Hi Folks,

Please find the questions below.

Capgemini:
++++++++++++++++++++
Core java:
============
what is the difference between jdk and jre (page no: 226)
what happens we don't use jdk, program will compile or not (page no: 226)
what are the methods of Object class (page no: 201)
how Hashmap internally works.(page no: 195)
what is the difference between comparable and comparator (page no: 190)
what is externalization how to achieve this.(page no: 17)
what is externilization.(page no: 17)
JVM memory hierarchy
String pool (page no: 207)
can we use final keyword with String. (page no: 207)
what is final,finally and finalize (page no: 36)
what is static (page no: 70)
where static variables gets stored

____
Spring
==============
what is bean
what is the difference between Bean factory and application context
different scopes of spring

____
Hibernate
================
what is the difference between get and load method
what are the states of hibernate
what are the different types of mapping in hibernate
explain different mapping annotations

____
Accolite:
++++++++++++++++++++++
How to find the last 3rd node form linked list.

How to find the each element count in an array
Why wait, notify, notifyAll are not in Thread class as they are used in threading
what is map in hibernate
what is cascade in hibernate

What is the difference between arraylist and linked list
suppose linked list has infinite loop then how to break that loop

Multithreading :
We have defined 2 methods – 1st one is synchronised method and other one is static synchronised method in a same class. we have threads t1 and t2 so they will access concurrently or not.

Ans: Since t1 is going to access synchronised method which has object lock and 2nd thread t2 is going to access static synchronised method which has class level lock
so, there will not be any issue.
How to sort an array based on the occurrence from greater to lowest no of occurance

Difference between comparable and comparator interface

Method overriding with declaration of variables
Ans : Suppose we have 2 classes. Child class is extending Parent class and both class have variables defined. We have created child object with the reference of parent
then, method will get overridden and child class's method will be called but we don't have anything like variable overriding.
so, Parent class variable will get called since we have Parent class reference and child class variable will be hidden.

NTT Data:
++++++++++++++++++++++
Architecture of your project
MVC Flow
SQL Query: to find the 11th highest salary of an employee and find the manager id of him
Difference between getter and setter injection

How IOC works
Spring scopes

CTS:
++++++++++++++++++++
Can we override main method : No
what is final class
How to iterate Hashmap
Method overriding using public method and protected method
What is prototype scope in spring
How to define controller class and a specific method in spring
How to write Query using HQL
How to define one to one mapping in hibernate
how to define multiple primary keys in hibernate
how to use join annotation in hibernate
Write a code to find the 1st occurance of a character in a string and get the position
How to define a bean instance in spring
what are the types of IOC container
what are the ways to inject bean.

Happiest Minds:
++++++++++++++++++++

Inside ArrayList, using Iterator and removing an object if value matches. Will it compile ?
ANS: will not compile, as concurrent modification will occur and to overcome we use CopyOnWriteArrayList class
What is the difference between Class not found exception and class def exception
Sort employee records based on Id then bases on Last Name without using collections.sort method
What is Marker interface and why do we use
What is serialization and why it is required and how jvm identifies Serial version Id
What is composite key and how to use with annotation
What is RowMapper in hibernate
Write Named query
Write Criteria

What are the scopes in JSP

How to redirect to another page in jsp
What is pageContext in jsp

Brillio:
++++++++++++++++++++++
How HashMap internally works – Pls mention hashcode and equals method work also
What is the contract between hashcode and equals() method
Write a code for to find the no of occurance of each character in a string or sentence
Java 8 features
Write lambda expression any code
what is Functional interface
What is MVC
Bean life cycles
What are the controllers in Spring
What is @Repository annotation
What are the ways to configure spring – (XML, Annotations, Java class)
What are the scopes in spring
What are the levels of cache in Hibernate
How to enable 2nd level cache
What is SOA

DB:
How to sort duplicate records in database using only one table
What is self join
Suppose you have a table called Employee which has empId, empName, MaanagerId. then how will you find the manager name.

