> What is JDBC?

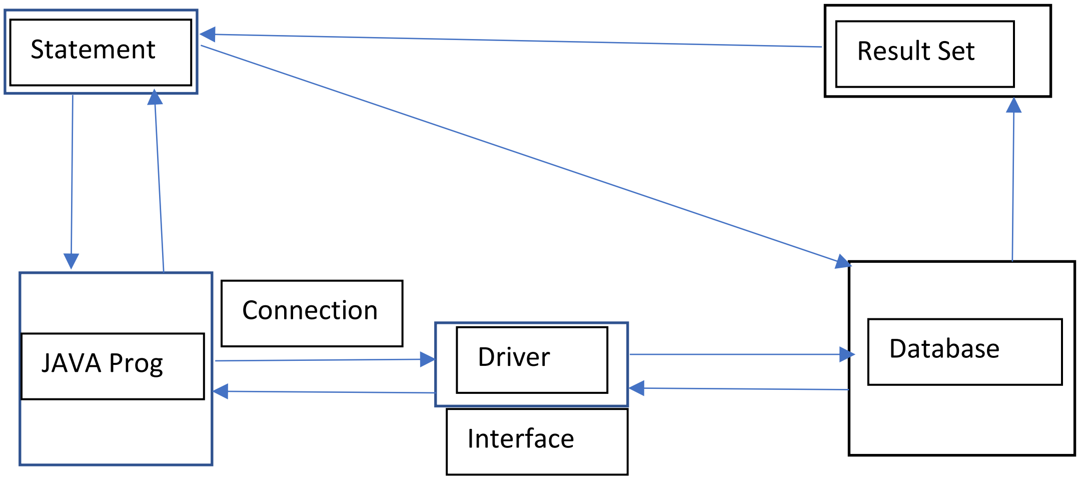
1. It is a java based technology which can be used to connect java applications with the database applications.

2. In other words, JDBC acts as an API between java and database.

> Explain the working of JDBC API?

`

s



To develop s simple JDBC application we have to use four interfaces, provided by java language.

1. Driver
   - It acts as a translator between java and database application.
   - Driver is responsible to convert java instructions into database instructions and database instructions into java instructions.
2. Connection
   - It provides communication link between java and database applications.
   - In other words, connection acts as a network provider
3. Statement

   > This interface acts as an input container.

   > Developer can perform various database operations by using statement interface.

4. Result Set

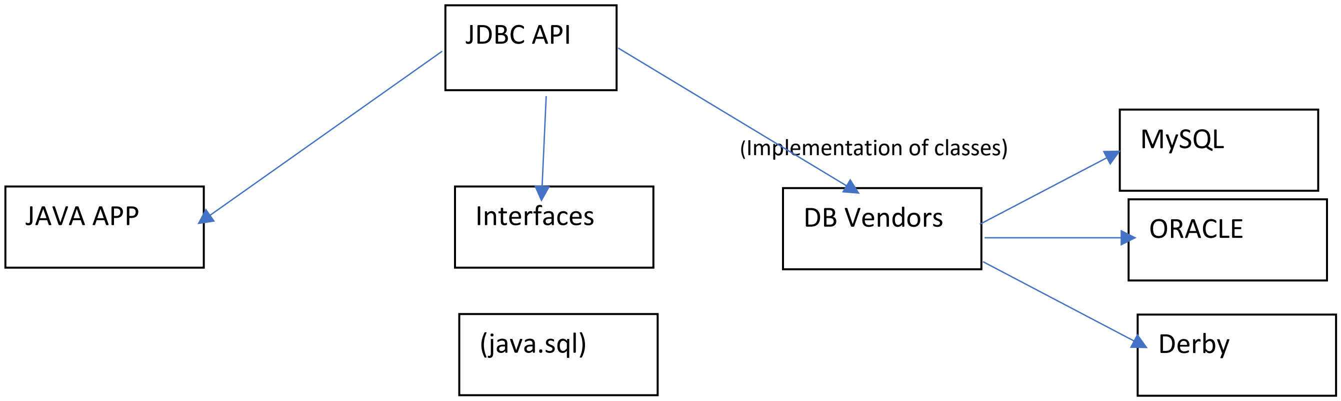
   > It acts as an output container.

   > After execution of SQL query output will be stored in a result.

What is JDBC API?

- JDBC API acts as a container for different interfaces and classes.
- JAVA language has provided interfaces required to communicate with database.
- Database vendor is responsible to provide an implementation for all the interfaces declared by java language.
- Developer is not responsible to create implementation classes for the interfaces.
- All interfaces and classes of JDBC API are present inside the following package

1. Java.sql- This package contains the basic classes and interfaces required to develop simple JDBC applications
2. Javax.sql- This package acts as a container for advanced classes and interfaces required to develop high scale database applications.



Dev Tools

1. MySQL Server
2. MySQL client
3. JDK (1.8)
4. IntelliJEIDEA
5. MAVEN

PART TO BE ADDED

What is MAVEN?

MAEN is project management tool.

It can be used to manage java based projects.

MAVEN is mainly used to build management and dependencies.

Why we need MAVEN?

We can develop an application by creating simple java project.

Every project is having some dependencies so we have to add those dependencies into the project.

If we have a simple java project this process needs to be done explicitly which means developer is responsible for the installation and configuration of all the dependencies.

It is very difficult to manage high scale projects w/o using project management tools.

Because of all these reasons we need a MAVEN project.

Explain the working of MAVEN project?

Every MAVEN project is having pom.xml file, which is a configuration file.

Developer can declare all requirements inside the pom.xml file

Based on the requirements provided by the developer MAVEN will download all the dependencies from central repository.

To access central repository internet connection is mandatory hence it is known as online repository.

After downloading the dependencies from central repository automatically it gets stored into local repository which is known as offline repository.

When we are creating new MAVEN project by default it will search all the dependencies into local repository.

Steps to develop JDBC Application

1. Load & register the driver.
2. Establish the connection
3. Create statement platform
4. Prepare and execute SQL query
5. Process the result set
6. Close the costly resources

Steps:

Q. Explain the process for loading and registering the driver

> 1. To develop simple JDBC application it's mandatory to load and register database driver.

2. Every database software is having different driver.

3. It means if we change the database then we have to change its driver.

4. To load and register the driver we can use Class.forName() method

5. forName() method accepts fully qualified class name as an argument.

6. This step is optional from JDK 1.8.

Q. Explain the process to establish the connection with the database vendor?

> 1. After loading the driver, we have to provide connection between java and database application.

2. To establish the connection with database we can use DriverManager.getConnection() method.

3. DriverManger is an internal class declared inside the java.sql package.

4. getConnection () method accepts 3 arguments i.e. url, username, password.

Q. What is the use of Statement platform?

> 1. Statement platform is required for the execution of sql queries.

2. In java, we can create statement platform by using 3 different ways,

   1. Statement

   2. Prepared Statement

   3. Callable Statement

3. In case of Statement interface we have to call connection.createStatement() method to build statement plateform.

Q. How to prepare and execute SQL query?

> 1.We can create a SQL query by using string variable.

2. To execute SQL query we have 3 options

   1. executeUpdate() method- DML statements only- returns int

   2. executeQuery() method- DQL statements only- returns the resultset

   3. execute() method- DML and DQL- return the boolean

1. executeUpdate()

This method is used for the execution of DML statements (i.e. insert, update, delete).

This method always returns int type of value.

2.executeQuery()

This method is responsible for the execution of DQL statements- i.e. select.

This method always returns the ref of result set.

3.execute()

This method is capable to execute DML as well as DQL statement.

This method always returns Boolean type of value. For DQL statements output will be true and it will be false for DML statements.

Q. Explain the working of the ResultSet?

-> 1. ResultSet is an interface declared inside in java.sql package.

2. At the time performing DQL operations response data will be stored into result set object

3. If java application sends the request to database server then database sever is responsible to perform following activities

1. Check the syntax and rules of the SQL query

2. Execute the SQL query

3. Store the data into result set object.

4. Developer is responsible to process the data present inside the ResultSet.

5. Following methods are required to process the data.

1. next() – This method is responsible to check whether the database table contains next record or not

This method always returns Boolean type of value.

If next record is available then output will be true else it will be false.

2.getters()- getters() method are used to fetch the data from specific column

Based on the datatype we have to change signature of the getter method.

e.g. if column returns the int value then we have to use getInt() method. If column returns the string value then we have to use getString()

Following is the syntax to process the data from resultset

```
While(rs.next())
{
getInt();
getString();
getDouble();
}
```

Problem Statement:

CMD

1.Enter a username

->root

2.Enter password

->123

Login Successful

.java

1.Fetch the values from database

2.Compare user defined values

3.Display Message

DATABASE

id	Username	Password
A1	root	123
A2	admin	456

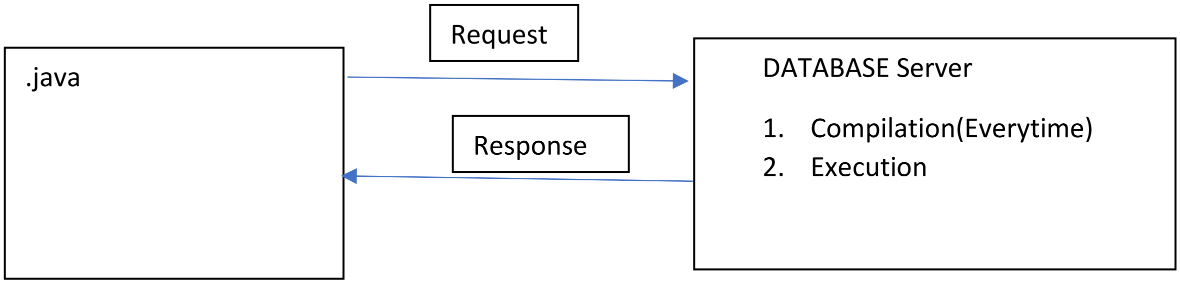
Problem:

1. Login-
   If (Successful)-> 1. Add Product
   2.Display Product
   Else-> login fail
2. Sign Up-> New UserName and Password-> Register

Q. What is the difference between Statement and PreparedStatement interface?

-> 1. Statement as well as PreparedStatement are used to create statement platform in order to execute SQL query.

2.In case of Statement interface query will be compiled and executed for every request sent by java application.



.java

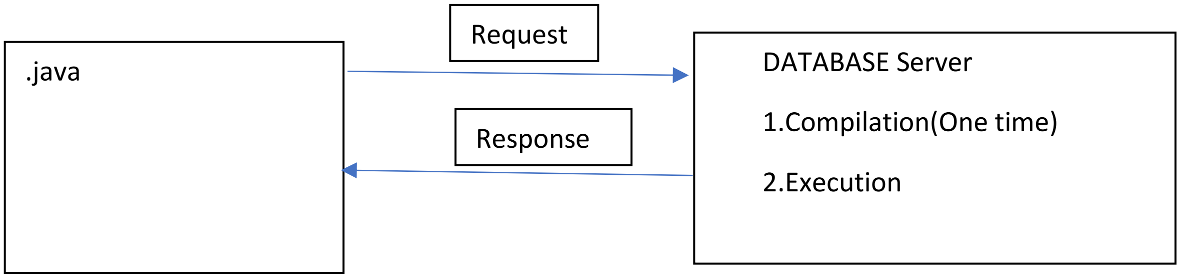
Request

Response

DATABASE Server

1. Compilation(Everytime)
2. Execution

3.In case of PreparedStatement query will be compiled only once but executed multiple times.



4.If programmer wants to perform the insert operation 100 times then Statement interface will compile and execute the request 100 times but in case of PreparedStatement it will compile it only one time and execute it 100 times.

Conclusion
Statement Interface can be used to perform different types of SQL operations whereas PreparedStatement is recommended to perform same operation multiple times.

Q. Explain the working of PreparedStatement?
-> 1. PreparedStatement is an interface declared inside the java.sql package.
2. To create a platform by using PreparedStatement interface we have to use following syntax.
#Syntax: con.prepareStatement(query)
3.In case of prepareStatement(query), we have to pass SQL query at the time of platform creation.
4.PreparedStatement interface supports the concept of placeholders.

