2 Point StringBuffer and StringBuilder objects are mutable. That means After creating the StringBuffer and StringBuilder also we can modify the contents of StringBuffer and StringBuilder objects.

2 point The methods of StringBuffer and StringBuilder are same like append(), insert, delete, replace, reverse etc...

String class methods:-

startsWith(), endsWith(), length(), size()
replace() etc...

Q) Can we override a static methods?

A) No, we can't. override a static Method.

1) What is the Difference Between String, StringBuilder and StringBuffer.

(2) point Difference Between String and StringBuffer and StringBuilder is that String object is Immutable and StringBuffer and StringBuilder objects are Mutable.

String object is Immutable that means Once the String object is Created then we Modify the Contents of the String object

Example:- String myString = "Hello";

Next we want to append the "Guest" to the Same String as follows.

myString = myString + "Guest";

Then the output will be "Hello Guest". Although we made use of Same object, But Internally a new object was created.

If we use String It will Not Improves the performence. Because for Every time it will Create the New objects.

(1) point String and StringBuffer and StringBuilder all are classes used for Storing the group of Characters.

By using String we can Store the group of Characters. for Storing the

group of characters we have to create the String object. We can create the string objects in two ways.

1) By using new operator
2) Directly By using Double Quotes (" ")

→ String mystring = new String("Hello");
→ String mystring = "Hello";

If we create the String object By using new operator then it will Store that object in the Heap memory.

(Note:- Every Object which is created By using new operator that will allocate/store the memory in Heap memory) the object

If the object is created By using new operator then it will not verify wheather that object is Available in Heap memory or Not. Every time it will create a New String object.

If the object is created by using Double Quotes("") then it will store that Object in ~~Heap Memory~~ String Constant Pool.

Again if we want to create a string Object By using Double Quest("") then

it will verify wheather that object is Available in String Constant Pool or Not. If the object is Available then it will Not Create New object. Just it will gives the Reference to that object. If the object is Not Available in String Constant Pool then it will create a New String object.

③point : StringBuffer and StringBuilder Both are classes used for storing the group of characters. If we want to store group of characters in StringBuffer and StringBuilder then we have to create the StringBuffer and StringBuilder objects.

We can create the StringBuffer and StringBuilder objects By using "new" operator only. If we creates the object By using new operator then it will store that object in Heap memory.

(∵ If we want to store the objects in StringConstant pool then By using "intern()" we can store the objects in "string constant pool").

⑤point : String and StringBuffer classes are Thread Safe that means we can Access only one Thread at a Time.

StringBuilder class is Not Thread Safe that means we can Access Multiple Threads at a Time..

Comparable	Comparator
1) Comparable is an Interface	1) Comparator is an Interface
2) Comparable Interface provides Single Sorting Sequence. That means we can Sort the Collection on the basis of Single Element Such as id, name etc	2) Comparator Interface provides Multiple Sorting Sequence which means we can Sort the Collection on the Basis of multiple Elements like id, name etc
3) Comparable affects the original class. That means original Class is Modified.	3) Comparator doesn't affect the original class. That means actual class is not Modified
4) Comparable provides compareTo() for Sorting the elements	4) Comparator provides compare() for sorting the elements.
5) Comparable Interface is Available in Java.lang Package.	5) Comparator Interface is found in Java.util Package.
6) we can sort the List elements of Comparable type by using Collections.sort(List) method.	6) we can Sort the elements of Comparator type by Comparator.sort(List, Comparator) method.

finally	finalize
1) finally is a Block	1) finalize is a method.
2) Finally Block is used to place an Imparitent code.	2) finalize Method is used to perform Code clean up processing Just Before object is Garbage Collected.
3) Finally Block will Be executed wheather Exception handle or Not. that means Even though exception occurs then finally Block get Executed.	

Garbage collector runs "java.lang.System.gc()" Internally.

Q) What is the Difference Between "==" operator and equals()?

By using "==" operator we can Compare only one object. But

By using equals() we can Compare Multiple objects.

Q) How many ways to Handle the Exceptions?

we can Handle the Exceptions in Two ways

1) By using throws Keyword (Not Recomended)
2) By using try, catch Blocks (Recomended).

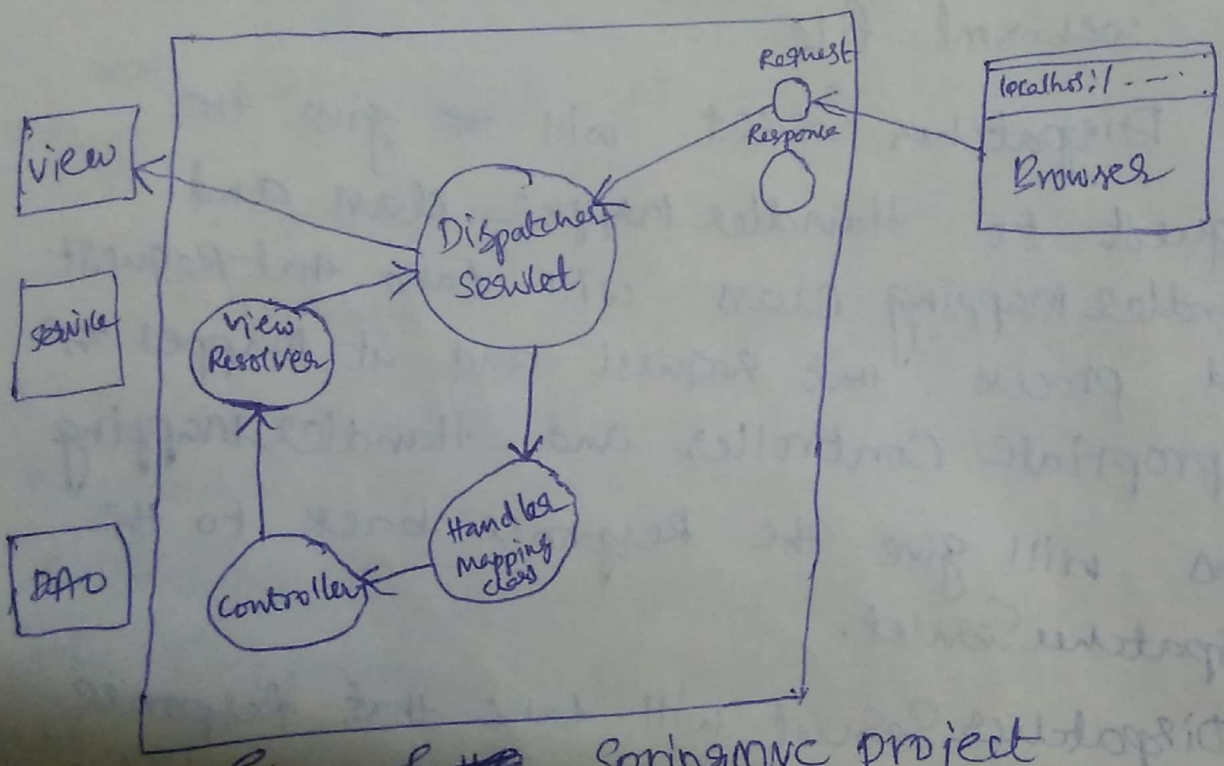
Q) Difference Between throw and throws Keywords?

* throw is a Keyword used to throw the Exception Explicitly. (User defined Exceptions)
If we want to throw any user defined Exception then we use throw Keyword.
* throws Keycoord used to Declare an Exception.
* By using throw Keyword we can throw only one Exception (we can't throw multiple Exceptions)
* By using throws we can throws Multiple Exceptions.

Project

* Currently In my project I am using Spring Framework. So My Project Architecture should be allowed By Spring MVC Framework.
* Whenever the Browser sends a request then DispatcherServlet will take that Request and DispatcherServlet is given By SpringMVC framework and which should be configured in web.xml file.
* DispatcherServlet will give the Request to HandlerMapper class and HandlerMapping class will take that Request and process that request and it mappes the appropriate Controller and HandlerMapping class will give the Response back to the DispatcherServlet.
* DispatcherServlet will take that Response and Instantiates the appropriate Controller.
* Controller will perform the Required logic and Controller will creates the ModelAndView object and Controller will give the ModelAndView object to the DispatcherServlet.

* DispatcherServlet will take that ModelAndView object and given to the viewResolver and viewResolver is given By Springmvc and which should be configured in the Spring-servlet.xml file and it transfer the request to appropriate view Page.
* We write the Business classes Inside the Service-layer
* In the DAO module we write the code to Interact with the Database.



Flow of Springmvc project

HSBC project:-

* Premium deposite Project is a option having Commudity, Normal Deposite and Premium Deposite.
* option is a Contract Between the Two parties without any obligations.

for Ex:- Clint (who are giving) and Counter (who are Receiving)

Q) Tell me about your project:-

I am Currently working in Premium Deposit project which is called as Cross-Currency project.