DXC Tech:
++++++++++++++++++++++
What is the difference between Hashset and HashMap
How HashSet internally works and why does not contain duplicate values
Ans: HashSet internally using HashMap. When we insert any object as value using add() method then it creates HashMap and insert object as Key of HashMap and value as PRESENT

(constant)
OOPS concepts
Can we call a constructor from a constructor
Ans-Yes (using this() in same class and using super() in super class)
Difference between Abstract and interface
How many ways we can create Thread (3- ways - using Reflection API)
Write a code for palindrome
Write a code for reverse a string
What is bubble sort
what is ORM
Suppose we have 2 files, read the date from each file and find the date when they match after reading the file
Write a code for reverse the string and sort it into ascending order
How we can terminate a thread

L&T :
++++++++++++++++++++++++++++
Write a code for fibonacci series
How to iterate Map
What is concurrenthashmap modification
what is fail safe and fail fast
what is cloneable
what is serialization
JVM automatic provides serial version Id then why do we need to define it manually
what are the types of container in Spring
what is lazy loading in spring
difference between arraylist and linked list
difference between arraylist and vector
difference between comparable and comparator
Which all design patterns u know
what is self join
what are types of cache in hibernate
what is @component, @repository, @controller, @service
Write a hibernate query to save a record
what are the states of hibernate

==========
Hashmap vs Hashtable

1. HashMap is non synchronized. It is not-thread safe and can’t be shared between many threads without proper synchronization code whereas Hashtable is synchronized. It is thread-safe and can be shared with many threads.
2. HashMap allows one null key and multiple null values whereas Hashtable doesn’t allow any null key or value.
3. HashMap is generally preferred over HashTable if thread synchronization is not needed

Why HashTable doesn’t allow null and HashMap does?
To successfully store and retrieve objects from a HashTable, the objects used as keys must implement the hashCode method and the equals method.
Since null is not an object, it can’t implement these methods.
HashMap is an advanced version and improvement on the Hashtable. HashMap was created later.

```
class Ideone
{
public static void main(String args[])
{
//----------hashtable ------------------------
Hashtable<Integer,String> ht=new
Hashtable<Integer,String>();
ht.put(101," ajay");
ht.put(101,"Vijay"); // hash table does not allows
duplicate key but values allowed
ht.put(102,"Ravi");
ht.put(103,"Rahul");
System.out.println("------------Hash
table-------------");
for (Map.Entry m:ht.entrySet()) {
System.out.println(m.getKey()+" "+m.getValue());
}

//---------------hashmap--------------------------------
HashMap<Integer,String> hm=new HashMap<Integer,String>();
hm.put(100,"Amit");
hm.put(104,"Amit"); // hash map does not allows duplicate
key but values allowed
hm.put(101,"Vijay");
hm.put(102,"Rahul");
System.out.println("----------Hash map----------");
```

```
for (Map.Entry m:hm.entrySet()) {
System.out.println(m.getKey()+" "+m.getValue());
}
}
}
```

==========
The contract between equals() and hashCode() is:
1) If two objects are equal, then they must have the same hash code.
2) If two objects have the same hash code, they may or may not be equal.
==========

++
Sapiens:

DB:

What is the difference between statement and prepared statement
How to achieve tuning of DB
How to design a database including PK and FK
What are the types of joins in DB
Suppose we have a Employee table contains EmpName, EmpId, Salary, DeptId, ManagerId and Dept table contains DeptId, Salary, deptName. How to find the Manager Name
How to write a join query
How to debug the procedure
How to execute the procedure by command and what are the commands we use
how to call a procedure
Which Oracle version you are using

Java:

What is the Serialization and SerialVersionUID
Suppose i don't want to serialize few variables then how to achieve that
What is exception and what are the types of it
What is the difference between Synchronised block and synchronised method
can we create an object without a constructor
How hashMap works and can we use Object as Key