Q. Explain the use of callable statement.
1. CallableStatement is an interface declared inside the java.sql package.
2. It is mainly used to execute stored functions and procedures.
3. CallableStatement extends the properties of PreparedStatement so we can use placeholders for dynamic values.
4. To create the platform for CallableStatement we have to use prepareCall() method.

(
Statement-> createStatement()
PreparedStatement->prepareStatement()
CallableStatement->prepareCall()
)

Q. What is stored function.
1.Stored function is a part of PL/SQL programming. (Procedural SQL)
2. Stored functions can be used to develop business logic.
3. We can call stored functions by using any programming language because all stored functions are language independent.
4.SQL developer is responsible to create and maintain stored functions.
5. We can perform different operations on table data as well as external data by using stored functions.

Q. What is stored procedure?
-> 1. Stored procedure is a part of PL/SQL programming.
2. Procedure acts as a container for multiple SQL queries.
3. If we call procedure all queries will be executed at a same time.
4. We can write DML and DQL queries inside the procedure.

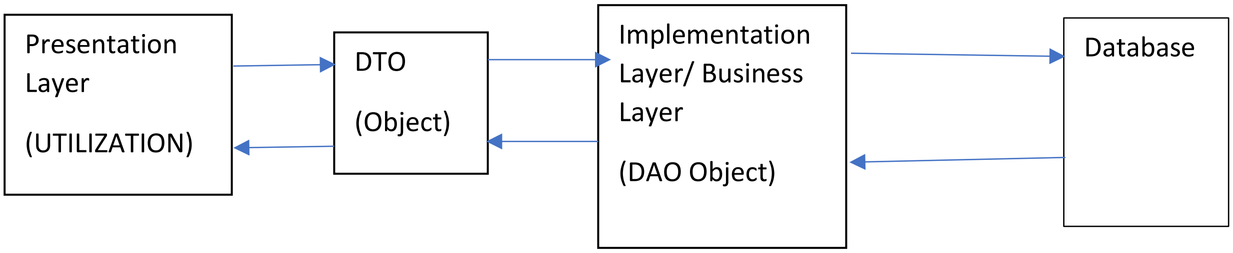
Q. What is the difference between stored function and procedure?

-> 1.Stored functions can be used to develop business logic whereas procedure is mainly used to execute the SQL queries.

2. Every function always returns the value but we cant return the value from procedure.

3.By using stored function we can do logical programming e.g. control statements, flow stats, variables, functions. Stored procedures can be used to perform database operations.

DAO- Data Access Object
DTO- Data transfer Object



DAO/DTO is a pattern used to develop standard database application by using JDBC. In this pattern we have to create 2 types of classes.

1.DAO class: It stands for Data Access Object. DAO class is used to develop business logic to perform database operations. DAO class directly communicates with database vendor.

2. DTO class: It stands for data transfer object. DTO class is a simple java bean class where we can declare private data members with public getters and setters. DTO object is used to transfer the data from one layer to another layer.

Q. What are the advantages of using DAO and DTO design pattern.

1. We can securely transfer the data from presentation layer to implementation layer and vice versa.
2. We can achieve code modularity with the help of DAO/DTO pattern.
3. We can separate presentation and implementation of an object to achieve abstraction.
4. We can avoid direct communication between presentation layer and database.

Q. Steps to develop JDBC application by using DAO and DTO design pattern.

Steps:

1. Create database table.
2. Create DTO class. (Number of columns = no. of variables (private)).
3. Create DAO class for database operations (Implementation class).
4. Create main class to present the data.

#Note: There should not be any printing statement in DAO or DTO class. Printing statement should be present only in presentation layer (Main Class).

Transaction Management:

Q. What is Transaction Management?

The process in which java application sends the request to the database application and database application provides the response to java application is known as transaction. Transaction management is the process of controlling the transaction with the help of commit and rollback operation. In JDBC, by default every transaction is autocomitted. We can manage the transaction explicitly with the help of following methods:

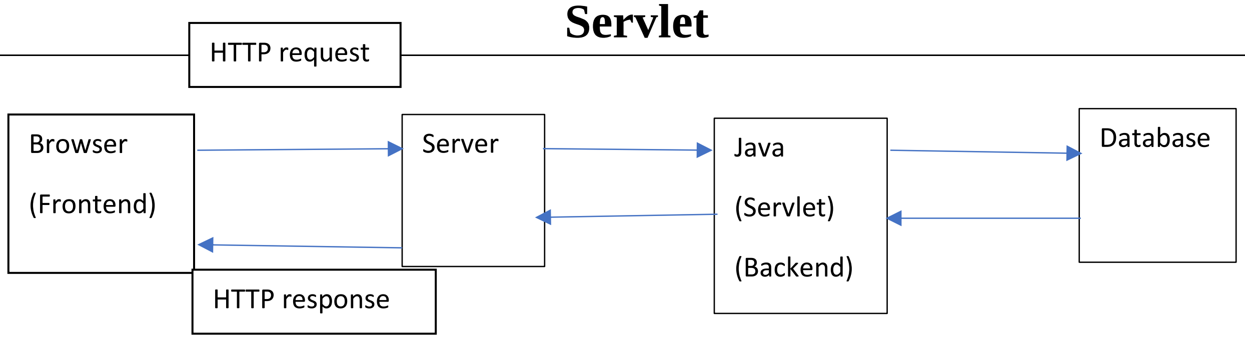
1.setAutoCommit(): This method accepts the boolean value as a parameter. The default value of this methos is always true. It means by default all transactions are autocomitted. In order to manage the transaction explicitly, we have to make setAutoCommit(false).

2.commit(): This method can be used to save the DML changes made by developer.

3.rollback(): This method can be used to revert the DML changes made by developer. After commit operation there is no use of calling rollback(). So we can rollback the changes which are not committed.

1. What is JDBC?
2. Explain imp components of JDBC
3. What is JDBC API?
4. Explain the steps to develop simple JDBC applications
5. Difference between Statement and PreparedStatements.
6. Difference between execute(),executeUpdate(),executeQuery()
7. Adv of using preparedStatements()
8. Purpose of CallableStatement
9. Explain the difference between stored function and procedure(SQL also)
10. What is DAO/DTO design pattern.
11. Explain the working of ResultSet interface.
12. How to manage Transaction in JDBC?

Servlet



1.Static request

2.Dynamic request

Browser application sends the request to the server application. If request is a static then server immediately provides the response to the browser. If request is dynamic then it will be forwarded to the backend where we can use different programming languages to handle such type of requests.

Backend technology can communicate with database in order to provide the response to the end user. This complete process is known as web transaction.

(Types of servers)

1.Web server: for web app (e.g. apache tomcat)

2.Application server: for enterprise app (e.g. glassfish)

3.Database server: (e.g. mysql)

Q.What is server?

Server acts as a container for different web resources.e.g. images, doc file, pdf file, etc. Server plays an important role in the development of web application. There are mainly 3 types of servers.

1. Web server
2. Application server
3. Database server

1.Web Server: Web server can be available in the form of hardware or software. The basic functionality of web server is accepting the request and provide the response to the end user. Web server can be used to deploy web applications e.g. apache tomcat server.

2.Application server: Application server is available in the form of software. The main functionality of application server is transferring the business from one application to another application. We can deploy enterprise application by using application server. e.g. glassfish.

3.Database server: Database server is available in the form of software. The main functionality of database is performing the database operation. We can deploy database application by using database server. e.g. mysql, oracle etc

MIME Type (Multipurpose Internet Mail Extension): Server is a collection of multiple resources. And every resource is having specific type which is known as MIME type or resource 3type. If browser application sends the request to the server application, then server provides the response based on resource type. E.g. if user has requested image file then automatically image response will be delivered to the end user it is possible only because of resource type. Following are the commonly used resource type in web development.

1.text/html

2.text/css

3.application/Jason

4.text/javascript

5.video/mp4

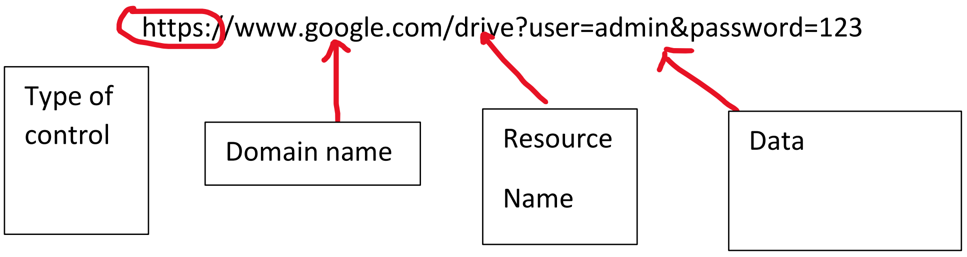
6.img/.png

7.img/.jpg

Developer is responsible to upload the resources in the web server by providing appropriate resource time. We can deploy multiple applications by using one server.

(URL: Uniform Resource Locator)

1. Resource path
2. Data



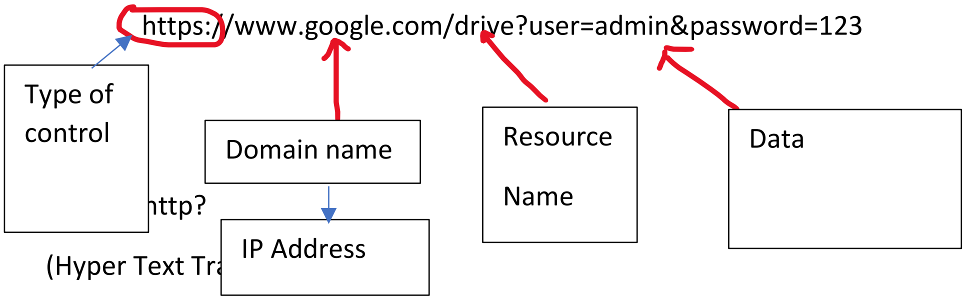
https://www.google.com/drive?user=admin&password=123

URL is used to access the specific resource available inside the server. In other words, URL is nothing but unique identity given to each and every web resource. URL is combination of two factors,

1. Resource Path

2. Data

Resource path consist of type of control, domain name or IP address and resource name. It is possible to pass the data through URL. But data must be present in the form of key and value pair. Every key value pair should be separated by ‘&’. Following is the syntax for creating an URL.



https://www.google.com/drive?user=admin&password=123

http?

(Hyper Text Tra

Hyper Text: Browser understandable language.

http is a set of instructions which can be used to provide the communication link between two resources. The basic functionality of http protocol transfers the data from one page to another page. Without http protocol we can’t access the web resources. http is stateless protocol which means server will not remember what was the request and what was the response after completion of one transaction. Following are the imp factors of http protocol

1. http request
2. http response

http request: It acts as container form

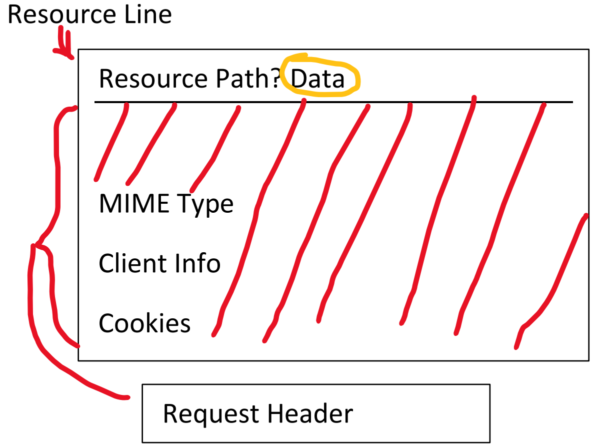
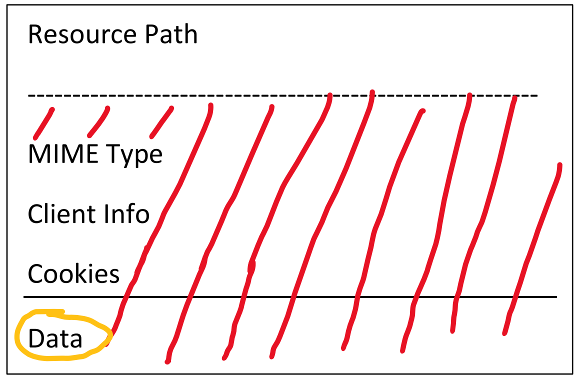
1.URL
2.Cookies
3.MIME Type
4. Data

Following are the imp types of request:

1.GET: it is used to perform select operation
2.POST: it is used to perform insert operation
3.PUT: it can be used to perform update operation
4.DELETE: It can be used to perform delete operation
5. HEAD: It is mainly used to fetch header information from HTML page.