Normally, In Normal Deposite project the Investors Invest some principle amount of money at a particular Busines date for a tenor (Period means like one year or two years) and Bank will offer some Interest for that amount and at the time of Maturity Date clint will get the maturity Amount. In the Normal Deposite project there is No risk Because Investors should know how much Amount they will get at the time of maturity Date.

In the premium Deposit project risk will Be there and we call the premium deposite project as a Cross-Currency project.

In the premium deposit project Investors Invest their money and it should be Any Currency like INR, Doller, Dinar etc and the Investor Invest their money at perticular Business Date for a tenor (period like one year or two years) and Investors earn the Maturity Amount Based on the current market Rate and it should be any currency like INR, Doller based on the current market rate. then the Investor get either profit or loss Based on the current Market Rate).

Modules of the project:-

1) Deal Booking Module
2) Deal Amendment module
3) Deal Reversal
4) Deal Expiry
5) Deal Reports.
6) Maintainence

1) Deal Booking:- (View)

Deal Booking is view module and In this Deal Booking Module Investors Register/Book New Deposites on Cross-Currency. and In this Investors should be register their Details like how much amount and the tenor and for that Bank will offer some Amount.

Deal Amendments:- (View)

* Deal Amendments is a module used for Modify the details of the Investors and like Investor want to continue their policy for term & and all those Amendment we should do. In Deal Amendments Module.

Deal Reversals:- (View)

Deal Reversal is a Module used for Cancellation of the policies

Deal Expiry:-

By using this Deal Expiry Module we should Know the Expiry date of the policy of the Investor.

Reports:-

By using Reports we can generate the Reports in the form of Pdf and we can generate the ~~Dep~~ Deal information and Deal Reversal and Deal Expiry By using this we can generate the Reports.

Q) List out some Annotations which we use in Springframeworks.

1) @Service 2) @Repository 3) @Component
4) @Autowired 5) @Transactional 6) @Scope
7) @Inject 8) @Resource 9) @Controller
10) @RequestMapping 11) @RequestParam
12) @profile 13) @PreAuthorize 14) @Qualifier

1) @Service:- By using @Service Annotation we can Annotate all our service classes with @service Annotation.

* All the Business Logic we write inside the service classes / service layer.

Ex:-

```
@Service
public class CompanyServiceImpl implements
                                CompanyService {

    // Business Logic

}
```

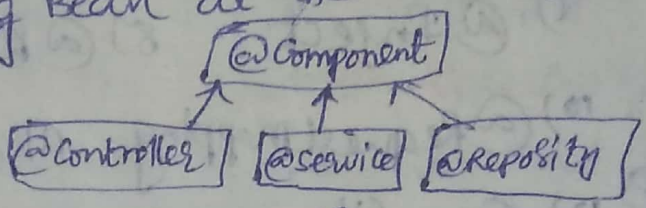
2) @Repository:- we can Annotate all our DAO classes with @Repository Annotation. Inside the DAO classes we write the code to interact with the Database.

Ex:-

```
@Repository
public class CompanyDAOImpl implements
                                CompanyDAO {
    // DAO Logic
}
```

5) @Component :- We can Annotate our Components (like Rest resource classes and DAO Implementation classes ~~or webservices~~).

* By using @Component we can turn the class into a Spring Bean at the time of auto-scan.



Ex:-
```
@Component
public class ContactResource {
    ----
}
```

4) @Autowired :- By using @Autowired Annotation we can make the dependency Between the objects.

* In @Autowired Annotation we can make the Dependence Between the objects By using "byName" and "byType". "byName" means we can establish the Dependency Between the object based on the property Name and "byType" means we can establish the Dependency Between the objects based on the Data Type of the property.

@Resource - Defined in javax.annotation package as a part of Java

@Inject - Defined in Javax.inject Package and part of Java.

@Autowired - Defines the Package org.springframework.bean.factory as a part of Spring framework.

* All @Resource, @Inject and @Autowired all are same.
* @Resource Annotation is Available in JSR220
* @Inject Annotation is Available in JSR330.

5) @Qualifier- There may be a situation when you create more than one bean of the same type and want to wire only one of them with a property. In such cases we use @Qualifier Annotation along with @Autowired.

Ex1-

```
public class Student {
    private Integer age;
    private String name;
    //Setters & Getters
}
```

```
public class Profile {
    @Autowired
    @Qualifier("student1")
    private Student student;
    ---
}
```

6) @Scope- By using @Scope Annotation we can specify the scope of the Spring Bean like Singleton or prototype.

If we create the container object By using BeanFactory Interface/DefaultListableBeanFactory then container will create the Spring Bean object when we call

getBean() even if the Scope is Singleton (or) prototype and By Default Scope is Singleton.

* If we create the Container object By using Application Context then Container will create the object to all the Spring Beans whose Scope is singleton, and Container will create the objects when we call Container.getBean() then Container will create the Spring Bean objects whose Scope is prototype.

* Container will create the object only one time whose Scope is Singleton and if we call that again the It will not create new object just it will use that existing object.

* Container will create the object for prototype how many time we call getBean() that many number of times Container will create the Spring Bean object. for Example If we call container.getBean() for 4 times the Container will create the Spring Bean objects for 4 times.

@Controller :-

@Controller Annotation used to Mark the class as a Controller.

@Req
in a
URI

@Res
Respo
for @
URI(

and
can
Execu
Get
POST
the

@RequestMapping:- we can use @RequestMapping in a class level or method level to provide URI pattren.

```
@RequestMapping(value="/method0")
@ResponseBody
public String method0(){
    return "method0";
}
```

@ResponseBody can be used to send the String Response for this web request.

for @Requestmapping we can specify multiple URI (Uniform Resource Identifier).

```
@RequestMapping(value={"/method1","/method0"})
```

and By using @RequestMapping Annotation we can specify for which Request it has to Execute eaither Get or POST.

Get → when the user enter the URI and click on enter

POST → when the user send the data By Submitting the Button.

```
@RequestMapping(value="/method0",
                method=RequestMethod.GET)

@RequestMapping(value="/method1",
                method=RequestMethod.POST)
```

By using @Requestmapping Annotation can be used to handle dynamic URIs and we can specify the Regular Expressions for URIs.

```
@Requestmapping (value = "/method1/{id:3}")
@Requestmapping (value = "/method1/{id:[\\d+]}/{name:3}")
```

@PathVariable:- used to map the URI variables to one of the method arguments.

```
public string method0 (@pathvariable ("id") long id ){
  3
```

@RequestParam:- used to retrive the URL parameter and map it to method arguments.

```
public string method1 (@Requestparam ("id") int id ){}
```

@profile:-

@profile Annotation allows developers to register the bean by condition. for Example If we want to register the Bean based on what operating System our Application is running and we can specify the Application is running in development, test, production Environments.

@PreAuthorize :- used to check for Authorization before entering into method.

@PostAuthorize :- used to check for Authorization on the basis of logged rules and etc.

* we can use @PreAuthorize and @PostAuthorize in Spring Security.

Q) Explain Annotations in Springboot and Rest webservices?

Springboot :- @EnableAutoConfiguration and remaining all the annotations are similar to spring

Rest :- @RestController, @RequestParam, @Path, @Get, @Post, @Put, @Endpoint etc..

Q) How to create userdefined Exceptions in Java?

.. Userdefined Exceptions in Java also known as Custom Exceptions. Most of the times when we are Developing an applications in Java, we often feel need to create and throw our own Exceptions. These Exceptions are known as userdefined Exceptions or Custom Exceptions.

In the projects already Exceptions are Available we provide the Implementations for those Exceptions.

Q) How to write Views in a Controller.

* First we Develop the Views.

Yes, it is possible to Develop our own Views Instead of JSP's.

* View is a class which implements the View Interface.

* In the Controller we can use two Methods.

```
@Controller
Public class SimpleController{

  public String methodOne(ModelAndView mav){
      mav.setViewNam("one.jsp");
      mav.setView(new ViewImpl);
  }
```

* If we Specify the setViewName() then Controller will goto the Specific JSP.

* If we Specify the setView() then Controller will go to the View classes.

Q) Which Annotation we use for Rest in SpringBoot.

We use @RestController annotation in SpringBoot

Q) What is the Default Server of SpringBoot?

In the SpringBoot by Default it uses Tomcat server.

Q) Can we use External servers like Weblogic in SpringBoot rather than Tomcat and Jetty? and How would you configure weblogic in SpringBoot?

* Yes, we can Configure External servers like Weblogic and Jboss in SpringBoot for that we have to configure like as follows.

```
@SpringBootApplication
public class Application {
    public static void main(String[] args){
        SpringApplication.run(Application.class, args);
    }
}
```

This is the Basic Configuration of SpringBoot Application and If you want to add Weblogic Server then we have to do the following configuration.

```
@ComponentScan
@SpringBootApplication
Public class Application extends
        SpringBootServletInitializer implements
                WebApplicationInitializer {
}
```

* Mandatory to ~~extend~~ implements WebApplicationInitializer Interface If we want to add ~~external~~ external Servers. like Weblogic and WebSphere etc.
* As we want to deploy our Application as a web Application on the weblogic server go to pom.xml file and change the default Packaging Jar to war.
* After that we have to add the Dependencies of maven-war-plugin in pom.xml.
* After that we have to add weblogic.xml file in the context root (classpath).
* And add another empty xml file dispatcherServlet-servlet.xml file.