How to find the hash value of a null key and do we get NPE or not
What is connection pooling and how to achieve it
What is the difference between 1st Level cache and 2nd level cache
What is memory lick and how to overcome this
What is dependency injection in Spring and why do we need
What is the default scope of a bean in Spring
What is IOC
What is blocking chain
why do we use Spring framework
What is the difference between jdbc and hibernate framework
How to configure the database in Spring
Difference between stateful and stateless session
How to check the performance of an application
How many ways we can create an object
Can we create an object without calling constructor

UI:

What is Servlet chaining
How to create custom tag
what is Servlet
what are the life cycle of Servlet
How to use Scriptlet tag
what are the tags we have
how to fetch the value of a variable
What are the implicit objects
Draw the architecture of Servlet
When is the init() method called in Servlet

++

All State:

How Array list works internally
How to overcome the multiple inheritance problem in an Interface
How to find the non-repeated char string in an array of string - use Map
Java 8 features
What is Functional Interface, explain
What is the use of default method in an interface in Java 8
What is lambda expression and how to pass parameters inside

it
What is factory design patterns
What are immutable classes and why is String immutable
How to sort employee records based on different parameters
How to configure Hibernate with spring
What is difference between save and persist method in hibernate
What are the different state in hibernate
What is the difference between get and load method in hibernate
What is second level cache in hibernate and how to enable it

++

Accolite:

What are immutable classes and why is String immutable
How to create an immutable class
What is the difference between by name and id in spring
What are different scopes in Spring
If we create 2 id's of a class in Spring config file and scope is singleton then how many instance will get created
Spring bean's life cycle
How to find the non repeated element from an array of list
What is autowire annotation
Multithreading question, suppose we have 2 threads and both are incrementing a counter 3 times then what would be the output

****
*

INFOSYS CORE JAVA INTERVIEW QUESTIONS & ANSWERS.
Question 1. What Is The Purpose Of Assert Keyword Used In Jdk1.4.x?

Answer :

In order to validate certain expressions. It effectively replaces the if block and automatically throws the AssertionError on failure. This keyword should be used for the critical arguments. Meaning, without that the method does nothing.

Question 2. How Will You Get The Platform Dependent Values Like Line Separator, Path Separator, Etc., ?

Answer :

Using Sytem.getProperty(…) (line.separator, path.separator, …)

Question 3. What Is Skeleton And Stub? What Is The Purpose Of Those?

Answer :

Stub is a client side representation of the server, which takes care of communicating with the remote server. Skeleton is the server side representation. But that is no more in use… it is deprecated long before in JDK.

Question 4. What Is The Final Keyword Denotes?

Answer :

final keyword denotes that it is the final implementation for that method or variable or class. You can’t override that method/variable/class any more.

Question 5. What Is The Significance Of Listiterator?

Answer :

You can iterator back and forth.

Question 6. What Is The Major Difference Between Linkedlist And Arraylist?

Answer :

LinkedList are meant for sequential accessing. ArrayList are meant for random accessing.

Question 7. What Is Nested Class?

Answer :

If all the methods of a inner class is static then it is a nested class.

Question 8. What Is Inner Class?

Answer :

If the methods of the inner class can only be accessed via the instance of the inner class, then it is called inner class.>

Question 9. What Is Composition?

Answer :

Holding the reference of the other class within some other class is known as composition.

Question 10. What Is Aggregation?

Answer :

It is a special type of composition. If you expose all the methods of a composite class and route the method call to the composite method through its reference, then it is called aggregation.

Question 11. What Are The Methods In Object?

Answer :

clone, equals, wait, finalize, getClass, hashCode, notify, notifyAll, toString

Question 12. Can You Instantiate The Math Class?

Answer :

You can’t instantiate the math class. All the methods in this class are static. And the constructor is not public.

Question 13. What Is Singleton?

Answer :

It is one of the design pattern. This falls in the creational pattern of the design pattern. There will be only one instance for that entire JVM. You can achieve this by having the private constructor in the class.

For eg.,

```
public class Singleton

{

private static final Singleton s = new Singleton();

private Singleton()

{

}

public static Singleton getInstance()

{

return s;

}

// all non static methods …

}
```

Question 14. What Is Driver Manager?

Answer :

The basic service to manage set of JDBC drivers.

