

->how to introduction our self ;

tell me about yourself?

Good Morning Sir, I am Naveen. Currently I am working with Ojas Innovative Technologies. having 4.6years of experience in Java 1.7, Spring and Hibernate. i am also aware about deployment in Jboss.

My current project is OBS means Ojas Business Suite. it is our internal project. it is web based application.

-> we have to know how to explain our project and clearly know about modules how to tell;

->cover all core java concepts, Collections, Multithreading and Exceptional Handling;

\*\*\*\*\*

**\*\*Maven**

->what is maven;

->Maven is a Project Management Tool. it is used for projectsbuild and dependency.

->Maven project eliminates the adding set of jars in each project.

->what is Build Tool?

->A Build tool take care of everything for building a process. generate the source code,compiles source code, packages compiled code into JAR or ZIP file,install the packages code in local repository,server repository and central repository.

->There are three types of Repositories?

->Local Repositories:means .m2 file in your system.

->Central Repositories: is located on the web. it has been created by the apache maven community itself.

->Remote Repositories:company specific library or custom library.

\*\*\*\*\*

**\*\*\*\*\*Github**

->what is github;

\*\*\*\*\*restful webservice\*\*\*\*\*

\*\*\*\*\*

-> What is web service?

we can communicate with different language and different platform.

-> types of web services?

there are two types of web service

-> soap

-> restful

-> what is soap ?

-> soap stands for simple object access protocol.

-> and it is xml based protocol for accessing web service

-> by using soap you will be interact with other programming language application.

-> soap defines its own security is called ws security. soap is secured compared to restful web services.

-> what is restful webservice?

->it is a Architectural style and not a protocol. here there is no protocols means no rules.

- > Language and platform independent
- > different data formats xml,html,json,text.

-> what is stateless?

- > whenever client request to server it treats a new request by using http protocol.

-> What is Cacheing?

- > caching is a process of storing server response in client side so that client need not to make again and again request to server.

-> Jax-rs Annotations?

- > @get: it get the data from database
- > @Post: it is used for create and insert purpose
- > @put : it Update purpose only we used.
- > @Delete: it is used to delete.
- > @Produces: which format we are given to the server.
- > @Consume: which format we want to get the data.

\*\*\*\*\*Microservices\*\*\*\*\*  
\*\*\*\*\*

-> what is microservices?

- microservices is break our large system into number of independent collaborating components.

\*\*\*\*\*H  
ibernate\*\*\*\*\*

-> what is hibernate;

- > Hibernate is an framework simplifies the development of java application to interact with the database.

- > Hibernate is an opensource and light weight.

- > internally hibernate uses ORM tool means object relation mapping. it is programming technique that maps the object data stored in the database.

- > fast performance means there are two types of cache in hibernate. 1st level cache is enable by default.

- > Hql means hibernate query language it is independent query language. you dont need to write data base specific queries. for example if u are using mysql database but your requirement to change oracle database it will lead problems incase you use hibernate you can resolve that problem.

- > Hibenate framework provides the facility to create the tables of database automatically.

- > Simplifies complex joins means to fetch data from multiple tables is easy in hibernate framework.

-> Hibernate Architecture?

-> The Hibernate architecture includes many objects like persistent object, session factory, transaction factory, connection factory, session and transaction.

- > there are four layers in hibernate
  - >java application layer
  - >hibernate framework layer
  - >backend api layer
  - >database layer

-> Elements of Hibernate Architecture?

Session Factory: > sessionFactory object build through configuration object.

- > Session Factory object holds for second level cache.
- > org.hibernate.sessionfactory interface has one method. that method provides objects to session.
- > one session factory object is heavy weight object. because it stores configuration object and hibernate files data.

Session : > The Session object provides time interval between application starts to end. for example if an application contains many sessions.

- > If we want to perform any db operation in hibernate session should be required. session.save();

Transaction : To perform insert,delete operation we should being a transaction

-> What is LazyLoading in Hibernate?

LazyLoading means improve performance. means in Hibernate 3 onwards lazy loading is enable by default. we no need to write lazy loading is="true" it means not load the child objects when

parent object is loaded.