Then it should (External Servers) will use in SpringBoot Applications.

Q) In Pom.xml we have both Server Tomcat and Jetty Dependencies then How to Avoid Conflicts?

At this Situation In the pom.xml of we configure Both the Servers Dependencies Tomcat and Jetty and whichever the server we want and we have to add exclusions Tag for another server Dependencies then it will take only one Server like this we can Avoid the Conflicts. For Example. I want Tomcat Server and I don't want the Jetty server then add exclusions Tag to Jetty Server. By Doing this we can Avoid the Conflicts.

Q) what are the Design Pattrens have you used in your project?

Design Pattren is a Solution for Commonly Occuring Problems in a Project.

Ex:- JDBCTemplate and HibernateTemplate.

Spring Framework used lot of DesignPattrens Some of the common DesignPattrens are.

Template DesignPattren: HibernateTemplate, JDBCTemplate
Singleton DesignPattren: Creating Bean with Default Scope
Factory Pattren: Bean factory classes.
Prototype Pattren: Bean Scopes
Adapter Pattren: Spring web and Spring MVC
proxy Pattren: Spring AOP Support.

Front Controller: Spring MVC DispatcherServlet

DAO (Data Access Object) pattern: Spring DAO Support

Dependency Injection pattern: Based on the Scope of the Beans DI will work Internally.

Q) How Dependency Injection and IOC Controller will work Internally? Prototype Design Pattern?

It works based on the Scope of the Beans. If we create the container object By using BeanFactory Interface the Container will create the spring Bean object when we call getBean() then only Container will create object to Spring Bean even if the Scope is Singleton or Prototype.

If we create the container object by using ApplicationContext Interface then Container will create all the SpringBean objects whose Scope is Singleton and Container will not create the objects to the beans whose Scope is prototype and when call getBean(). the Container will create the object to the Beans whose Scope is Prototype. If the Scope is Singleton the Container will create only one object and if we call again that object it will use the existing object. Incase of prototype number times we call the getBean() that many Number of objects will create the Container.

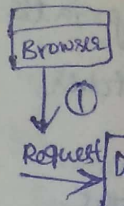
Q) Ex

I

work

Respo

Appli



Wh

Brows

Reque

Spring

Confi

The

to t

class

appr

back

Di

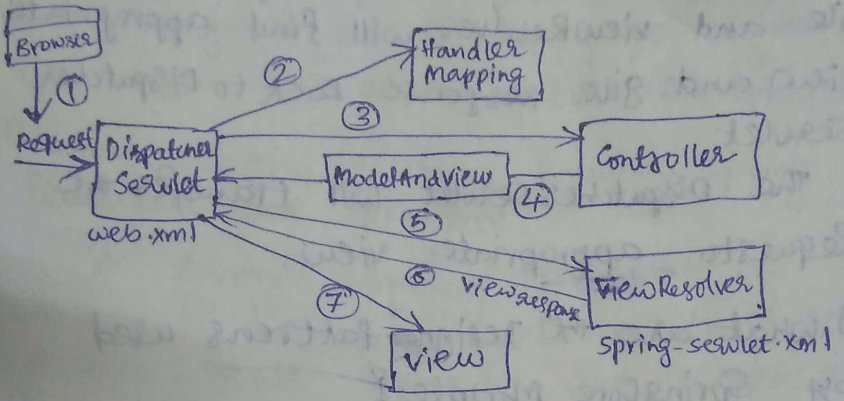
Cont

I

Cont

Q) Explain MVC flow in your project?

In spring web mvc, DispatcherServlet class works as the front controller, and it is Responsible to manage the flow of Springmvc application.



Whenever the request Comes from the Browser DispatcherServlet will take that Request and DispatcherServlet is given by springmvc framework and which should be Configured in the web.xml file.

Then DispatcherServlet will give that Request to HandlerMapping class and HandlerMapping class will process that Request and finds appropriate Controller and give the Response back to DispatcherServlet.

DispatcherServlet will Instantiate the Controller.

In the Controller we write Real code and Controller will Creates the ModelAndView

object and Controller will give ModelAndView object back to DispatcherServlet.

DispatcherServlet will given that ModelAndView object to ~~Handler~~ ViewResolver and which should be configured in spring-servlet.xml file and viewResolver will find appropriate View and give response back to DispatcherServlet.

The DispatcherServlet will transfer the Requesto appropriate view.

Q) what are the Design Pattrens used by SpringCore Module?

Springcore use, somany Design Pattrens and some of themare. (beans scopes)

1) Singleton DesignPattren
2) Prototype DesignPattren

Q) what is a Framework and what is the Difference between Springframework and Springmvc.

framework is a pieace of Software which contains Solutions for Commonly/ Repeatedly Occuring the problems across Multiple projects.

Spring is a big framework and it Contains a lot of Components. One of these Components in Spring-MVC is one of the Component/Module in Spring and SpringMVC we can Implement our web Application according to MVCDesignPattern.

Q) what is the Difference Between Spring framework and SpringBoot?

In Normal Spring framework if we want to develop web Applications then we have to use Spring MVC and in that for deploying our project we have to Configure Servers like Tomcat or weblogic and it takes More Amount of Time and

Coming to the SpringBoot we NoNeed to Configure External Servers and SpringBoot having Embeded Servers like Tomcat and By using main method we can Run our web Applications very quickly and easily and Internally Spring starts the Tomcat Server and it Simplifies the Development allwat

Q) What is Dependency Injection?

Dependency is Nothing But If one object is Dependent on another object then we call that as a Dependency Injection. and we can Establish the Dependency Between the objects in two ways

1) Setter Injection
2) Constructor Injection.

* If Setter Injection is responsible for establishing the ~~Depency~~ Dependency Between the objects then we call that as a setter Injection.
* If Constructor Injection is responsible for establishing the Dependency Between the objects then we call that as a Constructor Injection.

∴ If we have No mandatary fields that means If we have zero parameterized constructor then use — Setter Injection.
If we have mandatory fields that means No zero parameterized constructor and all are parameterized Constructors then use — Constructor Injection.

@Component-Scan - used to process all the Spring core - related annotation and it will Scan all the Annotations in base packages.

Q) What is Autowiring and How do we Establish Autowiring in Spring?

for Example we have Two Beans and they are ~~Dependent on~~ Product Bean and Customer Bean and Product Bean is dependent on Customer Bean In that Situation we can Establish the Dependency Between those objects in two ways.

1) manual wiring 2) Autowiring.

* If we (Developer) Establish the Dependency Between those two objects then we call that as a manual wiring.

* If Container Establish the Dependency Between those two objects then we call that as a Autowiring.

(Note:- If manual wiring is Enable then Autowire will Not work).

* We can make the Dependency by using Autowiring in "byName" and "byType"

* Autowire "byName" means container establish the Dependency Between the objects based on the BeanName.

* Autowire "byType" means Container establish the Dependency Between the objects based on the Data-type of the property.

Q) what is the difference between Authorization and Authentication?

Authentication:-

Authentication means, checking wheather user is valid or not by accepting certain Credentials like UserName and Password.

Authorization:-

Authorization is the process of allowing an Authenticated users to access the resources by checking wheather the user has access rights to the System.

Q) what is the difference between Normal webservices and Microservices?

* Webservice is a way to expose the functionality of an application to other application with out any user interface. It is a Service which exposes an API over HTTP.

* Web Sesvice allow applications developed in different tecnologies to Communicate with each other through a Common farmats like XML, Json etc.