Question 15. What Is Class.forname() Does And How It Is Useful?

Answer :

It loads the class into the ClassLoader. It returns the Class. Using that you can get the instance (“class-instance”.newInstance()).

Question 16. What Is A Marker Interface?

Answer :

An interface with no methods.

Example: Serializable, Remote, Cloneable

Question 17. What Interface Do You Implement To Do The Sorting?

Answer :

Comparable

Question 18. What Is The Eligibility For A Object To Get Cloned?

Answer :

It must implement the Cloneable interface

Question 19. What Is The Purpose Of Abstract Class?

Answer :

It is not an instantiable class. It provides the concrete implementation for some/all the methods. So that they can reuse the concrete functionality by inheriting the abstract class.

Question 20. What Is The Difference Between Interface And Abstract Class?

Answer :

Abstract class defined with methods. Interface will declare

only the methods. Abstract classes are very much useful when there is a some functionality across various classes. Interfaces are well suited for the classes which varies in functionality but with the same method signatures.

Question 21. What Do You Mean By Rmi And How It Is Useful?

Answer :

RMI is a remote method invocation. Using RMI, you can work with remote object. The function calls are as though you are invoking a local variable. So it gives you a impression that you are working really with a object that resides within your own JVM though it is somewhere.

Question 22. What Is The Protocol Used By Rmi?

Answer :

RMI-IIOP

Question 23. What Is A Hashcode?

Answer :

hashcode value for this object which is unique for every object.

Question 24. What Is A Thread?

Answer :

Thread is a block of code which can execute concurrently with other threads in the JVM.

Question 25. What Is The Algorithm Used In Thread Scheduling?

Answer :

Fixed priority scheduling.

Question 26. What Is Hash-collision In Hashtable And How It Is Handled In Java?

Answer :

Two different keys with the same hash value. Two different entries will be kept in a single hash bucket to avoid the collision.

Question 27. What Are The Different Driver Types Available In Jdbc?

Answer :

A JDBC-ODBC bridge
A native-API partly Java technology-enabled driver
A net-protocol fully Java technology-enabled driver
A native-protocol fully Java technology-enabled driver For more information.
Question 28. Is Jdbc-odbc Bridge Multi-threaded?

Answer :

No

Question 29. Does The Jdbc-odbc Bridge Support Multiple Concurrent Open Statements Per Connection?

Answer :

No

Question 30. What Is The Use Of Serializable?

Answer :

To persist the state of an object into any perminant storage device.

****
****

Spring Interview Questions.
---

1. what is IOC (inversion of control) Container.

-> Ioc is a principle or paradigm. we have some set of rules or guidelines to develop a application in a decouple manner.
-> ioc is a collabrating the object and managing the lifecycle of those objects is called ioc container.
-> Ioc container says that you don't bother about object creation or you don't create your objects only describe how they should be created i will manage it.
-> The basic concept of the Dependency Injection or Inversion of Control is that, programmer do not need to create the objects,
instead just describe how it should be created.

Benefits of Ioc.

-> minimize the code in our application.
-> it provides loose coupling between components in our application.
-> if any modification have to do then it doesnt effect other components.
-> it supports eager instantiation and lazy instantiation of services.
-> enhancement will be easy.

2. what is dependency injection. types of dependency injection.

-> dependency injection is a software design pattern that deals with how components are orgnizing their dependencies.
-> it is the process of injecting the dependencies in dependent class automatically. we no need to create obj and no need to map with obj.
-> Dependency Injection means injecting the dependency between two object as per as our requirement in our application,
this help to reducing the dependency to each other.
-> Dependency Injection (DI) is a design pattern that removes the dependency from the programming code so that it can be easy to manage and test the application.
Dependency Injection makes our programming code loosely coupled

-> it is internally follow strategy design pattern .
menas. favour composition over inheritance.
. always design to interface never code to implementation.

3. diff between setter injection and constructor injection.

-> Setter Injection.