What is the difference between GET and POST request?

GET Request (Fetch)	POST Request (Submit)

Resource Line  Resource Path? Data MIME Type Client Info Cookies Request Header	R  Resource Path MIME Type Client Info Cookies Data
GET request is used to fetch the info from server.	Post request can be used to submit the info at sever site
By using GET request we can send limited amount of data (256 char)	By using POST request we can send huge amount of data.(Unlimited)
GET request is not a secured request.	POST request is a secured request.
In case of GET request, data will be exposed through URL.	In case of POST request, data will be stored into separate sections.
We can bookmark GET request.	We can't bookmark POST request.

Explain HTTP response?

HTTP response contains following factors

1. Status code or message
2. Cookies
3. Actual Data

    1. Status Code
        a. 200: Successfully handled request
        b. 404: Page not found (When resource is not there)
        c. 405: GET-POST mismatch
        d. 500: Internal Server error (Error in the Logic)

Apache TOMCAT->TOMCAT 9->64bit Windows Zip->

What is Apache Tomcat Sever?

= It is an open-source web server which can be used to deploy web application. Tomcat sever is developed by Apache software foundation. Tomcat server can be used for java platform. Tomcat server is available in different versions.

Tomcat 11(alpha)

Tomcat 10

Tomcat 9

Tomcat 8

How to configure Apache Tomcat server?

=In order to configure Tomcat server, we have to add environment variable.

1.JAVA_HOME: This variable represents the location of root directory for JDK (c:/program files/java/jdk 1.8)

2.CATALINA_HOME: This is the path of container which is present inside the Apache Tomcat server. (c:/apachetomcat 9.073)

After configuration of environment variable we can start the server by using widows batch file(executable file) named as startup.batch. In order to stop the server we have to call shutdout.batch file

Q. What is the concept of welcome page or Landing page?

=Welcome page is the default initial page of web site. It means whenever end user visits the website automatically default initial page will be displayed to the end user.In most of web application index.html will be considered as default initial page.

It means after starting the server by default index.html page will be executed. We can provide the link of other html file inside the index.html file. If web applications does not contain index.html file then server will show 404 error. i.e. page not found

TO BE CONTINUED

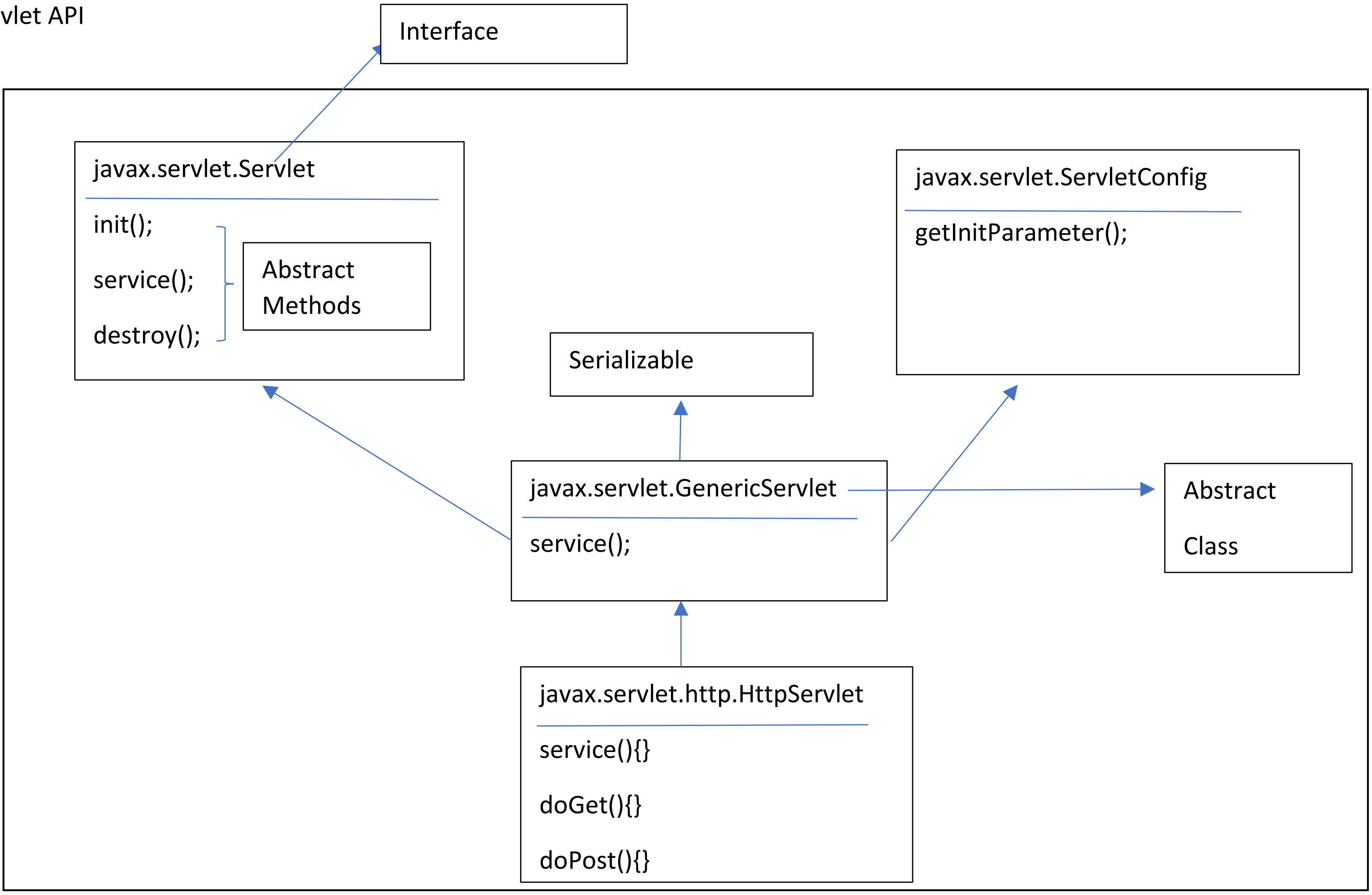
Servet API is available in the form of jar file as a part of Apache of tomcat server.

Q. Explain servlet API?

Servlet API is a collection of different classes and interfaces which can be used to develop web application. Following are the important packages declared inside the servlet API

1. Javax.servlet package
2. Javax.servlet.http package

Servlet API



1)init(): Initialize the resources

2)service(): Accepts the request and provides the response (main code)

3)destroy(): Terminate the servlet

Q. Explain Servlet interface?

-> 1. Servlet is an interface declared inside the javax.servlet package.

2. This interface contains following abstract methods.

1.init(): This is a special method which can be used to initialize the variables and web resources. In case of servlet we can't create parametrize constructor hence we have to use init() for initialization purpose.

2.service(): This method is responsible to provide service to end user. It means this method accepts the request from user and after processing the request provides the appropriate response.

3.destroy(); This method can be used to terminate the servlet. In other words destroy method terminates the object of servlet class.

Q. Explain the class of GenericServlet?

GenericServlet implements the properties of Servlet interface. It is an abstract class declared inside the javax.servlet package.

GenericServlet is abstract in nature because it does not provide the implementation for service() method. We can handle any type of request by using GenericServlet. E.g. http, FTP,SMTP

Q. Explain HttpServlet?

HttpServlet extends the properties of GenericServlet. It is an abstract class declared inside the javax.servlet.http package.This class provides the additional methods to handle the request apart from service(). E.g. doGet(),doPost(),Delete(),Head()

-><- What are the steps to develop simple web application using servlet?

1. Create new dynamic web project or MAVEN project.

2.Add required dependencies.

3.Create html page to interact with user(Front end).

4. Create servlet(.java) file to develop funcationality.

5. Map the html page with appropriate servlet by creating URL.

Following are the options to create servlet class.

1. class demo implements Servlet
2. class demo extends GenericServlet
3. class demo extends HttpServlet

Q. How to fetch the value from html page into servlet?

-> In order the fetch the values from html page we required object of ServletRequest.

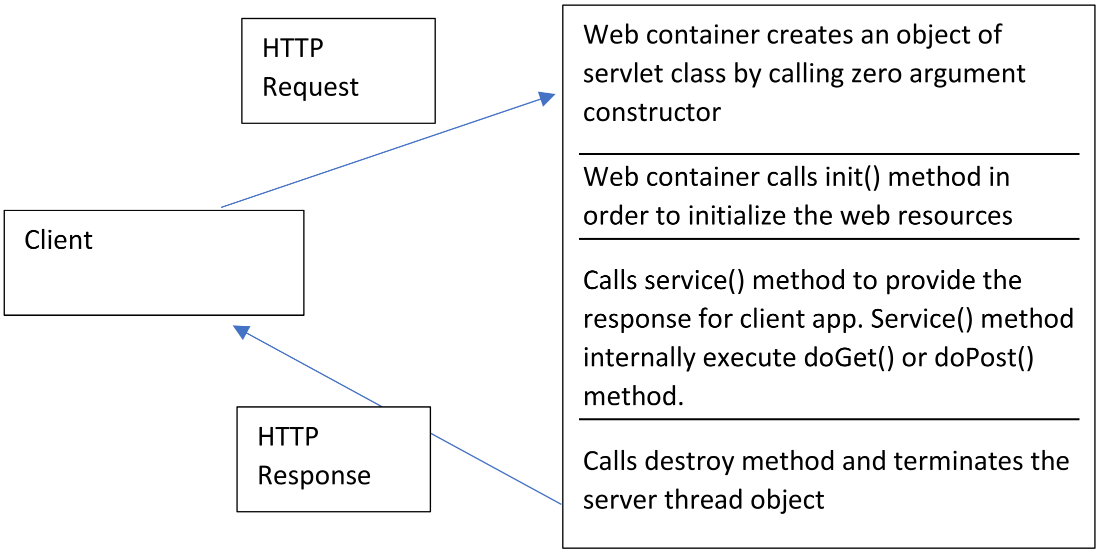
By using the reference of ServletRequest we can call following methods.

To fetch the values from different pom fields

1.getParameter(String args[]): This method can be used to fetch values from text box, drop down list, radio button, and text area fields. //public String getParamter(String str)

2.getParameterValues(String str): This method can be used to fetch the values from check box control. This method returns String array type of variable.//public String[] getParameterValues(String str)

Servlet Life Cycle



Web container creates an object of servlet class by calling zero argument constructor

Web container calls init() method in order to initialize the web resources

Calls service() method to provide the response for client app. Service() method internally execute doGet() or doPost() method.

Calls destroy method and terminates the server thread object

Servlet lifecycle represents execution flow of servlet program. Following are the important phases of servlet lifecycle.

1. Instantiation
2. Initialization
3. Service
4. Destruction

1. Instantiation: When client application sends the HTTP request container will create an object of servlet class. Container can create an object of servlet by calling zero parametrize constructor. Developer cant provide the parametrize constructor inside the servlet because the container will not be able to execute parametrize constructor.
2. Initialization: To initialize the resource we can override init() method which is declared inside the servlet interface. init() method is a replacement for parametrize constructor. Init() method will be executed only once in a servlet lifecycle.
3. Service: service() method is responsible to process the request sent by client application. If we are having generic servlet then we have to override service() method in order to provide response. In case of HttpServlet we can override doGet() or doPost() method to provide the response. This method will be executed multiple times depends on the request sent by the client application.
4. Destruction: After providing the response to the client application container terminates Servlet object. Container implicitly calls destroy() method for the termination of Servlet(). destroy() method will be executed only once in a application life cycle.

   Q. What is servlet chaining?
   -> Servlet chaining is the process of connecting one servlet with another servlet. In case of application development we can distribute the source code among the multiple servlets. If source code is distributed in the multiple servlets then we have to provide the communication link between servlet applications. Servlet chaining can be done by using two ways.
   1. RequestDispatcher Interface
   2. sendRedirect()

   1. RequestDispatcher: It is an interface declared inside the package javax.servlet package. Request dispatcher can be used for servlet chaining purpose. To get the reference of RequestDispatcher interface we have to call getRequestDispatcher() which accepts the URL of next servlet. Following are the imp methods declared inside the RequestDispatcher interface:
      1. Include();
      2. forward();

include()	forward()
Used to merge the responses from multiple servlets	User will get the response only from last servlet.
It is possible to call include() method multiple times inside the single servlet.	It is not possible to call multiple forward methods inside the single servlet.
Response delivery speed is very slow incase of include() method.	Response delivery speed is faster than include() method
It is nothing but merging of multiple servlets.	It means sending the request from one servlet to another servlet without merging the response.