* Webservices are not tied with any operating System or any programing language For Example an application developed in Java can Communicate with C#, Anroid etc or vice versa

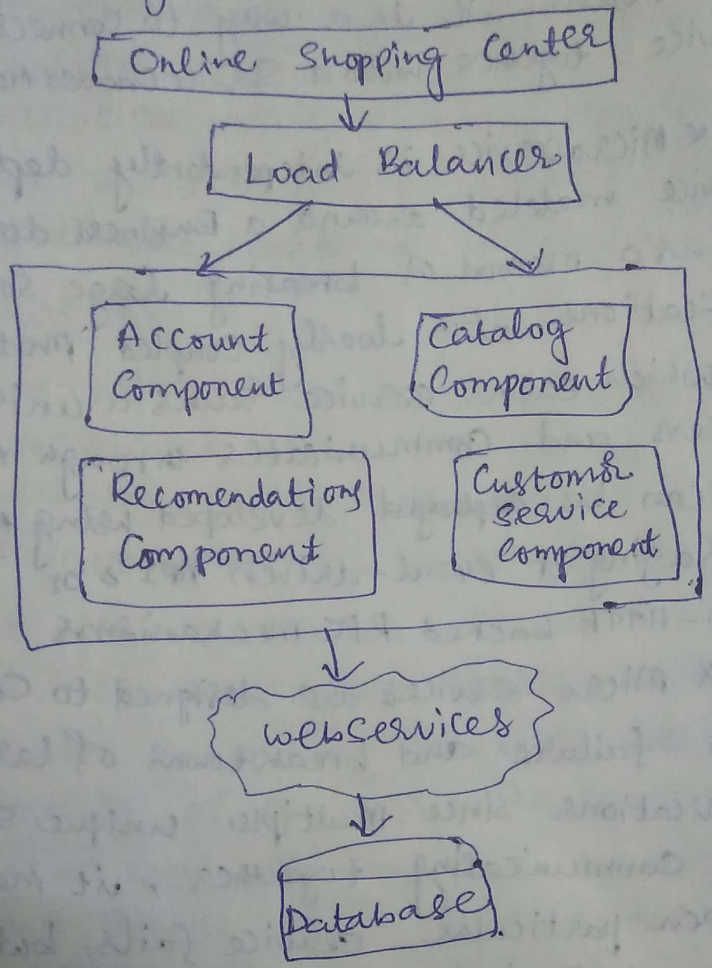
* Web Service is a Connection Technology, that means it is a way to connect Service together into a Service Oriented Architecture (SOA).

* Microservice is independently deployable service modeled around a Business domain. It is a method of breaking large software Applications into loosly Coupled Modules. in which each service runs a unique process and Communicates through API's. It can be ~~deployed~~ developed using ~~Message~~ Messaging or event-driven API's or using non-HTTP backed RPC Mechanisms.

* Micro Services are designed to Cope with failure and breakdowns of large Applications. Since multiple unique services are Communicating together, it may happen particular service fails, but overall large Applications remain unaffected by the failure of a single Module.

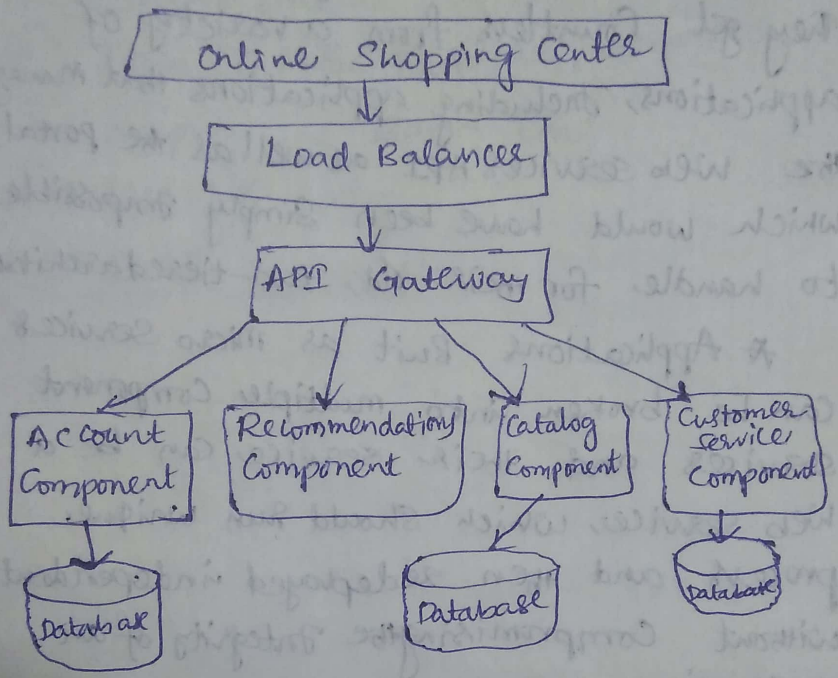
~~fyi~~ Let us understand these Concepts with the help of an example of Online Shopping Center.

Online Shopping Center with Webservices:-



In this Online Shopping Center Web Application is developed in Monolithic Architecture. In this Application there is one Webservice which Communicate with web WebApplication and Database. So this webservice might be performing many functional tasks related to database operations.

Online Shoping Center web Application with Microservices Architecture:-



* Online Shopping Center web Application is developed in Micro Services Architecture. All the Components of the web Application are developed Independently. single functional responsible, fine-grained clearly scoped Services.
* web Services Could be of any size, Including large enterprise apps retrofitted with APIs that to many other Apps depended on. Although "micro" in Micro Services. The Basic Concept is that each Service performs a single function.

for Example, one of the largest eCommerce portals, Amazon, has migrated to Micro services. They get Countless from a variety of applications, Including applications that manage the Web services API as well as the portal, which would have been Simply Impossible to handle for their old, two-tiered architecture.

* Applications Built as micro services can be broken into multiple Component services and their service can be a Web service, which should run unique process and then redeployed independently without Compromising the Integrity of an application.

HTTP - Hyper Text Transfer protocol

HTTPS - Hyper Text Transfer protocol Secure.

HTTP Request fasmat&!-

1. Request Line
2. Zero or more Headers
3. Empty Line / Blank Line
4. Message Body.

HTTP Response formats:-

1. Status Line
2. Zero (or) More headers/parameters.
3. Empty Line / Blank Line
4. Message Body

HTTP Error formats:-

Q) What is the use of Service Layer in Spring?

Service Layer is used for writing the Business Logic.

Ex:- If the Employee Salary = 70000 and he want to add tax and other services

70000 − tax = 7000 − 2000

= 68,000/-

like this Business logic we write inside the Service Layer.

Q) Explain Log4j.xml file Tags?

A)

```
<log4j:configuration>
<appender name="console"
        class="org.apache.log4j.ConsoleAppender">
<layout class="org.apache.log4j.PatternLayout">
<param name="ConversionPattern"
    value="%d{yyyy-MM-dd HH:mm:ss}%5"/>
</layout>
</appender>

<root>
        <level value="DEBUG"/>
        <appender-ref ref="Console"/>
</root>
</log4j:configuration>
```

Q) What are the Difference Between SOAP and Restful Webservices and which one we Recomended and why?

SOAP	REST
1) SOAP is a protocol	1) REST is an Architectural style
2) SOAP stands for Simple object Access Protocol	2) REST Stands for REpresentational State Transfer
3) SOAP can't use REST because it is a protocol	3) REST can use SOAP web services Because it is a concept and we can use any protocol like HTTP, SOAP

4) SOAP inte the B

5) JA API

6) SOA tobe

7) SOA Bandw than

8) SOA Own

9) SOA data

10) SOA than

4) SOAP uses services interfaces to expose the Business Logic	4) REST uses URI to expose Business logic
5) JAX-WS is the Java API for SOAP webservices	5) JAX-RS is the Java API for Restful webservices.
6) SOAP defines standards to be strictly followed.	6) REST does not define too much Standards like SOAP
7) SOAP requires more Bandwidth and resource than Rest.	7) REST requires less Bandwidth and resource than SOAP
8) SOAP defines its own security	8) Rest webservices inherits security measures from the underlying transport
9) SOAP permits XML data format only.	9) REST permits different data formats such as plain text, HTML, XML, JSON etc.
10) SOAP is less preferred than REST.	10) REST is more preferred than SOAP.

Q) Explain Internal working of HashMap:-

* HashMap is used for storing the ~~values~~ elements in the form of Key and Value Pairs.
* HashMap Contains an array of Node and Node can represent a class having following objects:
  1) int hash 2) K key
  3) V value 4) Node next.
* HashMap Internally works Basedon Hashing Process.
* Hashing is a Process of Converting an object into Integer form by using hashcode(). It is necessary to write hashCode() properly for Better performence of HashMap.
* hashCode() is used to get the hashCode of an object.
* hashcode() is used to Calculate the Bucket and Therefore Calculate the Index.

* equals() is used to check that 2 objects are equal or not.

* A Bucket is one element of HashMap array. Bucket is used to store nodes. Two (or) more nodes can have the same Bucket. In that case LinkList structure is used to connect the nodes. Buckets can have different capacity. Relation Between Bucket and capacity is as allows.