-> to perform setter injection we will use <property> tag.
-> partial dependencies is possible. means if we have 3 dependencies like int, long, String it is not necessary to inject all values.
-> if we have more dependencies eg 15 to 20 are there in our bean class then in this case setter injection is not recomended to use as we need to
write almost 20 setters right bean lenght will be increased.
-> setter injection makes bean class object as mutable(we can change).
-> setter injection support cyclic dependencies.

Constructor injection.

-> to perform constructor injection <constructor-arg> tag is required.
-> partial injection of dependencies cannot be possible becoz for calling a constructor we must pass all the arguments.
-> if we have more dependencies in this case constructor injection is highly recomended to use becoz we can inject all the dependencies with in
the 3 to 4 lines (by calling one constructor).
-> constructor injection makes bean class obj is immutable(we cannot change).
-> constructor injection doesnt support cyclic dependencies.

4. diff between BeanFactory and application context.

BeanFactory

-> BeanFactory is a basic container. it can only manage a bean life cycle. but it can not provide service like

transaction, security etc.
-> if we developing small scale application like mobile application embeded system then we use beanfactory.
-> Beanfactory is lazy initializer. beanfactory container will not create a bean obj upto the request time.
-> Beanfactory container supports only two scope(singeltone & prototype).
-> Beanfactory doesnt support internationalization, event handling, event processing.

Application Context

-> ApplicationContext is a advanced container it manage bean life cycle and also provide transaction security etc.
-> if we are developing enterprise application like(web application, distributed application) then ApplicationContext is recomended to use.
-> ApplicationContext container creates bean object of singelton bean at the time of loading only.it is eager initialzer.
-> ApplicationContext container support all the bean scope (singletone,prototype, session,request).
-> it supports internationalization, event handling, event processing also.

5. what is bean autowiring and types of autowire modes.

-> injecting the dependencies between the obj is called wiring.
-> instead of telling the spring to manage the dependency by writting <property> or <constructor> tag in spring bean configuration file.
-> if we instruct the spring to automatically detect the dependencies and perform the injection between them it is called bean autowiring.
-> it is used only when rapid application development is required.
-> In Spring framework, you can wire beans automatically with auto-wiring feature. To enable it, just define the “autowire” attribute in.
-> The Spring container can autowire relationships between collaborating beans without using and elements which helps cut down on the amount of XML configuration.

<bean id="countryBean" class="org.arpit.java2blog.Country" autowire="byName">

Modes of Autowire
---
1. Autowire= "byname"

-> if u enable autowiring byname, spring will inject the bean based on property name. it uses setter method.
-> Autowiring by property name. Spring container looks at the properties of the beans on which autowire attribute is set to byName in the XML configuration file
and it tries to match it with name of bean in xml configuration file.

2. Autowire= "bytype"

-> if u enable autowire bytype, spring will inject the beans based on the property type. it uses setter method.
-> Autowiring by property datatype. Spring container looks at the properties of the beans on which autowire attribute is set to byType in the XML configuration file.
-> It then tries to match and wire a property if its type matches with exactly one of the beans name in configuration file

3. Autowire= "byconstructor"

-> if u enable autowire byconstructor, spring will injects the beans uses constructor.
-> byType mode in constructor argument.

4. Autowire="byautodetect"

-> Spring first tries to wire using autowire by constructor, if it does not work, Spring tries to autowire by byType.

6. what is bean scope. and types of bean scope. diff between singletone and prototype bean scope.

-> Beanscope is a concept which is provided by spring people. in spring when u declare a class as a bean by

default the bean will be created under
the singleton scope.

Types of beanscope
---
1. singleton

-> bydefault every bean declared in the configuration file is singleton.
-> Scopes a single bean definition to a single object instance per Spring IoC container.
-> singleton is default scope of a bean in Spring. You have to explicitly change scope of a bean if you want different scope.
-> if beanscope is singleton then Ioc container creates the bean class obj and keeps in the HashMap element as value by having bean 'id' as key and
uses that obj across the multiple "factory.getBean() method.