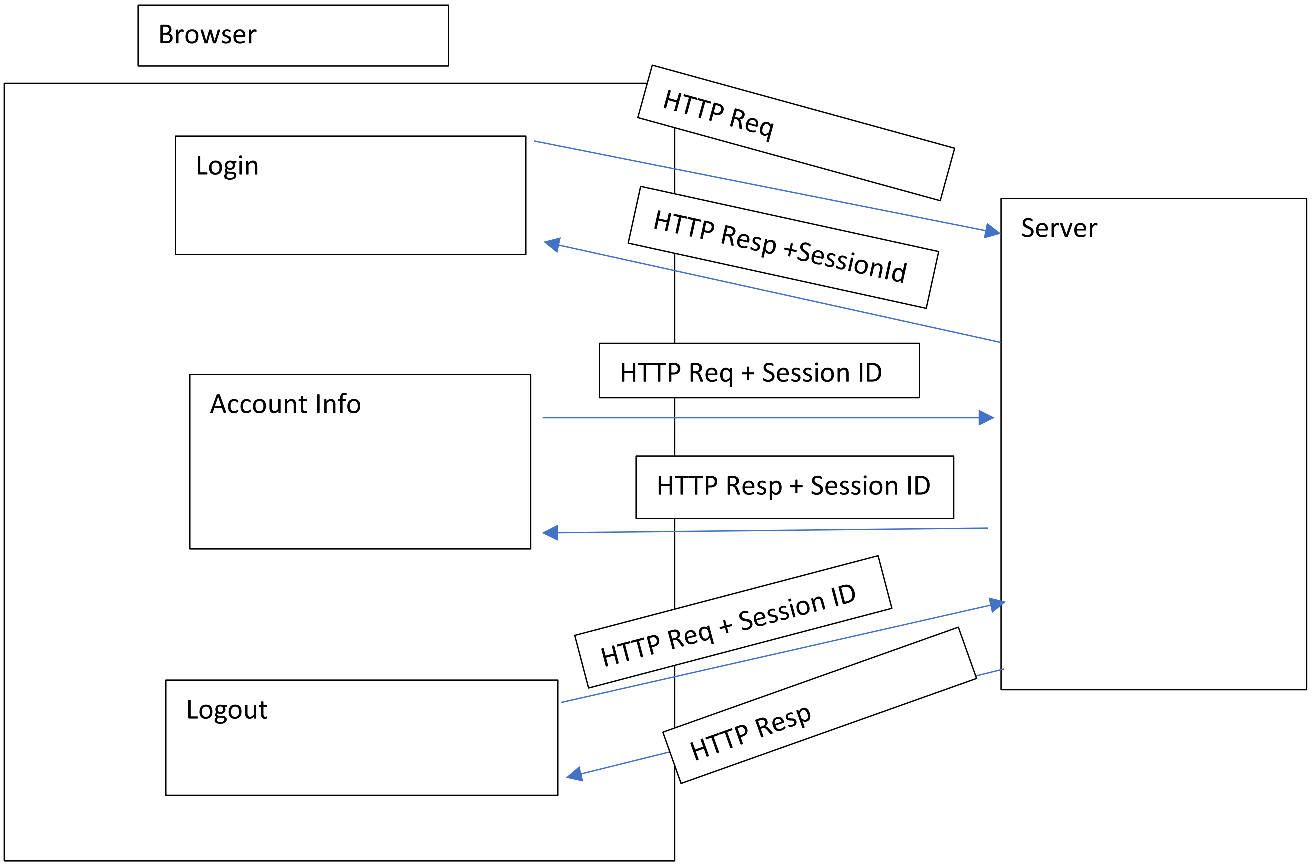
Attribute:
It is a special type of object present inside the servlet. Attribute object methods can be used to transfer the data from one servlet to another servlet. Usually we create the object of a class to access the properties outside the class body. In case of servlet developers can't create the object the class direct. Because of this reason servlet has provided the attribute object so that we can transfer the data from one servlet class to another servlet class without creating servlet object.
Following are the methods of Attribute object:
1.setAttribute(key,value): This method can be used to add the data in the request object of servlet. Data must be present in the form of key, value pair. We can call multiple setAttribute () inside the single servlet provided that every method is having unique key, value pair.
# public void setAttribute(String key, Object value)
2.getAttribute(key): To fetch the additional information from request object. In order to fetch the information, we have to pass the appropriate key. This method always accepts the key and returns the value present in the form of object.
#public object getAttribute(String key)

Q. What is redirection?
1. Redirection is a part of servlet chaining.
2. If one application or website wants to communicate with another application then we have to go for redirection.
3. We can't use request dispatcher for redirection process instead of that we have to call sendRedirect() method. By using RequestDispatcher we can perform intra-application communication. By using send redirect we can perform inter-application communication. sendRedirect() is a method declared inside the HttpServletResponse interface.

Q. Explain session tracking?
-> Session tracking is also known as session handling. Session is nothing but interactive time between client and server. In other words, session is the time duration between login and logout. The process of tracking the client information from login till logout is known as session tracking. Session tracking is required because Http is a stateless protocol. In order to track the session, we need to make Http as a stateful protocol. If we are not handling the session then every request will be considered as a new request. In servlet we can handle the session by using HttpSession interface and Cookies.

Q. Explain the internal working of session tracking process?

->



When user send the login request to server application then server will create new session with unique session id. After creating the session Id server can track the client information. After creating the session, server will provide the HTTP response along with session id. If browser sends the next request then server will check session id present inside the request. If the requested session id is matching with existing session id then old session will be continued for the user. Hence we don't have to provide the username and password for every request. Server application will terminate the session id when browser sends the logout request. Hence after log out compulsory we have to log in to access the features of account.

# Conclusion

In short, server application maintains only one session for specific user until the user sends the logout request.

Q. How to handle session with help of cookies?

-> Cookies are one of the session tracking techniques. In other words, cookies are small piece of data. In case of cookies, data will be stored in the form of key-value pair. Cookies are used to track the session at client side. It means cookies data will be stored into web browser. It is not recommended to track the sensitive information with the help of cookies. e.g. username, pass, OTP, contact no etc. We can use cookies for the tracking of non-sensitive information. e.g. online activities done by specific user. In case of servlet, we can create and manage cookies by using Cookie class which is declared inside the javax.servlet.http package. Following is the process to create and maintain the cookies.

1. Create cookies.

```
Cookie c1=new Cookie(key,value);

resp.addCookie(c1);
```

2.Get all the cookies

```
Cookie data[]=req.getCookies();
```

Q. What is the use of HttpSession interface?

->HttpSession interface is declared inside the javax.servlet.http package. It is one of the session handling techniques. HttpSession is more secure than cookies. Because it traps the client information at server side. Following are the steps the to handle a session by using the HttpSession interface.

1. Create a new session

HttpSession s=req.getSession(); ->Checks for existing session if not present then creates the new session

HttpSession s=req.getSession(true); ->

2.Get existing session

```
HttpSession s=req.getSession(false);
```

3.Stop the session

```
ses.invalidate();
```

Imp metods

1.req.getSession()

This method always checks for existing session. If it is available the same session will be continued else it will create the new session.

2.req.getSession(true)

This method directly creates the new session without checking for existing session.

3.req.getSession(false)

This method always checks for existing session, if it is not available then servlet exception will occur.

4.ses.invalidate()

This method can be used to terminate the existing session. We can call this method only once during the active session.

JSP

JSP stands for JAVA Server Page. It is an extension for servlet technology. By using JSP we can develop dynamic web pages. JSP is mainly used to present the data.

Features of JSP:

1. JSP supports different HTML elements. It means we can use HTML tags directly inside the JSP file. By using JSP we can separate business logic and presentation logic.
2. JSP files are lightweight as compared to java files. We can integrate the java statements inside the jsp file to generate the dynamic output.

JSP Scripting Elements

S.R.	Tag	Description
1	<%@ %>	Page Directive
2	<%-- --%>	JSP Comment
3	<% %>	Scriptlet
4	<%! %>	Declaration
5	<%= %>	Expression

1.Page Directive

Directory tags provide special instructions at the time of execution of JSP file. By using this tag we can import different classes and interfaces.

2.JSP Comment

This tag can be used to provide comment inside the JSP file. We can’t use this tag inside the other tag.

3.Scriptlet

It is the important element for JSP file. Because we can add normal java statements inside the JSP file with the help of scriptlet. it is possible to create any number of scriptlet inside the single jsp file

4.Declaration tag

It is mainly used to declare variables and methods. Variables declared by using declaration tag will be treated as global variable.

5.Expression tag

It is used to print different expressions of java language. It means we can print the outcome of the expression by using this tag. We can use any number of expression tags inside the JSP file.

Hibernate is a framework for JDBC. Complex SQL operations are performed using Hibernate. Spring is web framework which is used to develop web applications (extension for Servlet and JSP).

JDBC (Basic Operations)->Hibernate framework (Complex Operations).

Servlet & JSP(Basic)->Spring framework (Advanced Operations).

Hibernate

What is Hibernate?

Hibernate is a framework which is used to develop high scale database applications. Hibernate is an extension for JDBC. It means we can develop basic plus advanced applications by using Hibernate. Hibernate is not a replacement for JDBC.

Explain important terminology of hibernate framework?

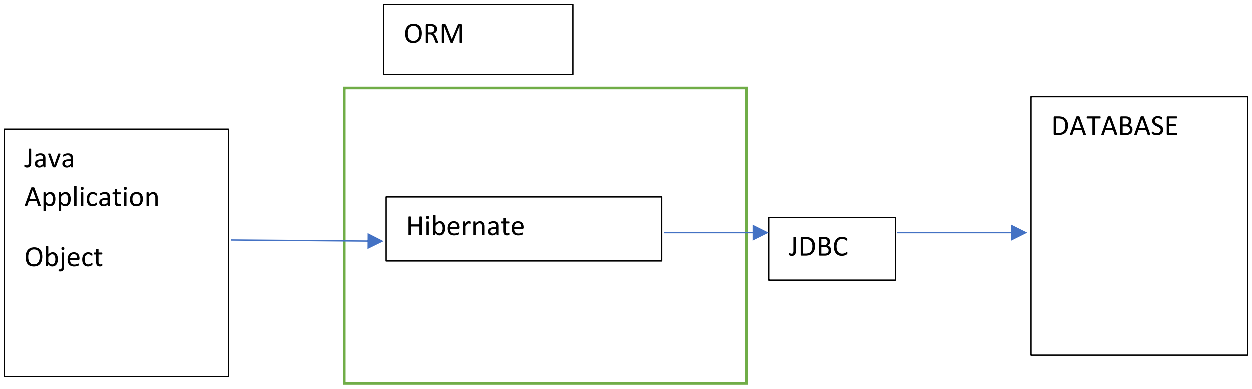
1. Persistence: It is a process for storing the data for long time.
2. Persistence Storage: It is a location where we can store the data for long time. E.g. file, database
3. Persistence data: It is data present inside the persistence storage is known as persistence data. e.g. file data, table data
4. Persistence operations: We can perform persistence operations on persistence data. e.g. insert, update, select, delete
5. Persistence logic: Developers need to develop persistence logic in order to perform persistence operations. E.g. JDBC code, Hibernate code

6. Persistence Tech/Framework: To develop persistence logic we have to use persistence technology or framework. e.g. JDBC is a persistence technology and Hibernate is a persistence framework.

Hibernate (Java framework)-
1.Open source
2.ORM (Object Relation Mapping)
3.Non-invasive (doesn't require to extend or implement any class)
4.Doesn't force to handle compile time exception.
5.Developed in 2001 by Gauin King.

Q. What is Hibernate?
Hibernate is a java framework that simplifies the development of java application to interact with database. Hibernate is an ORM (Object Relational Mapping) tool. Hibernate is an open source and lightweight software. Hibernate is non invasive framework means it won't force the programmer to extend or implement any class or interface. It was invented by Gauin King in 2001. Any type of application can be developed by using hibernate.