Capacity = number of Buckets * load factor

* A Single Bucket can have more than one nodes, it depends on hashCode() method. ~~the better your hashCode() is the better~~

* HashCode of key may be large enough to create an array. hashCode generated may be in the range of Integer and if we create arrays for such a range, then it will easily cause outOfMemoryException. So we generate index to minimize the size of array. Basically following operations is performed to calculate index.

index = hashCode(key) & (n-1)

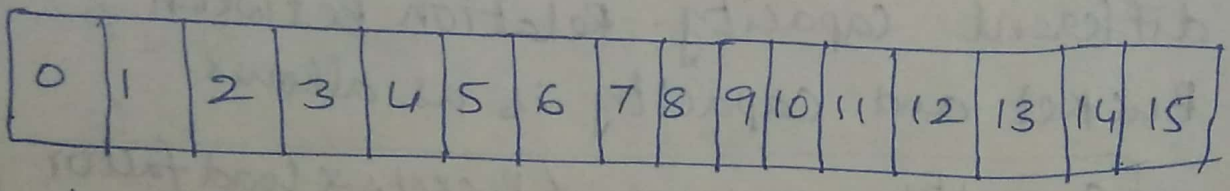
where n is number of Buckets or the size of array. In our Example, I will Consider 'n' as default size that is 16

* Initially Empty HashMap :-

Here, the hashMap size is taken as 16.

```
HashMap map = new HashMap();
```

HashMap :-



* Inserting Key-Value pair :-

putting one Key-Value pair in above HashMap

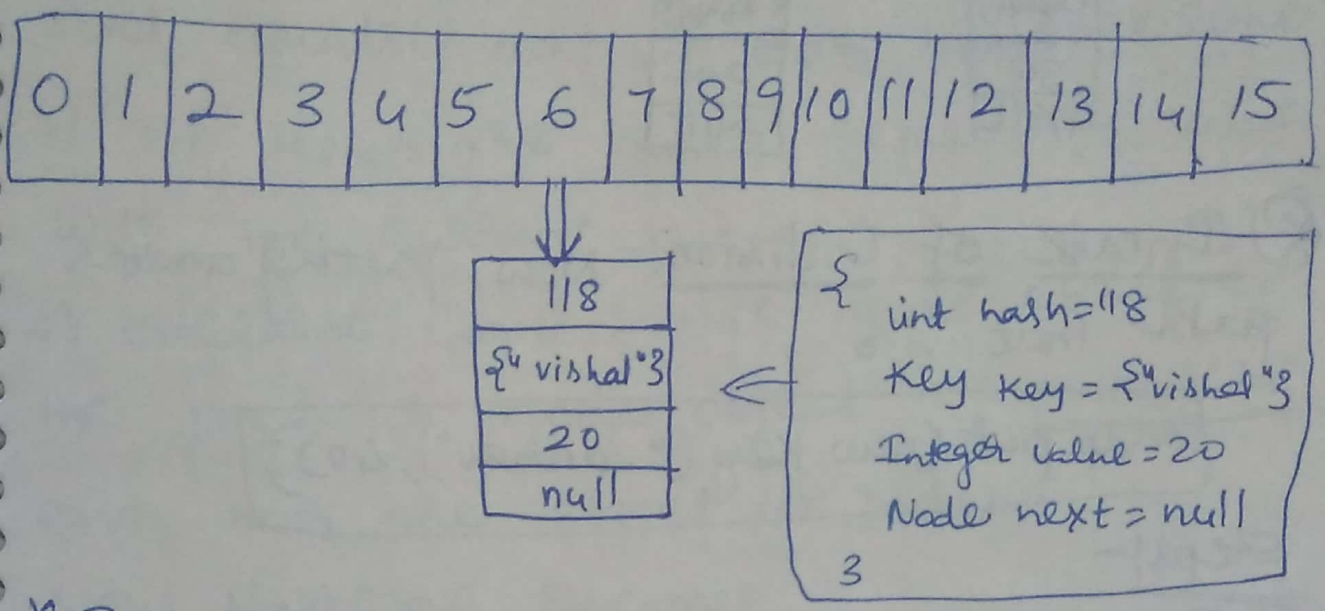
```
map.put(new Key("vishal"), 20);
```

Steps :-

1) Calculate hashCode of Key {"vishal"}.
   By using hasCode finding Mechanism
   It will generated as 118
2) Calculate Index by using Index Method and it will be 6
3) Create the node object

4) place this Node object at Index 6, if no other object is presented here.

Now HashMap Becomes:



*Inserting another Key-Value Pair:-

```
map.put(new Key("Sachin"), 30);
```

Steps:-

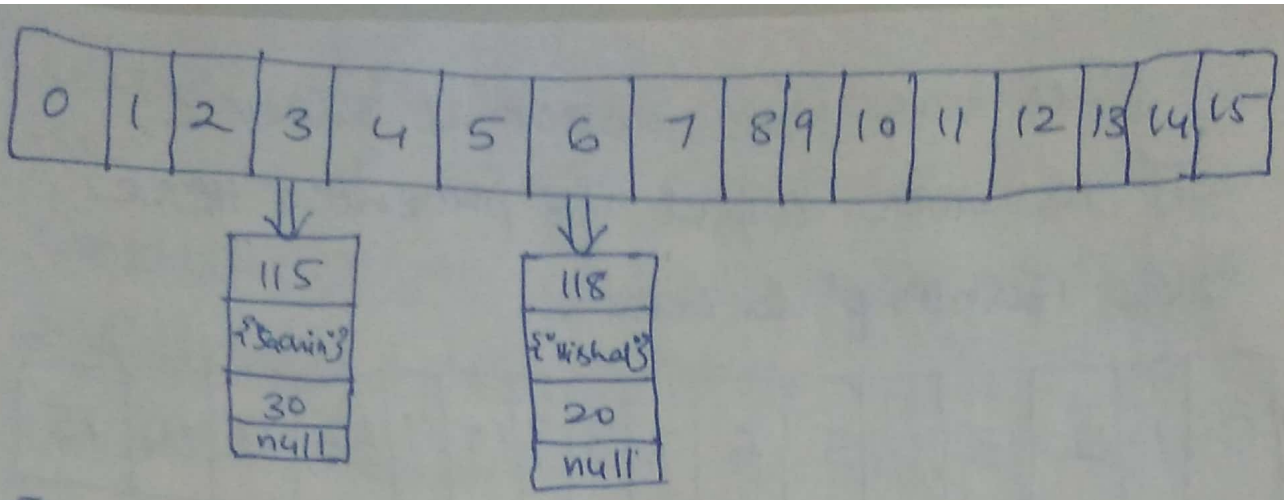
① Calculate hasCode of key {"sachin"}. It will be generated as 115.

② Calculate Index By using method it will be 3

③ Create the node object

```
{ int hash = 115
  Key key = {"Sachin"}
  Integer value = 30
  Node next = null
}
```

④ place this object at Index 3 if no other object is presented here Now HashMap Becomes.



(*) Incase of Collision:- Now, putting another pair that is

```
map.put(new Key("vaibhav"), 40);
```

Steps:-

① Calculate hashCode of Key {"vaibhav"} It will be generated as 118.

② Calculate Index By using index() and it will be 6.

③ Create a node object as:

```
{
  int hash=118
  Key Key={"vaibhav"}
  Integer value=40
  Node next=null
}
```

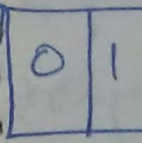
④ place this object at Index 6 if no other object is presented here.

5) fo

6) I and

7) I with

8) C the and Now



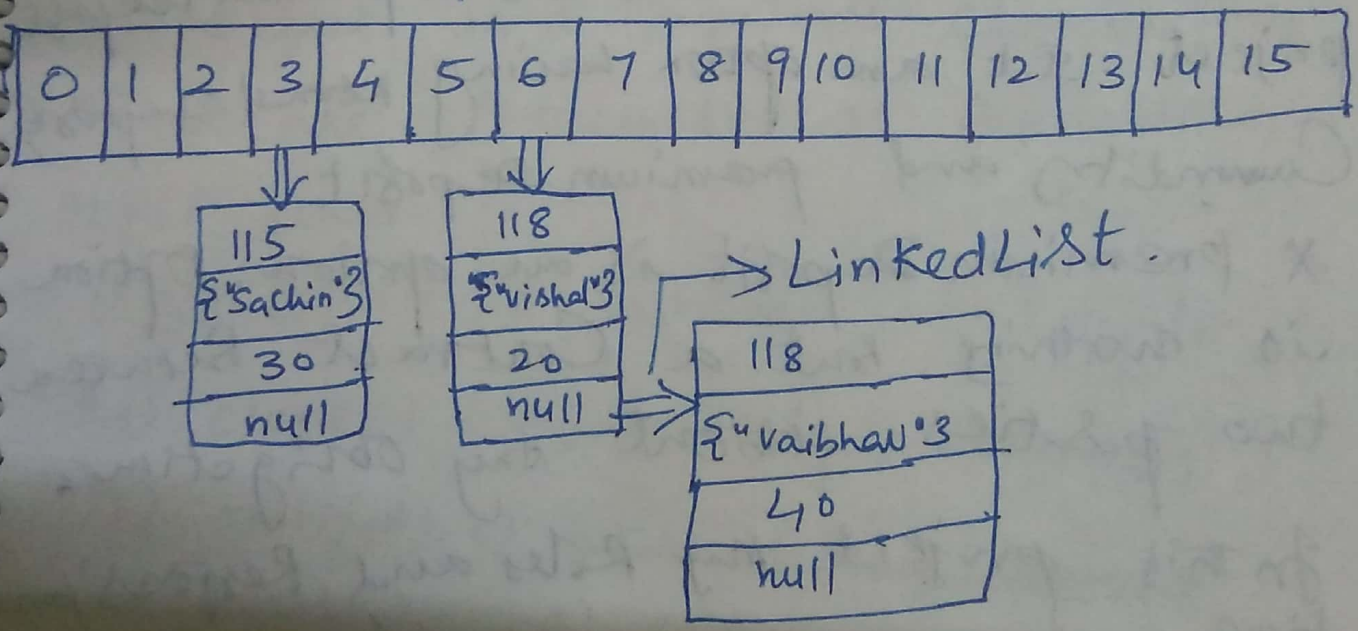
5) In this case a node object is found at the index 6 — this is called as Collision

6) In that case, check via hashCode() and equals() that if both keys are same

7) If keys are same, replace the value with the current value.