2. Prototype

-> prototype – Return a new bean instance each time when requested.
-> when we declare a beanscope as a prototype then Ioc container doesnt keep the created bean class object in HashMap so it returns new obj for every factory.getBean().

3. Request

-> request – Return a single bean instance per HTTP request.
-> when we declare a beanscope as request ,for every Http request new bean instance will be injected.

4. Session

-> session – Return a single bean instance per HTTP session.
-> for every new Http session ,new bean instance will be injected.

5. Global session

-> the global session scope is depricated in the market from spring 3.0.
-> Return a single bean instance per global HTTP session.

7. what is bean life cycle .

-> every obj in this world have life cycle. whatever the obj is performing after the birth and before the death is known as life cycle of the obj.
-> the spring container find the beans defination from the xml file and instantiate the bean .
-> using dependency injection spring populates all of the properties as specified in the bean defination.
-> in servlet life cycle we are used following life cycle methods . 1> init() 2> service() 3> destroy().
-> spring bean allows two life cycle methods. 1> init() 2> destroy().

Spring Aop
---
1. what is Aop. what is the principles of Aop. and where we apply Aop in projects.

-> Aop is not a programming language. it is a methodlogy or principles like oops. we have some set of rules or guidelines to make our application
in decouple manner.
-> Aop is the process of separating the primary logic from the secondry logic (crosscutting logic).
-> in every application there will be two types of logic, one is called primary buisness logic and other one is helper logic which makes your
primary buisness logic work better.
-> in the enterprise level application we used to add diff crosscutting functionlaties (means adding diff types of services to the application at runtime).

We can apply Aop

-> Logging.
-> Auditing.
-> Security.

-> transaction.
-> caching.
-> performance monitoring.

Principles of Aop

-> Aspect
-> Advice
-> joinpoint
-> pointcut
-> weaving
-> target
-> proxy

1.> Aspect.

-> it is the piece of code which will be separate from the primary logic. aspect represent secondry or crosscutting logic.
-> it is the piece of code that has to be applied across various classes of the application.
-> An Aspect is a class that implements concerns that cut across different classes such as logging. It is just a name.

2.> Advice.

-> Action taken by aspect at particular join point. For example: Before execution of getEmployeeName() method, put logging. So here, we are using before advice.
-> this principles talks about where actually we can apply that aspect.
a.> before advice.
b.> around advice.
3.> after returning advice
d.> throws advice.

3.> pointcut.

-> Pointcut is an expression that decides execution of advice at matched joint point. Spring uses the AspectJ pointcut expression language by default.
-> it is the set of joinpoint where advice are applied to excute the aspect.

4.> joinpoint.

-> this principles will talks about how many places we can advice the aspect .generally in spring you can apply an aspect of method execution.
-> It is a point in execution of program such as execution of method. In Spring AOP, a join point always represents a method execution.

5.> Target.

-> the class on which you want to advice the aspect.

6.> weaving.

-> the process of advising a target class with an aspect based on a pointcut to build proxy.

7.> proxy.

-> the outcome of weaving is called proxy.

Spring MVC
---

1. what do you mean by MVC.

-> it is a software design pattern. it is a structural (creational) design pattern comes in gang of four design pattern.
-> it provides loose coupling between model, view and controller.

model

-> it is responsible for data storage related logic.

view

-> it is representation for presentation logic.

Controller.

-> it is responsible for application execution logic or processing logic. it acts as a mediator between model and view.

2. what is spring MVC flow. how it works.

-> spring mvc is used for making a web application development faster, cost effective and flexiable.
-> in case of spring mvc user has send the request to the dispather servlet. dispatcher servlet acts as a front conroller with extension *.mvc,*.html etc.
-> the dispatcher servlet forward the request to the handler mapping.
-> handler mapping identify the appropriate handler or controller class for the given url request and sends to the dispatcher servlet.
-> the controller class perform operation like dao or buisness method according to the buisness requirement.then the resultant page with data will be forward to dispatcher servlet.
-> the dispatcher servlet forward it with logical view name to view resolver.
-> view resolver resolves it and create the view obj.
-> it calls the render method on view obj then the view page will be displayed.