->Difference between get and load?

get ---	Load ----
1. Return null if an object is not found.	1.Throws ObjectNotFoundException if an object is not found.
2. get method always hit the database.	2.load method doesnot hit the database.
3. it return the real object not proxy object.	3.it returns the proxy object.

-> Difference between Hibernate, ORM and JPA?

Hibernate: Hibernate is java framework. to interact with javaapplication and database.

ORM: ORM is an tool. in internally hibernate uses orm tool. ORM means object relational mapping to map the java objects and data stored in the database.

JPA: JPA is a java persistence API . jpa is an specification it will not perform any operation by itself. it is implemented of Hibernate.

(for example if application uses the java api and implementation of hibernate. if you want to switch hibernate to iBatis it is easy to switch. in other case if you are not use jpa api it is very hard to switch.)

\*\*\*\*\*Jenkins

-> What is Jenkins?

Jenkins means Continuous development and continuous integration.

\*\*\*\*\*Spring\*\*\*\*\*  
\*\*\*\*\*

->what is spring configuration and done with 1 example;

->spring boot annotation;

->how to tell which technologies we are used;

-> What is Log4j?

Log4j is a framework. framework means semi build application means a common code will be already developed.

->Difference between Hashtable and HashMap?

Hashtable is also key and value pair format. Hashtable is Synchronized. means at a time allow only one thread. Hashtable does not allow null key and null value.

HashMap also key and value pair format. HashMap is not Synchronized. HashMap allows one null key and multiple null values.

-> Why we are moving into Spring MVC?

because of easy to work with, flexibility, separation of concern.

-> Explain about Memory Model?

Register is memory area

-> What is JIRA?

JIRA is a Project Management Tool. we can manage the defects, we can manage the test cases, we can manage the sprint, we can manage the project. we can do lot more things in JIRA.

<https://jira.ojas.com/> this is url from jira.

-> what is java streams?

Stream is does not store the elements. it is import java.util.stream;

```
int a[]={2,5,6,7,8,9};
```

we want to find min value in the array.

```
int min=IntStream.of(a).min().getAsInt();
```

we can find min,max,sum and avg.

```
IntSummaryStatistics statistics=IntStream.of(a).summaryStatistics(); it will print all min,max,sum,avg values;  
system.out.println(stastices);
```

if we want to find first three numbers and not allow duplicates

```
IntStream.of(a).distinct.sorted.limit(3).sum();
```

```
l2.stream().sorted(Comparator.comparingInt(EmployeeModel::getPincode).reversed()).limit(3).map(EmployeeModel::getName).forEach(System.out::println);
```

```
l2.stream().sorted(Comparator.comparingInt(EmployeeModel::getPincode).reversed()).limit(3).map(EmployeeModel::getName).forEach(System.out::println);
```

-> How to create a resource in restful web service?

what is marker interface?

if an interface have no methods is called marker interface. we have to check by using instanceof operator that object is belongs to that interface or not.

package example;

interface Permission {

}

public class MarkableInterfaceExample implements Permission{

static void show()

{

System.out.println("showwwwwww");

}

public static void main(String[] args) {

// TODO Auto-generated method stub

MarkableInterfaceExample m=new MarkableInterfaceExample();

if(m instanceof Permission)

{

show();

}

else

{

System.out.println("not permission.....");

}

}

}

-> What is Serialization?

Serialization means converting the object state into bytestream is called serialization. deserialization means converting byte stream into object stream is called deserialization.

-> What is Immutable Object in Java? How we can create our own Immutable class in java.

Immutable class means that once an object is created we can not change the value is called Immutable Class. In java all the wrapper class like Integer,boolean,byte,short and String class is

Immutable.

-> The class Must be declared as final and data member variable must be as final.

-> Regular Expression in Password?