8) otherwise Connect this node object to the previous node object via LinkedList and both are stored in index 6.

Now HashMap Becomes:



Self-Introduction

Hi Good Morning/Good Afternoon
This is Kameswara Rao and Currently
I am working with Tata Consultancy Services
as a Software Engineer and As of Now
I am having around 2 years of experience
with Java/JEE Technologies and the Frameworks
like Spring, Spring Boot, Hibernate and Web services
As of Now I worked for Different Clients
like HSBC in that it is a premium Deposit
project. It an option having Normal Deposit
Cummdity and premium Deposit.
* premium Deposit is an option. Option
is nothing but a Contract Between
two parties without any Obligations
In this project my Roles and Responsibilities
is to Develop the Server side
programming and Developing the Business
Service classes and Controllers and
Developing the Validations and
fixing the Bugs and providing Tech Support
to my co workers.

Q) Su
at a
it i
Exec
this
A) B
Boot
Q) H
clas
A)
The
Objec
1)
4)
7)
8)
Q)
an
A)
Hy
ca
a

Q) Scenario-
If I want to Execute 100000 Records at a time By using Spring Boot But it is taking half-an-hour time for Executing 100000 Records then How to Resolve this Issue?

A) By using @Async Annotation in Spring Boot we can Resolve this Issue.

Q) what is object class and How Many Methods present in object class?

A) object is a Supermost class in Java There are 9 methods are Available in object class.

1) getClass() 2) hashCode() 3) equals()
4) clone() 5) toString() 6) notify()
7) notifyAll() ~~8) wait(long timeout)~~
8) wait() 9) finalize()

Q) what is the Difference Between wait() and sleep() in Threads?

A)

wait() is used for waiting the threads and it will wait untill it get called by notify().

sleep() is used to sleep the execution of a thread untill certain Amount of Time.

Q) What is BatchExecution in JDBC?

A) BatchExecution is a process of Executing more Number of Queries into a single Unit is called BatchExecution.

Q) Scenario1-

for Example I have a Bank Customer and for that Bank Need to maintain the Minimum Account Balance is 5000 and The user has withdrawn 500 for a Day then when the Batch will Execute?

For Every Transactions perticular Query will be Executed and that will sent a message to the user Mobile Number and Email. But

BatchProcedure will Execute Every Day mid Night it will Execute.

Q) What is Angular Js?

A) Angular JS is a Javascript framework By using Angular we can Design the Applications in the form of MVVM Architecture (Model/ViewViewModel)

In Angular we Do not have Controller.

SVN

Q) What is Local Server?

A) A Server we have Setup on our Current Machine (Local Machine).

Q) What is Remote Server?

A) A Server that is Setup in Some other System or Machine.

Q) What is SVN?

SVN is a Version Control System is a Software, that helps the Developers to work togeather and maintain a Complete history of their work.

the major Advantages of SVN are,

* Allows the Developers to work togeather
* Do Not override each other changes
* Maintain history of Every version of Everything.

Q) What is Repository in SVN?

Repository is the place where developers store all their work. Repository not only stores files But also the history.

Q) What is Trunk! - (Production code will be Available in Trunk)

Trunk is a Directory where all the Main Development happens and is usally checkdout by the developers to work on the project

Q) What is Tags! - (Released code is Available in Tags)

Tags is a Directory is used to store named Snapshots of the project.

Q) Branches:- (Development code Available in Branches)

what is Branch is an Operation used to create another line of Development.

Q) what is Workingcopy:-

Workingcopy is a Snapshot of the Repository. The Repository is Shared by all the teams, But people do not modify it directly.

8) Different Commands in SVN:-

* checkout - checkout is used for getting the Code from Repository to local Repository.

* Checkin/ Commit - Checkin/ Commit is used for putting our code to Repository

* update/ merge - If we Developed some code and that code is not mached with the Repository code then By using update/merge we can Synchronize our code with Repository.

* Revert - By using Revert if we lost any code then By using this Revert we can get back that code

* History - we can Show the History

* Diff - we can find the differences of two files.

Q) What are the Different Resource formats in REST?

A) There are 4 Resource formats Available those are ① GET ② PUT ③ POST ④ Delete

① GET:- GET is used to Retrive some Resources (Data)

② PUT:- PUT is used to update the Resource

③ POST:- POST is used to create the New Resource.

⑥ Delete:- Delete is used to Delete the Resource.

Q) How Don you map your Service with the WebService (Either REST/SOAP)?

A) By using URI (Uniform Resource Identifier) we can map the webservice with the URL.

Q) What are the Idempotent Resources in webservices?

A) Idempotent is an Important Concept in the HTTP Specifications

that states idempotent HTTP requests will result in the same state on the server no matter how many times that same request is executed.

GET, PUT and Delete all are Idempotent But POST Does Not Idempotent.

Q) Simple Example for JSON?

JSON is a JavaScript Object Nation By using JSON machine and human can understand the JSON farmat Data.

Ex:-

```
{
  "age" : 100,
  "name" : "Java",
  "messages" : ["msg1", "msg2", "msg3", "msg4"]
}
```

Q) How to Read JSON Data through Java?

By using JSONParsar we can Read the JSON Data.

Data Structures:-

Data structure is a Systematic way to Organize the Data inorder to utilize it efficiently.

Algorithem:- Algorithem is a step-by-step procedure which defines a set of Instruction to be executed in a certain order to the Desired output.

Examples of Algorithems:-

* Search - Algorithem to search a item in a Data Structure.
* Sort - Algorithem to Sort items in a certain order
* Insert - Algorithem to Insert an Item in a Data Structure.
* Update - Algorithem to update an Item in a Data Structure.
* Delete - Algorithem to Delete an Item in a Data Structure.

Example for Adding two Numbers Algorithem:-

Step1:- Start ADD
step2:- get values of A & B
Step3:- c ← a+b
Step4:- display C
step5:- STOP.

(Note:- By using TIME factor we can Know the Complexity of the Algorithem).

Reverse

(1) way

pub

(2) way

pub

pu

Reverse of a String Programs :-

① way By using predefined method

```
public class ReverseOfAString {
    public static void main(String[] ar){
        String str = "ABCD";
        S.O.P(new StringBuilder(str).reverse().toString());
    }
}
```

o/p :- DCBA

② way without using predefined method :-

```
public class ReverseString {
    public static void main(String[] ar){
        String name = "ABCD";
        char[] data = name.toCharArray();
        for(int i= data.length-1; i>=0; i--){
            S.O.P(data[i]);
        }
    }
}
```

o/p :- DCBA

Removing the Duplicates in a String

Oway using Predefined Methods

```
Public class RemoveDuplicates {
    public static void main(String[] ar) {
        String name = "AAABBCC";
        char[] data = name.toCharArray();
        Set<Character> set = new LinkedHashSet<>();
        for(char c : data) {
            set.add(c);
        }
        StringBuilder sr = new StringBuilder();
        for(Character cr : set) {
            sr.append(cr);
        }
        S.O.P(sr.toString());
    }
}
```

O/P: ABC

Inseas of Integer we use ~~Int~~ & Integer

Palind

Palindrom Program in:-

```
public class Palindrom{
 public static void main(String[] ar){
   String original, reverse = "";
   Scanner scr = new Scanner(System.in);
    System.out.println("Enter the Data to
                        Check Palindrom");
   original = scr.nextLine();
   int length = original.length();
    for(int i=length-1; i>0; i--){
      reverse = reverse + original.charAt(i);
    }
   if(reverse.equals(original)){
      S.o.p("It is a Palindrom");
   }
   else{
      S.o.p("It is Not a Palindrom");
   }
  }
}
```