3. what is controller. how many types of controller. which controller u are used in project. and how to create controller class in spring mvc.

-> controller is a simple java class.
-> the controller class is act as a mediator between front controller to service components.
-> the controller classes are creating by application developer to communicate with service layer.
-> the controller classs method always returns model and view.

types of controller.
---
a> simple form controller.
b> abstract controller.

c> abstract command controller.

-> To create a Controller in Spring MVC, create a class and annotate it with @Controller and @RequestMapping.

```
@Controller
@RequestMapping("/page")
public class PersonController {
@Autowired
private IPersonService personService;
@RequestMapping("/login")
public String hello(@RequestParam(value="userId",
required=false) String userId,
@RequestParam(value="location", required=false) String
location,
Model model) {
model.addAttribute("msg", "Hello
"+personService.getPersonName() );
model.addAttribute("userId", userId);
model.addAttribute("location", location);
return "result";
}
}
```

4. whtat is dispather servlet.

-> dispatcher servlet act as a front controller.
-> every incoming http request is accepting and proceesing by dispatcher servlet.
-> the dispatcher servlet we will configure in web.xml just like a normal servlet.
-> To work with Spring MVC, we need to define it in our web.xml .

```
<servlet>
<servlet-name>dispatcher</servlet-name>
<servlet-
class>org.springframework.web.servlet.DispatcherServlet</
servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
```

5. what is handler mapping . and which handler mapping u are used in project.

-> handler mapping are used to map an incomming http request url with a controller class.
-> whenever the request coming from client.the dispatcher servlet will delegate the request obj to handler mapping to identify the appropriate controller class.
-> handler mapping beans we can configure in spring configuration class.

types of handler mapping.

1.> simple url handlere mapping.
2.> bean name url handler mapping.
3.>. handler interceptors.

6. what is view resolver. and which view resolver u are used in project.

-> the view resolver are used to find out the actual view obj for the given logical view name.
-> after getting the model & view object, the dispatcher servlet will delegate the logical view name to view resolver to identify actual view object.

types of view resolver.
---
a.> xml view resolver.
b.> url based view resolver.
c.> resource bundle view resolver.

7.> what is model and view.

-> model and view is predefined classes. the model and view object is holding model data and logical view name.

How to handle views in Spring MVC using XML.

Ans: To handle views in Spring MVC, we need to configure InternalResourceViewResolver bean in spring XML where we need to define prefix and suffix of our views name.
Find the sample declaration.
<bean
class="org.springframework.web.servlet.view.InternalResourc

```
eViewResolver">
<property name="prefix" value="/pages/"/>
<property name="suffix" value=".jsp"/>
</bean>
```

What is minimum web.xml configuration to run Spring MVC.

-> To run the Spring MVC, we need to define DispatcherServlet, contextConfigLocation and ContextLoaderListener in web.xml. Find the sample web.xml.

```
<servlet>
<servlet-name>dispatcher</servlet-name>
<servlet-
class>org.springframework.web.servlet.DispatcherServlet</
servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>dispatcher</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/dispatcher-servlet.xml</param-value>
</context-param>
<listener>
<listener-
class>org.springframework.web.context.ContextLoaderListener
</listener-class>
</listener>
```

8. which annotation u are used in project. can us list them.

-> 1.> @Autowired

-> this annotation can be applied at field level,setter method level,constructor level and orbitary level .

-> when we write @Autowire then bydefault an attribute become true ie required =true, if we use @Autowire annotations then that setter or constructor or attributes become mandatory.

2.> @Qualifier

-> it is used to control which bean should be autowire on a field.
-> You can have more than one bean of same type in your XML configuration but you want to autowire only one of them , so @Qualifier removes confusion created by @Autowired by declaring exactly which bean is to autowired.

@Required

-> This annotation simply indicates that the affected bean property must be populated at configuration time:
either through an explicit property value in a bean definition or through autowiring.
-> Suppose you have very large application and you get NullPointerExceptions because required dependency has not been injected then .
it is very hard to find out what goes wrong.So this annotation helps us in debugging the code.