Q. What are the drawbacks of JDBC?
JDBC code is database software dependent. It means if we change the database then we have to change the persistence logic. Switching from one database to another database is very complex process. Exception handling is mandatory at the time of developing persistence logic. In JDBC, we have to represent every SQL query as a string value. Transaction management is very big in terms of JDBC.



ORM- ORM is object relational mapping. ORM tool simplifies the data management, data creation and data retrieval. ORM tool is used to map the non-static data members of the class with the columns of database table. One object will be considered as one record in the database table.

What are the advantages of ORM?

-ORM is database software independent. We can easily change the database software. Exceptional handling is not mandatory in hibernate. We can represent information in terms of object without converting them into SQL queries. Persistence logic is portable across multiple database platform.

Q. Definition of Hibernate?

Hibernate is an open source, lightweight ORM framework that can be used to develop plateform independent database application. Open source doesn't mean free of cost, also source code is available. By using hibernate, we can operate relational database. In hibernate, we can write object-oriented queries.

When to use JDBC?

- To operate huge amount of data.
- Batch by batch. (At a time big data)

When to use Hibernate?

- Operate huge amount of data.
- Part by part. (at a time small data)

Q. When to use JDBC and when to use Hibernate?

-> JDBC can be used to operate huge amount of data coming batch by batch. One batch consists of thousands of records for example, in e commerce application multiple customers can place the order at same time so that data will be stored into the batch. Hibernate is recommended to operate huge amount of data coming part by part e.g. if you are adding 10-20 records at a same time then we can use Hibernate framework. It is not recommended to use Hibernate for the processing of huge amount of data coming batch by batch because Hibernate will create separate object for every record.

POJO- Plane old Java Object

POJI- Plane old Java Interface

Q. POJO class?

->(Plane old Java Object). The java class without any speciality is known as plane old java object. In other words, the class which doesn't extends or implements any advanced classes or interfaces is known as POJO class.

e.g

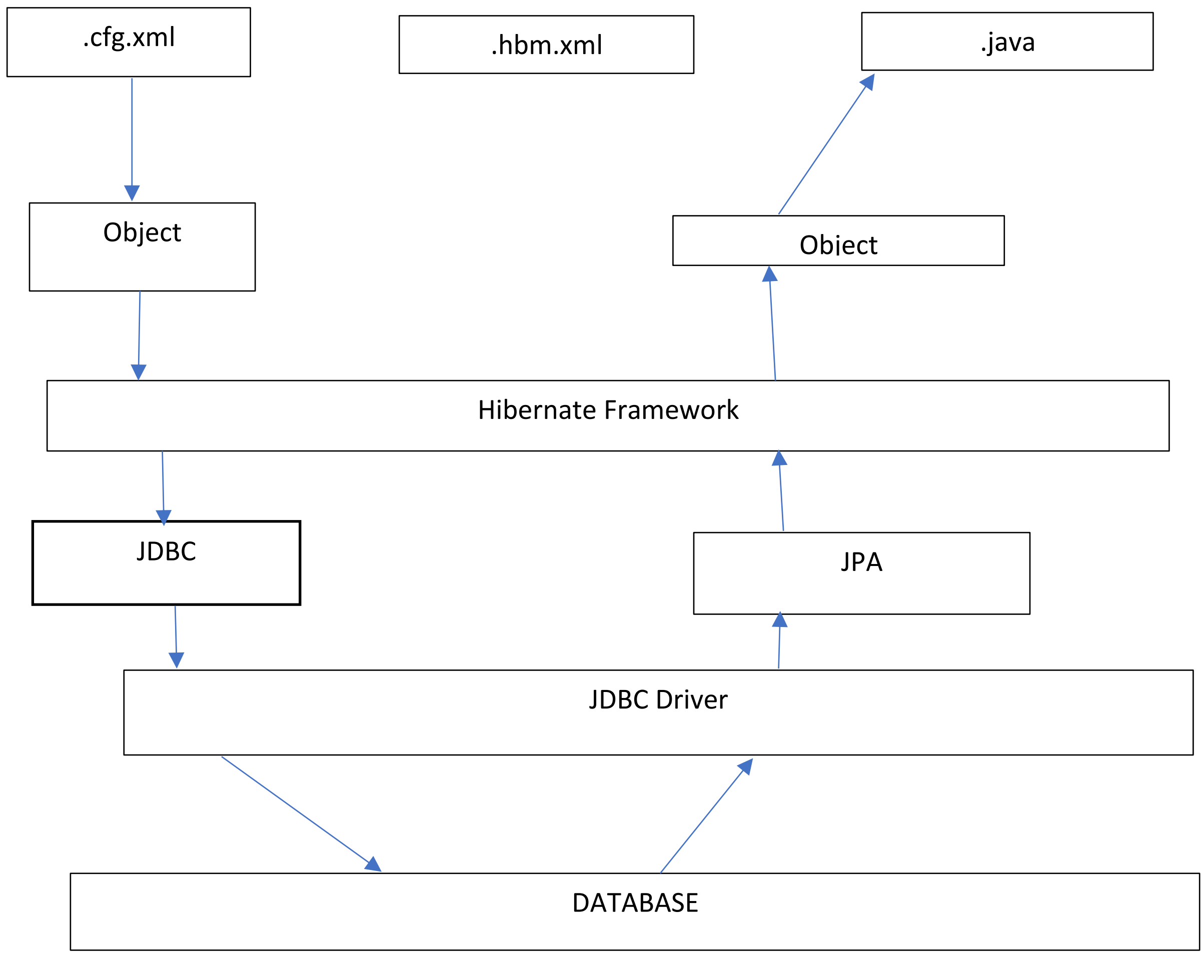
```
public class Demo

{

(POJO Class)

}

public class sample extends HttpServlet

{

Not POJO Class


}
```

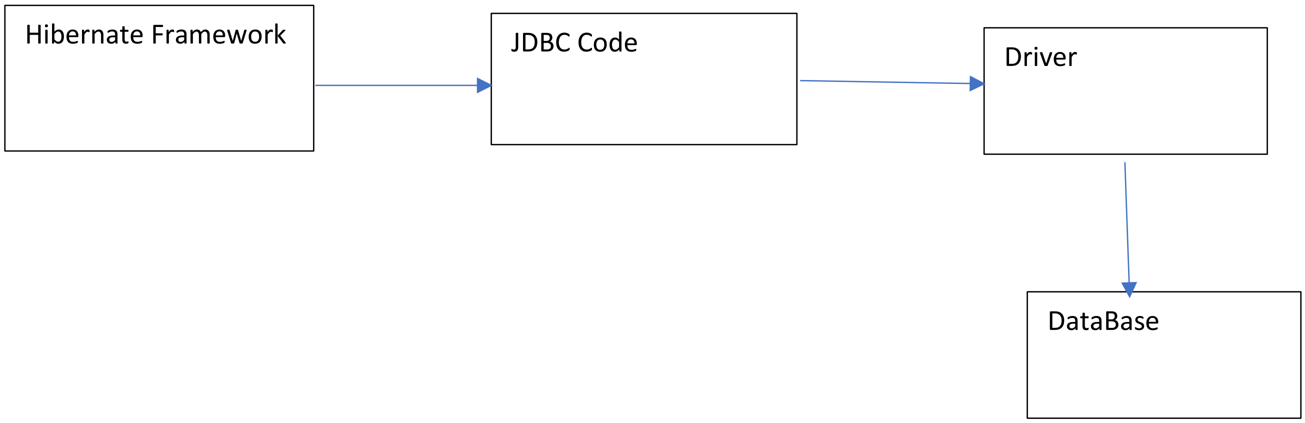
Q. Plane old java interface(POJI)?

The interface without any speciality known as plane old java interface. The interface which doesn't extends the properties of advanced interfaces is also known as plane old java interface.

```
Interface A

{

POJI

}

Interface B extends Servlet

{

Not POJI

}
```

Hibernate Architecture





1. Developer layer: .cfg.xml:- Configuration file- Database information
   . hbm.xml: - Table Information
   .java: - POJO Class
2. Framework layer: Hibernate (Takes object form data and converts it into simple SQL form and gives it to technology layer)
3. Technology layer: JDBC- (Simple SQL form ->Driver)
4. Database Layer: Driver+ Database

   Hibernate Architecture is having different layers

   1. Developer Layer
   2. Framework Layer
   3. Technology layer
   4. Database Layer

1. Developer layer: Developer is responsible to create 3 types of files.
   1. . cfg.xml: Configuration file
   2. . hbm.xml: Mapping file
   3. .java: Java bean class
2. Framework Layer: Hibernate framework understands object-oriented queries created by developers. Hibernate is also responsible to convert object-oriented query into simple form.
3. Technology Layer: Hibernate requires one or more technologies to communicate with database. Usually hibernate takes the help of JDBC technology to communicate with the database. JDBC communicates with the database with the help of JDBC Driver.
4. Database layer: Database server accepts the request sent by JDBC server and provides the appropriate response.

   JPA- Java Persistence API- Converts simple SQL form->Java object form.

Q. Explain important files in Hibernate Application?

1. To develop simple hibernate application, we need to create following files

1. .cgf.xml- This is the main configuration file in hibernate application. Here we can configure database related information e.g. database name, version, URL, username, password etc. One hibernates project can have only one configuration file. Execution of hibernate program starts from .cfg.xml file.

2. .hbm.xml file: This is the mapping file which is used to map database table with the java class. This file contains the information about database tables. We can create multiple mapping files based on number of tables. It is mandatory to provide the reference of mapping file inside the configuration.

3. .java file: In this file, we can create simple java bin class with private data members and public getters and setters. A class can have multiples members variables depends on column present inside the db tables. We have to create multiples java classes based on number of tables.

Number of tables = number of java classes=number of mapping files=1 .cfg.xml file

Important Objects

1. Configuration: (class)-pkg : org.hiberante.cfg
   Configuration cfg=new Configuration ();
   cfg.configure();   -read .cfg.xml file
2. SessionFactory: (interface)-pkg.org.hibernate
   SessionFactory factory=cfg.buildSessionFactory();
3. Session: (Interface)-pkg:org.hibernate
   Session ses=factory.openSession();
   ses.save()-
   ses.update()
   ses.delete()
   ses.load()/ses.get()
4. Transaction: Interface
   Transaction tx=ses.beginTransaction(); For commit and rollback operations

Explain important objects of hibernate application?

= Following are the important object of hibernate application.

1.Configuration- Configuration is a class declared inside the org.hiberante.cfg package. We have to create an object of configuration class in order to activate hibernate framework. After creating an object of class, we have to call configure() method. Configure method is responsible to read the data from cfg.xml file.

Configuration cfg =new Configuration();

cfg.configure();

2.SessionFactory: SessionFactory is an interface declared inside the org.hibernate package. SessionFactory object is responsible to establish the communication link with the database server. To get the reference of type SessionFactory, we have to call buildSessionFactory() method.
SessionFactory factory=cfg.buildSessionFactory();

3.Session: It is an interface declared inside the org.hibernate package. Session object is responsible to perform persistence operations i.e. insert, update, delete,   To get the ref of type Session interface, we have to call openSession() method.

```
Session ses=Factory.openSession();
ses.save()
ses.update()
ses.delete()
ses.get()/ses.load()
```

4.Transaction; It is an interface declared inside the org.hibernate pkg. Transaction object mainly used to perfaorm DML operations. To start the Transaction, we have to call beginTransaction() method. In hibernate, transaction management need to be done explicitly.

```
Transaction tx=ses.beginTransaction();
```

Steps to develop simple hibernate app

We have to follow several steps to develop hibernate application.

1. Create database table
2. Create MAVEN project with hibernate dependency.
3. Create cfg.xml file
4. Create java bin class (POJO) class.
5. Create hbm.xml file
6. Create main class to declare hibernate object.

Dependencies

Hibernate 5.6.9 dependency
Mysql connector 8.0.29

Q. What is the difference between ses.load() & ses.get() method?

-> Both the methods are used to fetch the specific record from the database table. If we call ses.load method by passing invalid id then jvm will throw ObjectNotFoundException. If we pass invalid id to ses.get() method then this method will return null value. It is recommended to use ses.get method to fetch the record from the table.