Embeddable class in JPA :- ✓

Entity manager in JPA ✓

SpringBoot Initializer

Components in SpringBoot

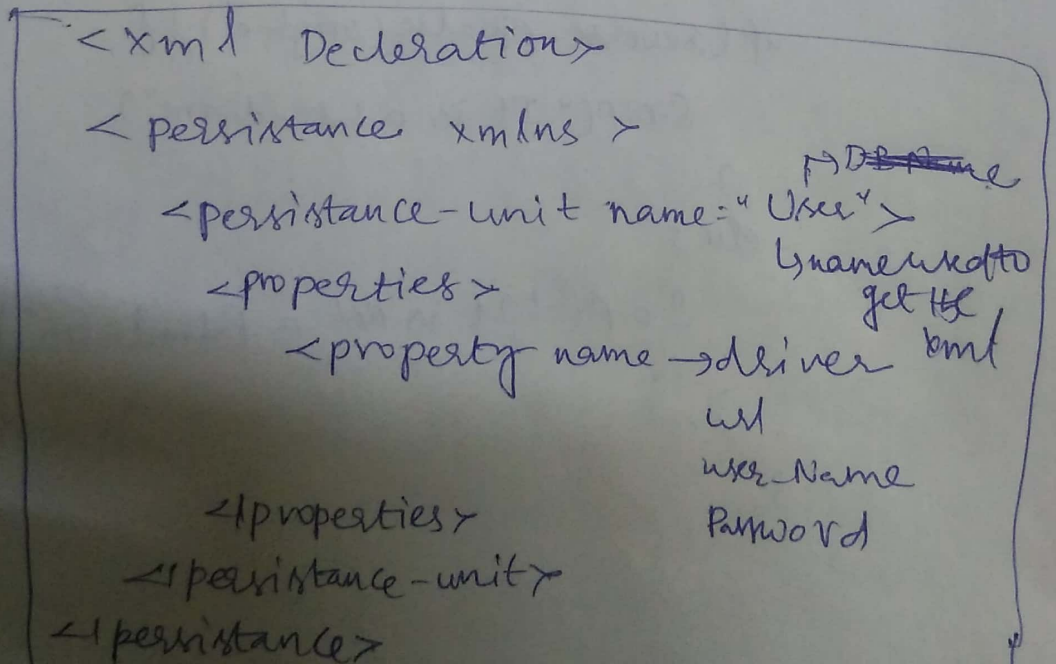
JPA Life cycle ✓

Directives in Angular

Classloaders in Java ✓

JPA Simple Application -

JPA xml file -



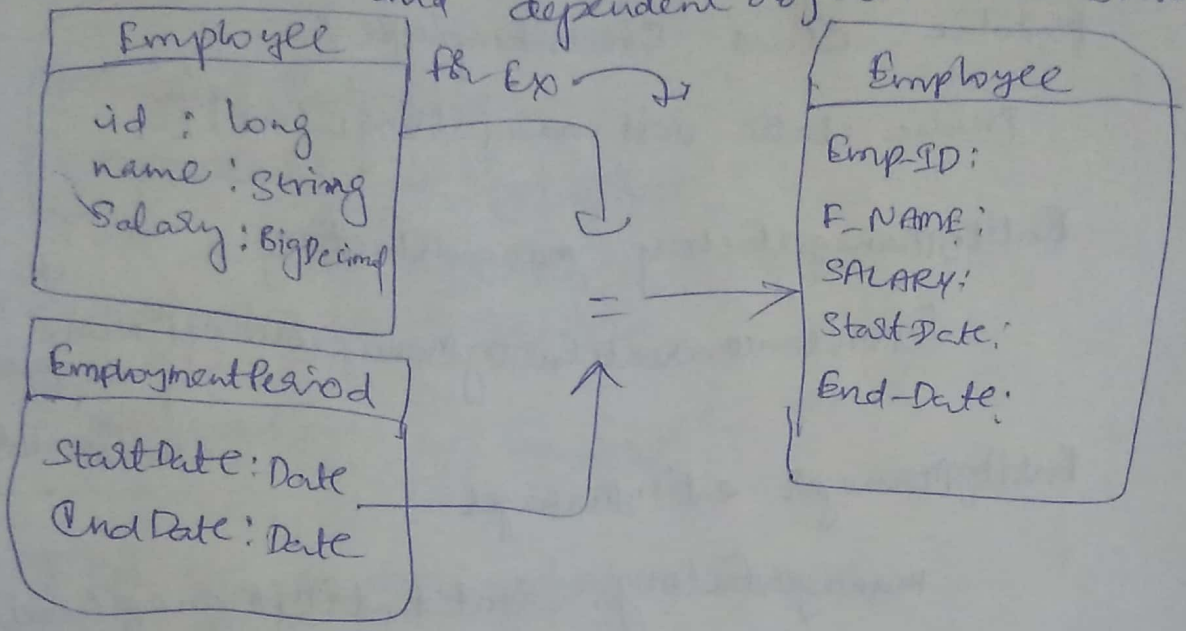
<persistance-unit name=" ">

JPA:-

```
public class CreateEmployee {
    Public static void main(String[] ar) {

EntityManagerFactory managerfactory =
        Persistance.createEntityManagerFactory("name we gave in persistance.xml");

EntityManager entityManager =
        managerfactory.CreateEntityManager();
entitymanager.getTransaction().begin();
    Employee e = new Employee();
    e.setEid(1); e.setEname("Sukumar");

    entitymanager.persist(e);
    entitymanager.getTransaction().commit();
entitymanager.close();
    managerfactory.close();
    3
```

Embeddables in JPA/— In An Applications

Ex/- we have Independent objects and dependent objects are Available




```
@Embeddable
public class EmploymentPeriod {

    @Column(name="start_Date")
    private Date startDate;

    @Column(name="end_Date")
    private Date endDate;

}

@Entity
public class Employee {
    @Id
    private long id;
    ---

    @Embedded
    private EmploymentPeriod period;
    ---
}
```

Ex/-1)

2)

3)

the

the

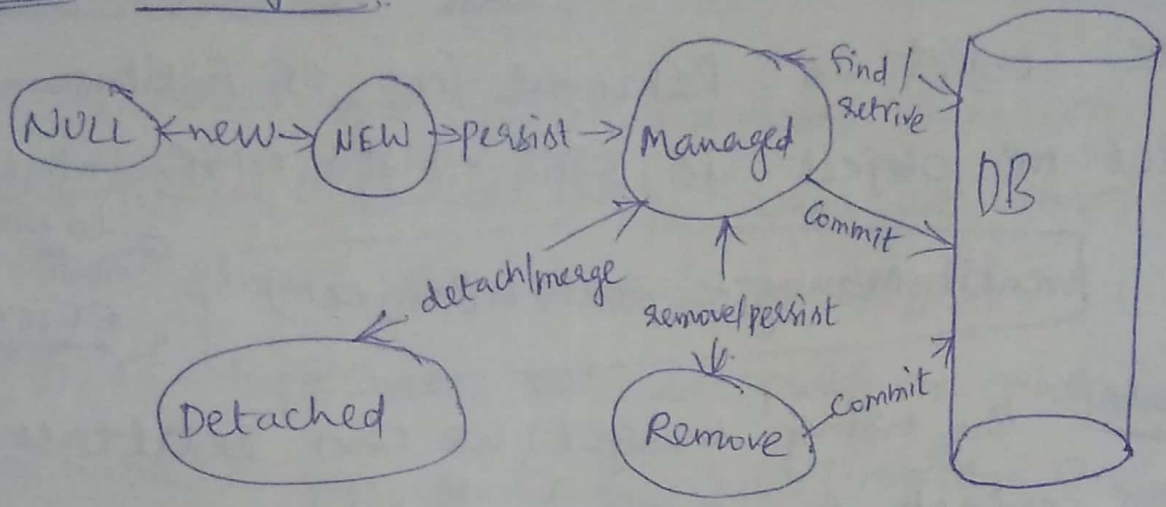
the

objec

be

NUL

JPA Life Cycle :-



Ex:- 1) Object Does Not Exist → This is NULL object

```
MyObject object = null;
```

2) New Object – Not Associated with the EntityManager, and Doesn't exist on Database

```
MyObject object = new MyObject();
```

3) Managed – This is the stage where the object become persisted & Managed By the EntityManager. To do this, we Need to call the persist() from within a Transaction. The object is then persisted to the Database when we Commit() is called

```
entityManager.getTransaction().commit();
MyObject object = new MyObject();
entityManager.persist(object);
entityManager.getTransaction.commit();
```

Detached:- when we call Detach() then the object is Removed from the Entitymanager But the object is still exists on the Database.

`entitymanager.detach(object)` → in Hibernate ↑ evict()

(**Note:-** By using merge() we can reattached the object to Entitymanager)

Callback Methods in JPA Entities:-

1) @PrePersist → @PrePersist is executed Before persist() execute.
2) @PostPersist → @PostPersist is executed After persist() execute.
3) @PreRemove / @PostRemove
4) @PreUpdate / @PostUpdate
5) @PostLoad

Note: If either Pre/Post persist methods/ annotations throw an Exception then the Transaction will be **Rolled-Back**

ClassLoaders in Java:-

We know that Java Programs runs on Java Virtual Machine (JVM). when we Compile a Java class. It Transforms it in the form of ByteCode that is platform and Machine Independent compiled Program and stores that in ".class" file.