Password validation regexp

-----  
^(?=.\*?[A-Z])(?=.\*?[a-z])(?=.\*?[0-9])(?=.\*?[#?!@\$%^&\*~]).{8,}\$

This regex will enforce these rules:

At least one upper case English letter, (?=.\*?[A-Z])

At least one lower case English letter, (?=.\*?[a-z])

At least one digit, (?=.\*?[0-9])

At least one special character, (?=.\*?[#?!@\$%^&\*~])

Minimum eight in length .{8,} (with the anchors)

Email Validations

-----  
1. return Pattern.matches("^[\_A-Za-z0-9-]+(\\.[\_A-Za-z0-9-]+)\*@[A-Za-z0-9\_-]+(\\.[A-Za-z0-9]+)\*\\\\.\\\\[A-Za-z]{2,}\$",number);

2.System.out.println(Pattern.matches("([a-zA-Z0-9]\*)([@]{1})([a-zA-Z]\*)([.]+)([a-zA-Z]\*)([.]?)([a-zA-Z]\*)", "a11sdddds@ffffammm.com.in"));

////////////////////////////////////Email Code.....////////////////////////////////////

MimeMessage message = sender.createMimeMessage();

MimeMessageHelper helper = new MimeMessageHelper(message);

```
try {
    helper.setTo(email);
    helper.setText("Hi" + ll.getUsername() + "you have succesfully registration");
    helper.setSubject("Mail From Spring Boot");
} catch (MessagingException e) {
    e.printStackTrace();
}
sender.send(message);
```

```
//applicationproperties
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=challanaveen521@gmail.com
spring.mail.password=8985352910
spring.mail.properties.mail.smtp.starttls.enable=true
spring.mail.properties.mail.smtp.starttls.required=true
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.connectiontimeout=5000
spring.mail.properties.mail.smtp.timeout=5000
spring.mail.properties.mail.smtp.writetimeout=5000
```

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-mail</artifactId>
</dependency>
```

//////////////////////////////////////Hibernate.....//////////////////////////////////////

\*\*\*\*\*Spring Boot\*\*\*\*\*

-> Why do we need Spring Auto Configuration?

before spring boot we use spring mvc. in spring mvc we need to configure dispatcher servlet, view resolver, component-scan all these in xml files. coming to spring

-boot we no need to write that much of code.

-> what is an inmemory database?

inmemory database means h2 data base. when it stores the data at start the application and destroy the data when it stop the application.

-> for suppose when we use oracle database we need to install the database and all these. in here there is no need to install in h2 database.

-> h2 is a relational database management system written in java. it is embedded in java application or run in the client or server.

-> just adding the H2 runtime jar into dependencies should be sufficient.

-> what is @RestController?

-> spring RestController annotation is used to create restful web service.

->it contain two annotation

->@Controller

->@RequestBody

-> What is @Controller?

->if you declare @controller it is a controller class.

->that class is front controller and responsible to handle user request and return appropriate response.

-> what is @service

if you declare @service it is a service class.

it holds the business logic.

@service Annotation is a specialization of @component annotation.

-> what is @Repository?

this annotations indicates that class deals with crud operation.

-> what is inversion of control?

inversion of control means order of creation. normally first we create objects and next assign values to that object. but inversion of control is first assigning

the values and next create the objects for particular values.

-> what is @Component

@Component is a class level annotation

\*\*\*\*\*  
\*\*\*\*\*

## DeadLocks

>>if two threads waiting for each other forever this situation is called deadlock.

>>Synchronized keyword is the cause for deadlock

package multithreading;

```
public class TestDeadlockExample1 {  
    public static Object Lock1 = new Object();  
    public static Object Lock2 = new Object();
```



```

public static void main(String args[]) {
    ThreadDemo1 T1 = new ThreadDemo1();
    ThreadDemo2 T2 = new ThreadDemo2();
    T1.start();
    T2.start();
}

private static class ThreadDemo1 extends Thread {
    public void run() {
        synchronized (Lock1) {
            System.out.println("Thread 1: Holding lock 1...");

            /* try { //Thread.sleep(10); }
            catch (InterruptedException e) {} */
            System.out.println("Thread 1: Waiting for lock 2...");

            synchronized (Lock2) {
                System.out.println("Thread 1: Holding lock 1 & 2...");
            }
        }
    }
}

private static class ThreadDemo2 extends Thread {
    public void run() {
        synchronized (Lock2) {
            System.out.println("Thread 2: Holding lock 2...");