Q. What is the use of dialect property in configuration file?

-> dialect property of hibernate converts object-oriented information into appropriate SQL query. This property dynamically generates the query based on selected database vendor. E.g. mysql, oracle. Developer can write object-oriented query while developing hibernate application but relational databases understand only structured query language. By configuring dialect property we can write platform independent query.

hbm.xml- to map java class to database table.

Imp tags in hbm.xml

1. (Class tag) <class></class> : This tag is having 2 attributes i.e. name, table.
   name attribute represents fully qualified class name. table attribute represents the name of database table.
2. (id tag) <id/>: This tag is mainly used to represent primary key column. This tag is having two attributes i.e. name and column. name attribute represents the variable name declared inside the java bin class. Column attribute represents the name of table-column which is having primary key constraint.
3. (property tag): This tag is used to represent non primary key columns in the database table. This tag is also having two attributes i.e. name and column.

Q. Explain annotation-based configuration on hibernate?

Configuration can be done with the help of xml file or annotation. Annotations are the pre-processors declared inside the java class. Annotation provides the additional info about class and class members. Annotations are processed at the time of compilation or execution. Following are imp annotations which can be used to configure hibernate application.

1. @Entity- This annotation represents the java class available for mapping.
2. @Table- This annotation represents the table name which has to be mapped with the java class.
3. @Id- This annotation represents the variable which can be mapped with the primary key column.
4. @Column- This annotation represents the column name which has to be mapped with variable.
   We have to import these annotations from javax.persistence pkg. If we are using annotations in the domain class then no need to create hbm.xml file.

Q. What is InstanstiationException in hibernate?

-> If developer calls the get or load method to fetch the specific record from database table then hibernate creates an object of that particular method. Object creation is possible only if domain class contains zero parametrize constructor. If class doesn’t have default constructor, then hibernate can’t create an object of that particular record. In such a case JVM will throw InstanstiationException.

Hibernate Generation Strategy

Q. Explain the generation strategies in hibernate?

-> Hibernate generation strategies are mainly used to generate primary key values. Following are the different types of generation strategies.

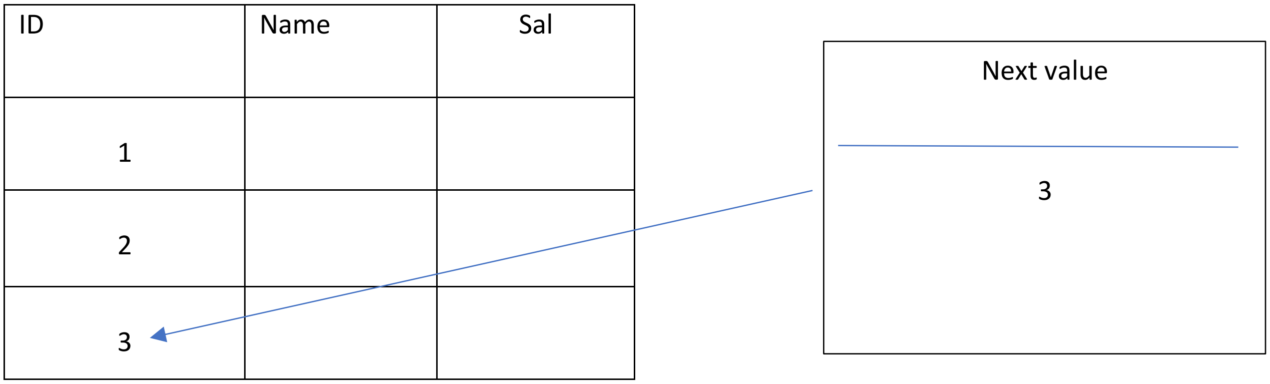
1. GenerationType.AUTO

2. GenerationType.IDENTITY

3. GenerationType.SEQUENCE
4. GenerationType.TABLE

1. GenerationType.AUTO: If GenerationType is auto then ORM framework will select anyone generation strategy from the available options. In case of hibernate, If we use GenerationType.AUTO then it will select SEQUENCE Strategy to generate primary key.

2. GenerationType.IDENTITY: This strategy is mainly applicable for the database which supports autoincrement feature. Without autoincrement there is no use of GenerationType.IDENTITY. Hence it is not reliable strategy to generate the primary key value.
3. GenerationType.SEQUENCE: This strategy can be used to generate the primary key values without having the support of autincrement feature. This is the most reliable strategy to generate primary key becauses this is applicable for all type of database. By using GenerationType.SEQUENCE, we can create custom sequence. In case of this strategy, sequence will be maintained in the db separate table which is having only one column i.e. nextvalue



ID	Name	Sal
1		
2		
3		

Hibernate HQL (Hibernate Query Language):
Hibernate Query Language is used to perform the bulk operations. It is platform independent query language. It is object-oriented query language.

Q. What is HQL?
-> HQL stands for Hibernate Query Language.
HQL is the own query language of the hibernate framework.
HQL can be used to perform the bulk operations. e.g. fetching the multiple records, updating the multiple records, deleting the multiple records.
HQL is platform independent query language. HQL is object oriented in nature. HQL query is similar to SQL query, only the difference is that column name is replaced with the variable name and table name is replaced with class name (domain class). Since it is the platform independent query language, we can use any database server in backend. E.g. oracle, mySQL etc.

SQL Query:
Select column_name from table_name;

Select * from product_info;

Select name from product_info;

HQL Query:
Select ref variable.variableName from DomainClassName ref variable;

Select p from Product p;

Select p.productName from Product p;

Hibernate Criteria Query Language (HCQL):

It is the advanced version of HQL.
Criteria: Requirement of fetching the data based on the specific conditions.

Q. What is HCQL?

It stands for Hibernate Criteria Query Language. HCQL is an extension for HQL. By using HCQL, we create advanced criterias to fetch the data from database based on the specific conditions. In case of HCQL, we can create single criteria of multiple conditions. Following are the steps to create and execute criteria query.

Step 1: Create criteria for specific class.

Step 2: Add conditions or restrictions into criteria. (Optional)

Step 3: Execute the whole criteria.

Step 4: Process the output.

Hibernate Mapping:

1. One to one mapping
2. One to many or many to one
3. Many to many

Q. What is Hibernate Mapping?

1. Hibernate mapping mainly used to perform joins operations. We can perform 3 types of mapping in hibernate framework

1. One to one mapping

2. One to many or Many to one

3.Many to many

1. Hibernate One to one mapping

If two tables are having one to one relationship then we can use hibernate one to one mapping. If one record from left table is having only one matching record in right table, then it is known as one-to-one relationship. In case of one-to-one relationship, any one table can have foreign key of another table.

2. Hibernate one to many or many to one mapping

If two tables are having one to many relationship, then we have to use hibernate one to many mapping table. If one record from left table is having multiple matching records in right table then it is known as one to many relationship. In case of one to many relationship, primary key of one side will become foreign key of many side.

3. Hibernate many to many mapping

This mapping technique is used if two tables are having many to many relationship. If one record of left table is having multiple matching records of right table and one record from right table is having multiple matching records in left table then it is known as many to many relationship. In case of many to many relationship, we have to create separate table which contains the primary keys of both the tables. In order to use hibernate mapping techniques it is mandatory to normalize the data by using different normalization forms i.e 1 NF, 2 NF, 3NF, 3.5NF(optional).

Movie_Info

Movie_ID (Primary Key)	MovieName	Relese date

User_Info

User_ID(Primary key)	User_Name

Rating_info

Rating_ID(Primary key)	Movie_ID(Foreign Key)	User_ID(Foreign Key)	Rating

Actors_info

Actor_Id(Primary Key)	Actors_Name

Role_info

Actor_ID(Foreign Key)	Movie_ID (Forign Key)	Role

Crew_Info

Crew_ID(Primary key)	Crew_Name

Role_Info

Crew_ID(FK)	Movie_ID(FK)	Role

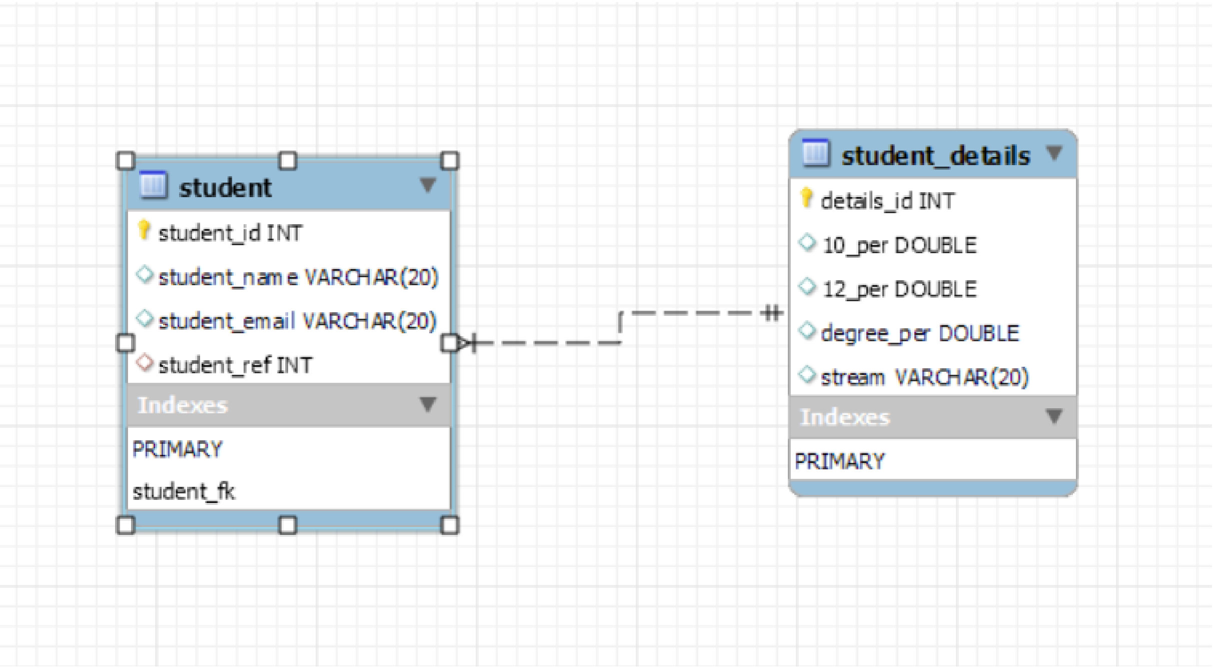
1. Hibernate one to one mapping
   - 1.1 Create database tables with relationships
   - 1.2 Create MAVEN project with hibernate dependency
   - 1.3 Create cfg.xml file for configuration
   - 1.4 Create domain classes with annotation
   - 1.5 Create mainapp to perform the operations

Student

SID(PK)	Name	Email	Sref(FK)

Student_Details

Sd_ID(PK)	10th per	12th Per	Degree	Stream



Q. Explain cascade operations in SQL(Hibernate)?

-> 1. Cascade operations are nothing but parallel operations. It means if we perform some operations on one entity then same operations will be performed on related entity. Following are the types of cascade operations.

1. CascadeType.PERSIST -This is also known as cascade insert operation. It means if we save the changes in first entity automatically changes will be applicable on second entity. (INSERT)

2.CascadeType.MERGE- This is also known as cascade update operation. It means if we modify first entity then parallely second entity will be modified.(UPDATE)

3. CascadeType.DETACH- This operation mainly used to disable the relationship between two tables.

4. CascadeType.REMOVE- This is also known as cascade delete operation. It means, if we delete the record from one table then automatically record will be deleted from related table.(DELETE)

5. CascadeType.REFRESH: This is also known as cascade select operation. It means if we are fetching the information from one table then automatically information will be fetched from related table.(SELECT)

6.CascadeType.ALL : This will perform all the operations mentioned above.

Many to many relationship

Q. Explain the process to implement Many to many relationships in hibernate?

- If two tables are having many to many relationship then we have to create the separate mapping table which contains the primary keys of both the tables. Following are the steps to develop hibernate many to many application

1. Create the database table and provide the relationship with the help of foreign keys.

2.Create domain classes in order to map database table.

3.Create main class to perform the operations.