After when we want use a class Java Classloader loads that class into memory.

There are three-types of Built in-Classloders in Java:-

1) Bootstrap Class Loader:- It Loads JDK Internal classes. typically loads rt.jar and other core classes for example "java.lang" Package classes.

2) Extensions Class Loader:- It loads classes from the JDK Extensions Directory usually $JAVA_HOME/lib/ext directory.

3) System Classloader:- It Loads classes from the current classpath that can be set while Invoking a program using -cp or -classpath Command line options.

(MVVM)

Directives in Angular:-

Angular templates are Dynamic. When Angular Renders them, it transforms the DOM According to the Instructions given By Directives.

A Directive is a class with @Directive decorator. A Component is a Directive with a Template.

A @Component decorator is actually a @Directive decorator extended with template-oriented features.

Ex of Directive:-

```
@Directive({
    selector: '[appHighLight]'
})
export class HighLightDirective {
    constructor(el: ElementRef){
      el.nativeElement.style.backgroundColor="yellow";
    }
}
```

ElementRef in the Directive's Constructor to inject a reference to the Host DOM element, the Element which we applied appHighLight

Ex of C

Con

that u

Suitab

Struct

Ex:-

Ex of Se

Angu

that a

Injection.

Share

@D

ex

3

Ex of component:-

Component is a special kind of Directives that uses a simpler Configuration which is suitable for Component-Based Application Structure.

Ex:-

```
@Component({
    selector: 'my-app',
    templateUrl: './app.component.html'
    template:
})
export class AppComponent{
    val = true;
}
```

Ex of Service:-

Angular Services are substitutable objects that are wired together using Dependency Injection. You can use services to organize and share code across your app

```
@Injectable()
export class HeroService {
    constructor() {}
}
```

Q) What is the Difference Between Static Binding and Dynamic Binding in Polymorphism?

A) Before going to the Static & Dynamic Binding Let's know the Binding first.

Binding Refers to the Link Between method call and method Definition. as shown Below

```
class A {
    public void methodOne{
        S.O.P("MethodOne of class A");
    }
    public void methodTwo{
        S.O.P("MethodTwo of class B");
    }
}
public class Binding {
    public static void main(String[] ar){
        A a1 = new A();
        a1.methodOne();
        a1.methodTwo();
    }
}
```

Binding

Static Binding :- (compile time Polymorphism) (Method Overloading)

Static Binding is a Binding in which happens during Compilation. It is also called as Early Binding Because Binding happens Before program actually Runs

Static Binding (Early Binding as Shown Below):-

```
class A {
    void method(){
        S.O.P("From class A");
    }
}
class B extends class A{
    @Override
    void method(){
        S.O.P("From class B");
    }
}
public class Binding{
    public static void main(String[] ar){
        A a1 = new A();
        a1.method();
        A a2 = new B();
        a2.method();
    }
}
```

Static Binding

Dynamic Binding :- (Run Time Polymorphism)
(Method Overriding)

Dynamic Binding is a Binding which happens during the Run Time. It is also called as "Late Binding".

```
public class Binding{
    public static void main(String[] ar){
        A a1 = new A(); a1.method();
        A a2 = new B(); a2.method();
    }
}
```

Q) Difference Between Interface and Abstract class?

Interface	Abstract
① By using Inheritance we can Acheive the Multiple Inheritance. A class can implement Multiple Interfaces.	① Multiple Inheritance is not possible A class can extend only one Abstract class.
② Interface is a Collection of Abstract methods	② Abstract class can have Combination of Abstract methods and Concreate methods.
③ Interface Can't have Constructor	③ Abstract Class can have Constructor
④ Interface Can't have final methods	④ Abstract ~~Can~~ Methods Can have final methods
⑤ Interface can contain only Static Variables	⑤ Abstract class can Contain ~~Both~~ Instance and Static Variables
⑥ The members of Interface will be only public	⑥ The members of Abstract class can be either public or non public

Q) Wh
A) R
to R
Modi
Ident
RE
a Re

Q) Wh

and
Garbo
Say
Situ
used
Garb
ab

Q) Wh

the
Data
and
as

Q) What is Resource in REST webservices?

A) REST Server simply provides Access to Resources and REST clint Accesses and Modifies the Resources. Here Each Resource is Identified By URI's.

REST uses various Representations to Represent a Resource where Text, JSON, XML, HTML etc.

Q) What is memory Leak in Java?

This proces is called Garbage Collection and the Corresponding piece of JVM is called as Garbage Collector. Simplifying a bit, we can Say that a memory Leak, in Java is a Situation where Some Objects are not used by the Application Any more, But Garbage Collector fails to Recognize them as unused we call this as a memory Leak.

Q) What is Normalization in Java?

In Database, The proces of organizing the columns and tables of a Relational Database to Reduce the Data Redundency and Improve the Data Integrety is called as Normalization.

Q) ~~What is the use of SerialVersionId?~~

Q) What is SerialVersionUID in Java?

The Serialization Runtime Associates with Each Serializable class a version number, called a serialVersionUID. which used During Deserialization to verify that the Sender and Receiver of a Serialized Object have loaded classes that objects are Compatible with Respect to Serialization.

If a Receiver has ~~loded~~ loaded a class for the object that has different SerialVersionUID then that of Corresponding Sender's class, then Deserialization will Result in an InvalidClassException.

A Serializable class can be declared its own serialVersionUID explicitly by declaring a field named "serialVersionUID" that must be static, final and type 'long'.

Q) Examples of Immutable classes?

String, All the Wrapper classes.

Q) Two properties firstName & lastName Now this class I want to Convert as Immutable class How Do you Do that?

A) First Declare class as final So that from the Sub class we should Not modify that second Declare that firstName and lastName as private and final. and write only getter method & Don't write setter method.

Q) What is the purpose of Immutable class?

A) Ex of Immutable as String class.

String str="hello"; Reusability Because If we create the string object as above the ~~other~~ It will store that Object in the string Constant Pool. So where ever If you create a String object str the reference will pointout to the Same that means Reusability.

Q) pu

hasC

Some

that

Uniq

equal

Two

Q) Who

to Sp

Keywor

~~hourly~~

So, Cu

Improv

they ha

Curren

HashM

Q) purpose of hashCode() and equals()?

hashCode():- Whenever we create object some Integer value will generate for that object. To Identify wheather it is unique or not.

equals():- When you want to Compare two objects then By using equals().

Q) What is ConcurrentHashMap?

HashMap is Not threadSafe. If we want to Synchronized then By using Synchronized keyword we use then Synchronization is not good (Entire thread will lock) Synchronization is much Expensive.

So, Currency package By using this we can Improve the Concurrency. In Concurrency Package they have given some Blocking Technics. CurrentHashMap Improves the Concurrency of the HashMap.

Cognizent:- SDLC

1) Advantages of ~~SOAP~~ REST over SOAP?

In the case of → ~~SOAP~~ we need an IDE

SOAP → Bottomup → top Down Approach
→ WSDL file → Communication will
→ It is heavy wait happen only xml
→ more secure only

REST → Light weight → without IDE we can Develop
→ Everything consider as Resource
→ @Path we have to write with @class
→ We can simply Expose with http protocol.

Cache Machanism in Hibernate:-

Cache:- Cache is the Representation of the Database near to our Application. For Example we have a flipcart.com Now It has to Display Some set of mobiles to the users. Now for Every users that means for Every Request I have to go to the Database collecting all the mobiles and Display the user so It will Increase the Network Traffic that means it will Impact the performance. Now Instead of this

For the First Request We Collect all the set of Mobiles and we can store all the Mobiles Information in Cache Memory. So this will Increase the performance

and Decrease the Network Traffic.

So, In Hibernate we have Two types of Caches are available

1) First level cache - we can called as Session cache

2) Second level cache - we can called as Sessionfactory cache

1) Session cache :-

How many session objects we are creating for each session it will maintain a special memory called as session cache. whenever we send a first Request then it will hit the Database and store that Data in the session memory(cache) If we do the same like operations then It will Not go to the Database Directly it will access from the cache memory like insert, update etc. first level cache will be closed If we close the session object. By default it will enable By the Hibernate

2) sessionfactory cache :-

we have to configure if we want second level cache in our Application in the Configuration By using EH cache, OS cache etc, JBoss cache. Whenever we perform the Database operation then it will hit the Database and store that Data in Globally & locally.

Q) When Did Cache Expire?

First level cache → whenever we close the session

Second level cache → we can specify the cache time in the Configuration file.

HDFC world kamiwdcteo.in
pwi mail pmsql

~~Static~~

sequences - oracle → generate Numeric Value Automatically
autogeneration - mySQL ↗

REST → Communicate heterogeneous Systems

Representational State Transfer

REST → cache the data in Rest
SOAP → we can't cache the Data

JSON → object → JSON API
↳ Employee object

@Consumer

Singleton
Factory → Abstract factory → Abstract classes
↳ Factory method
MVC
DAO
Controller