            /*try { //Thread.sleep(10); }
            catch (InterruptedException e) {}
        */
            System.out.println("Thread 2: Waiting for lock 1...");

            synchronized (Lock1) {
                System.out.println("Thread 2: Holding lock 1 & 2...");
            }
        }
    }
}
}

////////////////////////////////////

```

-> Java synchronized method?

If you declare any method as synchronized, it is known as synchronized method.

Synchronized method is used to lock an object for any shared resource.

When a thread invokes a synchronized method, it automatically acquires the lock for that object and releases it when the thread completes its task.

(

-> when we use Synchronoization means there is no dependency one thread to another thread. means employee consis  
t have fields salary and awards. means there is no  
dependency.

-> Synchrnoization means allow at a time only one thread. means locking that thread.)

```
class Table{
synchronized void printTable(int n){//synchronized method
for(int i=1;i<=5;i++){
    System.out.println(n*i);
    try{
        Thread.sleep(400);
    }catch(Exception e){System.out.println(e);}
}
}
}
```

```
class MyThread1 extends Thread{
Table t;
MyThread1(Table t){
this.t=t;
}
public void run(){
t.printTable(5);
}
}
```

```
class MyThread2 extends Thread{
Table t;
MyThread2(Table t){
this.t=t;
}
public void run(){
t.printTable(100);
}
}
```

```
public class TestSynchronization2{
public static void main(String args[]){
Table obj = new Table();//only one object
MyThread1 t1=new MyThread1(obj);
MyThread2 t2=new MyThread2(obj);
t1.start();
t2.start();
}
}
```

//

-> Difference between fail fast and fail safe?

Fail Fast:

->When you perform add, update or remove on collection while iterating on it.

It throws java.util.ConcurrentModificationException.

->if you share between two or more threads. one thread is adding another one is iterating on elements it will throw ConcurrentModificationException.

Fail Safe:

->it iterate over the copy of original collection. So it would not throw Exception.

List:in List when we are performing at a time iterating and adding it will throws concurrent modification exception . in this problem overcome to use

Copy write on Array List.

-> 1.CopyWriteOnArrayList.

-> 2.Collections.synchronizedList();

package collections;

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.List;
import java.util.concurrent.CopyOnWriteArrayList;
```

```
public class FailFastList {
```

```
    public static void main(String[] args)
    {
```

```
        //List<Integer> al = Collections.synchronizedList(new ArrayList<Integer>());
        List<Integer> al = new CopyOnWriteArrayList<Integer>();
```

```
        al.add(1);
        al.add(2);
        al.add(3);
        al.add(4);
        al.add(5);
```

```
        synchronized(al)
        {
            Iterator<Integer> iterator = al.iterator();
```

```

        while (iterator.hasNext())
        {
            System.out.println(iterator.next());
            al.add(6);
        }
    }

    System.out.println(al);
}
}

```

Set: in Set when we are performing at a time iterating and adding it will throw concurrent modification exception. in this problem overcome to use

ConcurrentSkipListSet.

1. ConcurrentSkipListSet.

-> 2. Collections.synchronizedSet();

package collections;

```

import java.util.Collections;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.Set;
import java.util.concurrent.ConcurrentSkipListSet;

```

```

public class FailFastSet {

```

```

    public static void main(String[] args) {

```

```

        //Set<Integer> s1=new HashSet<Integer>();
        Set<Integer> s1=new ConcurrentSkipListSet<Integer>();
        ConcurrentSkipListSet<Integer> s2=new ConcurrentSkipListSet<Integer>();
        //Collections.synchronizedList(list)
        s1.add(1);
        s1.add(2);
        s1.add(3);
        s1.add(4);

```

```

        Iterator<Integer> itr=s1.iterator();
        System.out.println(s1);

```

```

        while(itr.hasNext())
        {
            System.out.println(itr.next());
            s1.add(8);
            //System.out.println(itr.next());