Student Table

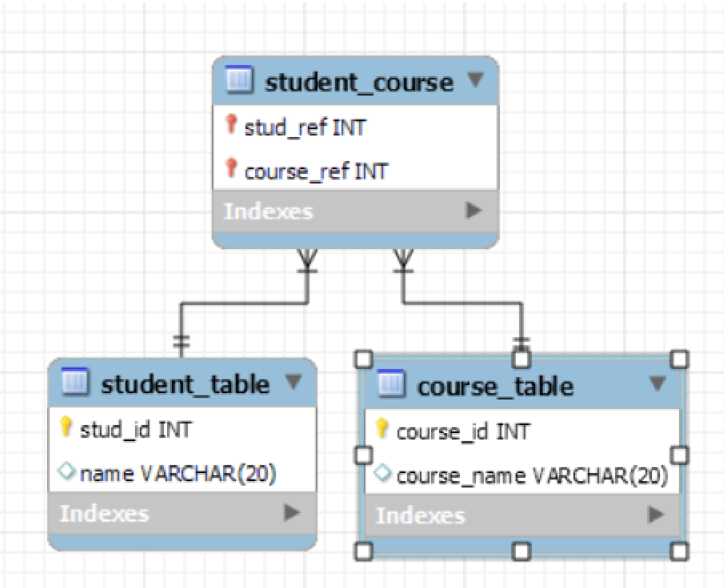
S_ID(PK)	NAME

Course Table

C_ID(PK)	NAME

Student_Course Table

S_ID	C_ID


student_course
stud_ref INT
course_ref INT
Indexes
student_table
stud_id INT
name VARCHAR(20)
Indexes
course_table
course_id INT
course_name VARCHAR(20)
Indexes

Spring

Framework: It is a platform or foundation.

Spring: Open-source web framework.

Spring-

1. Spring framework
2. Spring Boot
3. Spring MVC
4. Spring Data
5. Spring Security

   Spring-High scale application for web + mobile (Hybrid).
   New version of Spring is Spring 6.

Q. What is framework?

Framework is a platform for developing software application. It provides foundation on which software developers can build program. Framework provides advanced classes and libraries to develop application.

Q. What is Spring?

Spring is an eco-system which acts as a container for different projects. Spring framework is open-source application framework which can be used to develop web and mobile application. It is a popular framework to develop enterprise application. Following are the important projects of Spring.

1. Spring framework
2. Spring Boot
3. Spring MVC
4. Spring Data
5. Spring Security
6. Spring Batch

Q. What are the features of Spring?

1. Spring framework provides the flexible set of extension and third-party libraries.

2. Spring framework provides support for IoC (Inversion of Control) & DI (Dependency injection)

3. Spring framework allows to integrate persistence API. (e.g. JDBC, Hibernate)

4. Spring framework allows developers to build web application based on MVC architecture.

5. Spring framework helps users in error handling with the help of in-built libraries.

Project Building

1. Microservice- (One project is divided into many sub projects)
2. Monolith- (Entire project as a single Project)

Reactive Programming- (Asynchronous programming)

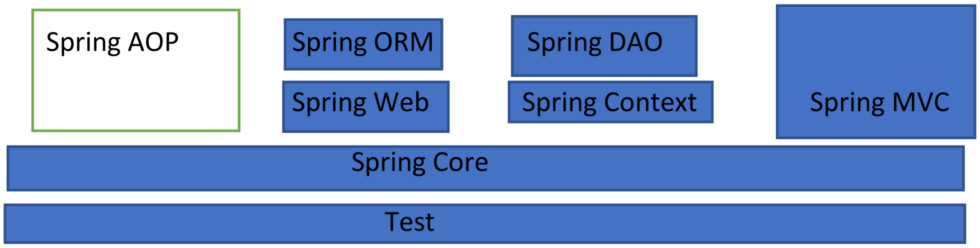
Q. Why spring is popular?

1. Spring is a popular web framework because of some advanced features.

1.1 Microservices- It is modern approach to develop software application. In case of microservices application code will be divided into small, manageable piece. All microservices are independent of each other. Microservices are easy to build and maintain.

1.2 Reactive: Reactive processing enables the developers to build asynchronous and non-blocking applications. Reactive systems can easily use modern processor.

Spring Modules:



->Test- Unit Testing

->Spring Core: IOC- Inversion of Control

DI-Dependency Injection

Bean Factory

->AOP- Aspect Oriented Programming

-Security

- Transaction

-> Spring ORM- Hibernate

->Spring DAO- JDBC

->Spring Web- HTML, CSS, JS, AJAX

->Spring Context- Desktop

Spring framework acts as a container for different modules which plays an imp role in application development.

1. Spring Test Module: Testing is an integral part of software development. Spring test layer provides the support for testing with different tools. Spring framework has provided different packages for testing.

    1.1 Unit Testing
    1.2 Integration Testing

2. Spring Core: Spring core module acts as a container for the core components of the spring framework.
    2.1 IoC- Inversion of Control
    2.2 DI- Dependency Injection
    2.3 Bean Factory
3. Spring AOP (Aspect Oriented Programming)- AOP is a different approach to manage multiple objects present inside the application. AOP plays an imp role in transaction management and handling the security concerns.
4. Spring ORM- Spring ORM module is used for accessing the data from database. This module provides the readymade API to perform the operations on database with Hibernate and other ORM frameworks.
5. Spring DAO- This module of spring framework provides the support to access the data by using the JDBC technology. By using this module we can achieve heterogeneous java database connectivity.
6. Spring Web- This module of spring framework can be used to develop web-based applications. This module supports different frontend tech(like HTML,CSS,JS,AJX)
7. Spring context- This is used for development of desktop applications.
8. Spring MVC- This module implements the MVC architecture of servlet to develop web based applications. This module seperates the Model, View and Controller layer.

Spring IoC (Inversion of Control)

Approach of outsourcing construction and management of object.

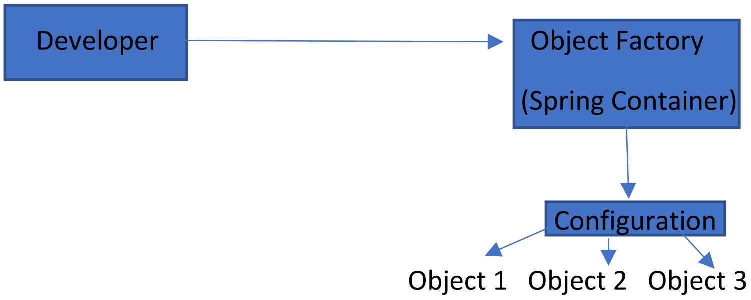
Q. Explain Spring IoC?

-> IoC stands for Inversion of Control. It is a principle of software engineering by which control of objects transferred from developer end to the container end (Framework end). It is the approach of outsourcing the construction and management of the object.

Advantages of IoC:

1. We can easily achieve loose coupling between objects.
2. We can easily switch from one implementation to another implementation.
3. Spring IoC helps to achieve code modularity (easy to test and maintain).
4. We can easily test applications by isolating different components (unit testing).

Big Picture:



Steps to develop spring IoC application:

1. Create MAVEN project with Spring Dependency (Spring Core(6.0.8) and Spring Context(Same version as that of core)).
2. Create .xml file for the configuration of the objects.
3. Create interface with different implementation classes.
4. Load the .xml file into main class to access the object configuration.

Q. What is the use of applicationContext.xml file?

-> This is the main configuration file for spring application. Developer can configure objects and dependencies with the help of applicationContext.xml file.

Following is the syntax

<bean id=”m1” class=”Master.class”/>

In case of Spring IoC, developer is not responsible for object creation. Based on the configuration provided by the developer Spring container will create an object.

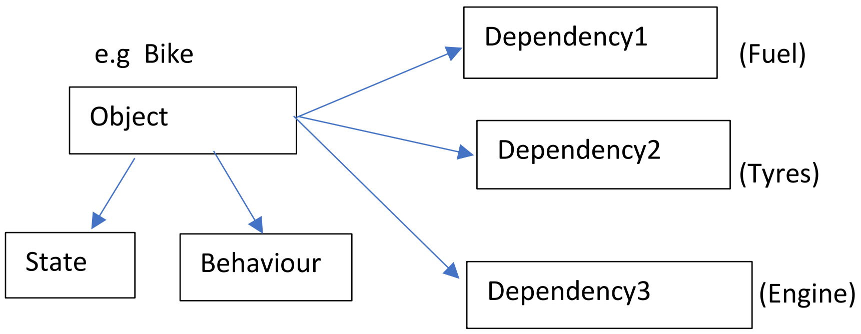
Object Creation

Master m1=new Master();

Q. What is ClassPathXmlApplicationContext?

->ClassPathXmlApplicationContext is a class declared inside the org.springframework.context.support package. This class is responsible to load the data and configurations declared inside the applicationContext.xml. To get the reference of an object created by spring container we have to call getbBean() method. getBean() method accepts the classname and ref variable as an input parameter and returns the address of an object.

Dependency Injection



1. Constructor Injection
2. Setter Injection

Q. What is Dependency Injection?

-> Dependency Injection is a software engineering principle where we can deploy the object along with its dependency. Every object will have its own dependencies (requirements). Developer is responsible to fulfil the requirements of an object before deploying to the user.

Q. Following are the types of dependency injections?

1. Constructor Injection

2. Setter injection

Q. What is constructor injection?

-> The process of injecting the dependencies through constructor is known as constructor injection. To achieve the constructor injection we need to create parametrize constructor which accepts the ref of dependency object. It is possible to inject multiple dependencies by creating one or multiple constructors.

Following is the syntax to use constructor injector

Constructor (Ref dependency 1,-----,n)