```

$$\}$$

## ConcurrentHashMap.

## ConcurrentHashMap.

Program:

package collections;

```
import java.util.HashMap;
```

```
import java.util.Map;
```

```
import java.util.concurrent.ConcurrentHashMap;
```

```
public class FailFastHashMap {
```

```
public static void main(String[] args) {
```

```
Map<Integer,String> h1=new ConcurrentHashMap<>();
```

```
h1.put(1, "aaa");
```

```
h1.put(2,"bbb");
```

```
h1.put(3,"ccc");
```

```
for(Map.Entry<Integer,String> m2:h1.entrySet())
```

 $\{$ 

```
System.out.println(m2.getKey()+"...."+m2.getValue());
```

```
h1.put(4, "ddd");
```

}

```
h1.put(4, "ddd");
```

```
System.out.println("jjjj");
```

$$\}$$

}

////////////////////////////////////

-> Contains method in array list?

suppose we are entering the employee details whether it is avilbe or not by using contains method.

package collections;

```
import java.util.ArrayList;
```

```
package collections;

import java.util.ArrayList;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import java.util.TreeSet;

public class RemovingDuplicates {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}
```

```
package interthreadcommunication;
public class WaitExample {
    private static Object monitor = new Object();
    private static boolean fileProcessed = false;
```

```
public static void main(String[] args) {
    new Process1Thread().start();
    new Process2Thread().start();
}
```

////////////////////////////////////



-> Volatile?

Volatile is related to the visibility of variables modified by multiple thread during concurrent programming. every read or write of volatile variable will be to main memory and not cpu cache.

//

-> Atomic?

When data (typically a variable) can be accessed by several threads, you must synchronize the access to the data to ensure visibility and correctness.

```
package atomic;
```

This class works really well in single-threaded environment, but doesn't work at all when several threads access the same Counter instance

```
import java.util.concurrent.atomic.AtomicInteger;
```

```
public class Counter {  
    private AtomicInteger counter=new AtomicInteger();  
  
    public int getValue(){  
        return counter.get();  
    }  
  
    public int increment(){  
        return counter.incrementAndGet();  
    }  
}
```

```
package atomic;  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
import java.util.concurrent.TimeUnit;  
//import com.infotech.count.Counter;
```

```
public class ClientTest {  
  
    public static void main(String[] args) {  
  
        ExecutorService executorService = null;  
        Counter counter = new Counter();  
        try {  
            executorService = Executors.newFixedThreadPool(2);  
  
            Runnable task1 =()->{  
                for (int i = 1; i <=20000; i++) {  
                    counter.increment();  
                }  
            }  
        }  
    }  
}
```

}

---

because of class name as constructor name thats why we cant override in constructor.

-> What is abstract class? Why a class become abstract?

fuel,ac,power-window methods. but all vehicles are not mandatory to ac and power-window but fuel is mandatory. 100% defination is there is class.

-> why abstract cant be final and why abstract cant be static?

////////////////////////////////////

>Lambda expression is a new and important feature of Java which was included in Java SE 8.

>It is very useful in collection library. It helps to iterate, filter and extract data from collection.

>The Lambda expression is used to provide the implementation of an interface which has functional interface. It saves a lot of code. In case of lambda expression, we don't need to define the method again for providing the implementation. Here, we just write the implementation code.

- >Java lambda expression is treated as a function, so compiler does not create .class file

-> forEach method in java.

forEach() to iterate the elements.it is defined in iterable interface. it is a functional interface.

-> Method Reference in java?

Method reference is used to reference of functional interface.

it is easy and compact of lambda expression. each time when you are using lamda expression refering to method.

## three Types of Method References

>Reference to a static method.

>Reference to an instance method.

>Reference to a constructor.

-> Default method in java.

java provides facility to create default method in interface in java8. default methods we can overrideing.

-> Stram in java?

stream is a new package in java it is `java.util.stream`.

in this package contains classes,interface and enums

stream does not store the elements. operation performed on stream doesnt affect the main source.