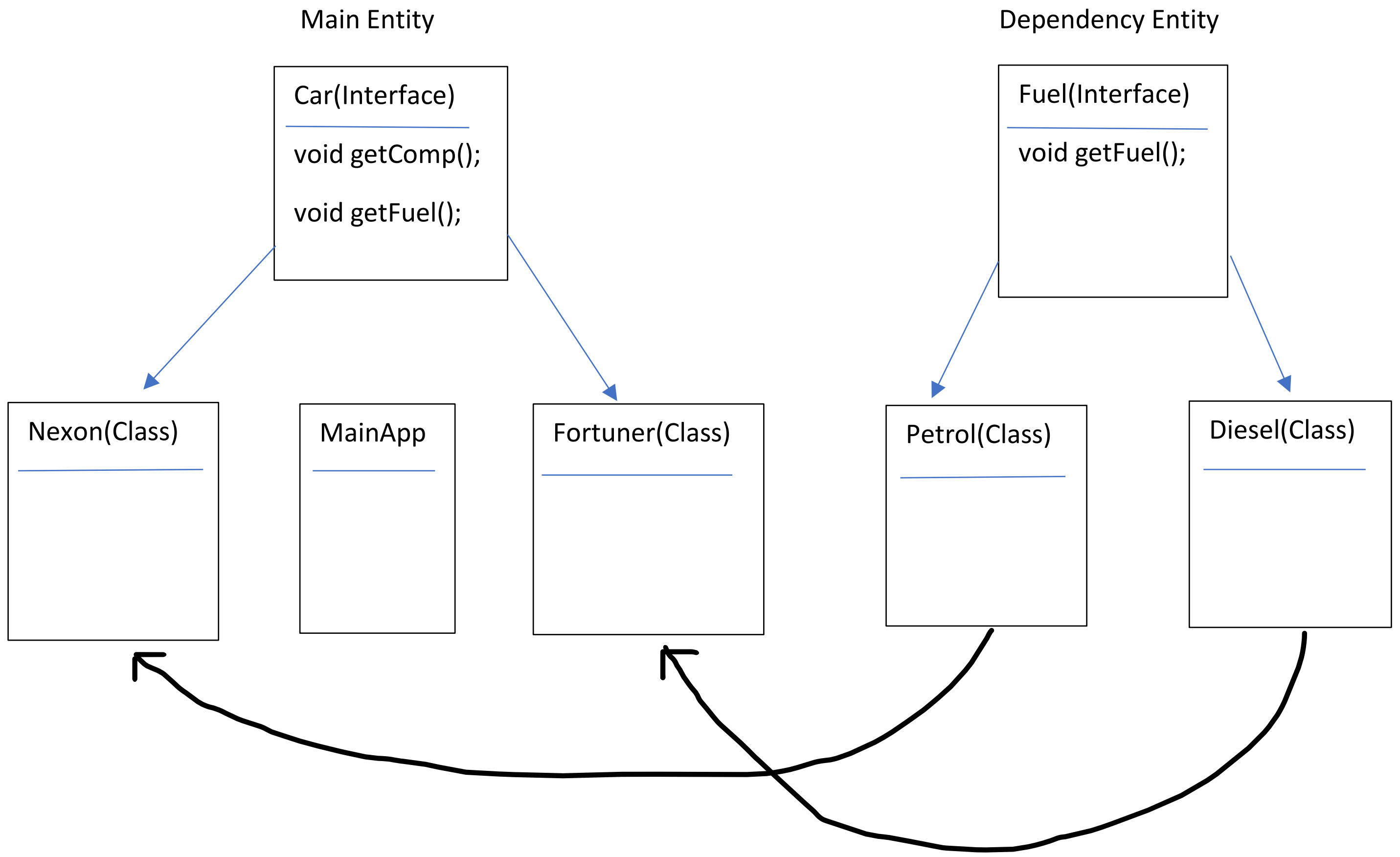
{

}

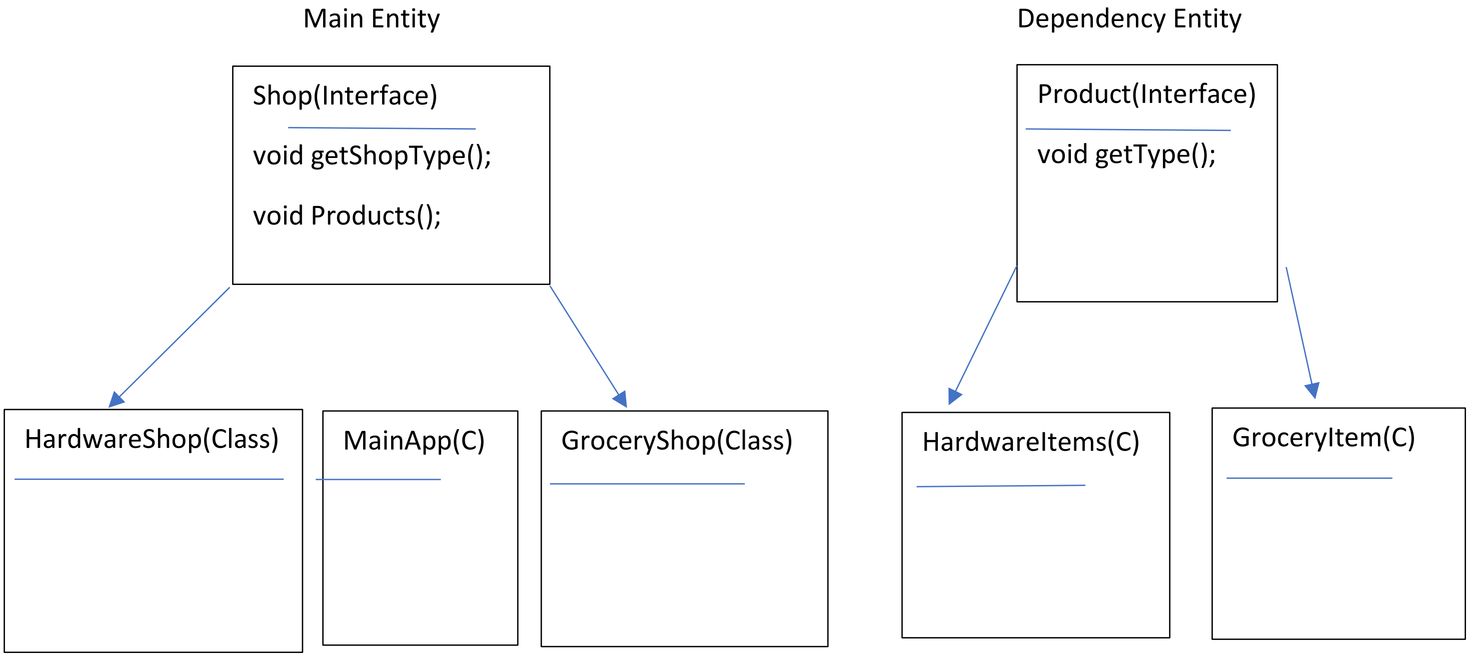
Q. What is Setter Injection?

The process of injecting the dependencies by creating setter method is known as Setter injection. In case of setter injection, we have to declare private reference of dependency object. It is possible to inject mulltiple dependencies by creating setter methods.

Q. Develop spring application by using constructor injection.



Constructor injection	Setter Injection
1. Constructor injection needs to be done through parametrize constructor.	1. Setter Injection needs to be done through setter method.
2. Constructor Injection must be done at the time of object creation.	2. Setter injection can be done after the object creation.
3. Constructor injection is used to inject all the dependencies at the same time (Object creation).	3.Setter injection can be used to inject partial dependencies.
4. Constructor injection increases the code complexity.	4. Setter injection reduces the code complexity.
5. Constructor injection can't override setter injection.	5. Setter injection can override constructor injection.



Q. Explain Spring annotation?

-> In spring framework, configuration can be done by using two ways,

1. .xml-based configuration

2. Annotation based configuration

If you are using annotation based configuration, then we don't have to configure beans in xml file. Spring framework has provided different annotations for inversion of control and dependency injection.

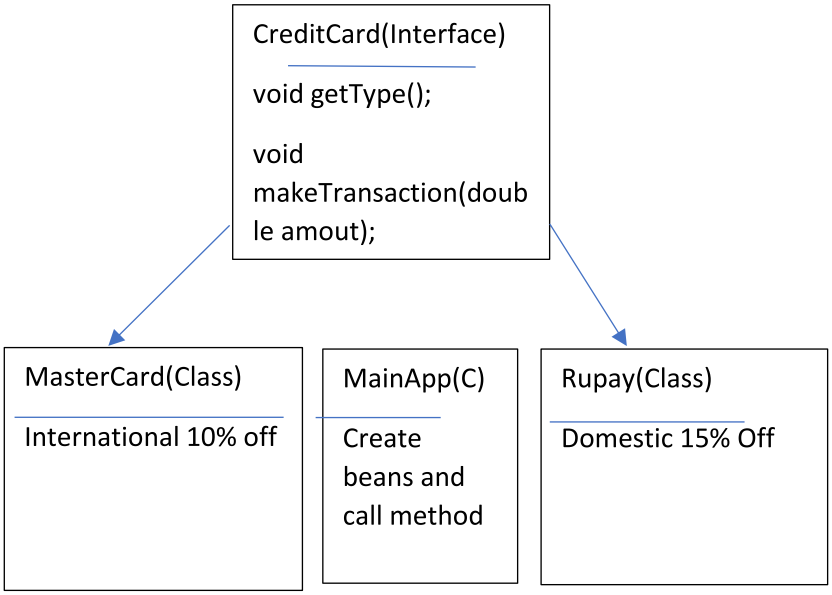
Q. Explain spring IoC annotations?

->In case of IoC, we can configure the bean by using component annotations (@Component). This annotation represents the java class which will be considered as Spring Bean. Following is the syntax we use for component annotation.

1.@Component

2. @Component(“identifier”)

In order to use annotation-based config, first we have to enable context component scanner in applicationContext.xml. After using @Component, we don’t have to add bean tag in xml file.



CreditCard(Interface)

void getType();

void makeTransaction(double amout);

MasterCard(Class)

International 10% off

MainApp(C)

Create beans and call method

Rupay(Class)

Domestic 15% Off

Enter the country name

➔ India/USA-> Mastercard

Display options

1: Mastercard

2:Rupay

Choice

Enter amount:

Spring DI Annotation

Q. How to achieve dependency injection with annotation?

-> 1. As we know that dependency injection can be done either by creating parametrize constructor or else by creating setter method.

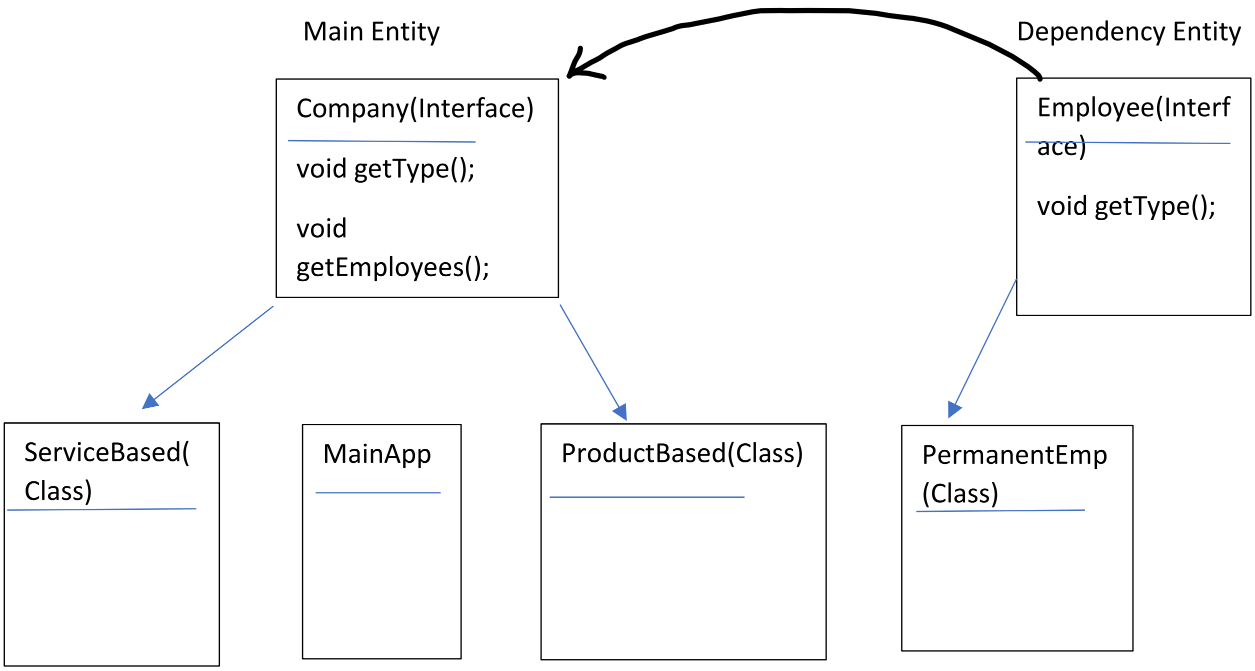
2.If you are using constructor or setter injection then we have to configure particular dependency in applicationContext.xml file.

3.By using dependency injection annotation we can reduce the configuration of dependencies in applicationContext.xml file.

4. To inject the dependency we can use @Autowired annotation.

5. Before using @Autowired annotation, we have to configure dependency class with @Component annotation.

6.By using @Autowired annotation we can inject one or more dependencies.



Q. What is the use of Qualifier Annotation?

-> This annotation is mostly used at the time of dependency injection. If object is having only one dependency implementation, then Spring container automatically injects the dependency with the help of @Autowired annotation. If object is having more than one dependency implementation classes then we have to use @Qualifier annotation to differentiate the dependencies. We can use @Qualifier annotation in setter injection as well as in constructor injection.

Q. What is bean lifecycle?

-> Bean lifecycle is managed by Spring container. When we run the program then first of all spring container will create a bean. After bean creation, the bean will be initialized with its property. After bean initializing, init() method will be executed. After init() method we can use the bean. At a time of closing the container destroy() method will get executed. If we want to execute some code at a time of bean initialization, we can write inside init() method. If we want to execute some code at a time of closing the container, we can write it inside the destroy() method.

BeanFactory(Interface)

ApplicationContext(Interface)

{

ClassPathXmlApplicationContext

AnnotationConfigApplicationContext

FileSystemXmlApplicationContext

}(Classes)

1. Bean Creation
2. Instantiation
3. Init()
4. Use bean
5. Destroy
6. Bean Stop

We can use annotation for bean life cycle method. Those annotations are @Postconstruct and @Predestroy. Both these annotations are part of JavaEE (Enterprise Edition). Since JavaEE was deprecated from Java9 and removed from java 11. We have to add an additional dependency to use this annotation.

(Dependency javax-annotation-api). By default all the annotations are disabled in the spring. To enable these annotations in spring, we have to use following tags in applicationContext.xml file

(Tag <context:annotation-config/>)

This tag will enable all the annotations in spring.x