Streotype Annotations

-> streotype annotation are used to class level. it mean we can write these annotations only at on the top of the class.
-> these annotations are used to detect classes and creates object without xml <bean> code.
-> another way to make our class as bean and place into ioc container we have to use streotype annotations.

1.> @Ccomponent (utils, validator).

-> by using @Component we can make our class as bean and ioc container will take the class and places to the ioc container as a bean.

2.> @Service (service layer).

-> @Service annotation also used for making our class as bean into ioc container.
-> it indicates a service component in the buisness layer.

3.> @Repositry (data access layer)

-> it indicates Dao component in the presentation layer.
-> @Repositry annotation also used for making our class as a bean into ioc container.
-> @Repositry indicates the class contains dao persistency logic.

4.> @Controller (presentation layer)

-> it indicates a class has presentation logic.
-> it is also used for making our class as beans into ioc container.
-> it is Used at the class level.
-> it Tells the spring framework that the marked class acts as a controller.

```
@Controller
public class EmployeeController{
}
```

@RequestMapping

-> it Can be used at the class level and method level in controllers.

Spring Security
---

1. what is spring security. how it works.

-> spring security provides various security features like: authentication, autherization to create secure java enterprise application.
-> authentication is the process of knowing and identifying the user that wants to access.
-> autherization is the process to allow authority to perform actions in the application.
-> username - identification.
password - verification.
-> dependencies needed developing spring boot security application.

step-1 -> add spring security dependencies in pom.xml.

1.> spring-security-core.
2.> spring-security-web.
3.> spring-security-config.

step-2 -> create a separate xml file for security configurations (security-context.xml).

```
<!-- we will be defining all security related confuguration in this file-->
<http use-expressions = "true">
<intercept-url pattern="/** access = "isAuthenticated()">
<!--this means all url in this app will be checked if user is authenticated-->
<form-login/>
<!-- we will just use the built in form login page in spring-->
<logout logout-url = "/logout" logout-success-url="/index"/>
<!--the logout url we wil use in jsp-->
</http>
<authentication-manager>
<authentication-provider>

<user-service>
<user name = "mohan" password = "kumar" authorities = "admin, user"/>
<user name = "gandhi" password = "rakesh" authorities ="user"/>
</user-service>
<authentication-manager>
<authentication-provider>
```

step-3 -> configure web.xml to include spring security.

```
<filter>
<filter-name>springSecurityFilterChain</filter-name>
<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
<filter-name>springFilterChain</filter-name>
```

<url-pattern>/*</url-pattern>
</filter-mapping>
-> all request now will go through "sprigSecurityFilterChain" filter which will apply app security.

LDAP (light weight directory access protocol) important dependencies.

-> spring-ldap-core.
-> spring-security-ldap.
-> unbound id-ldap sdk.

what is delagating filter proxy.

-> it provides the link between web.xml and the application context.
<filter>
<filter-name>myFilter</filter-name>
<filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>
</filter>
<filter-mapping>
<filter-name>myFilter</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>

19. what is inheritance mapping in hibernate .and how many types of inheritance mapping.

****

1. What is Tight Coupling?

When a class (ClassA) is dependent on another class's object (ClassB), then we say ClassA is "tightly" Coupled with ClassB. Spring helps us to create classes in a way that Tight Coupling can be removed and Loose Coupling can be done.

2. What is Loose Coupling?

Loose Coupling removes the dependency of an object (ClassB) on a class (ClassA). Loose Coupling is approached by creating an interface and a setter & getter method, or by using a constructor which takes the interface object.

3. What are Beans in Spring?

When a class is annotated or decorated using the @Component, such a class is called a Bean in Spring. Beans are maintained by Application Context.

4. Explain Bean creation process?

The process of Bean creation has the following phases

(i) Starts with a class (c1) which has the annotation @Component.

(ii) Checks if the component annotated class (c1) is dependent.

(iii) If yes, then Spring will create a bean for that class (c2) too.

(iv) A connection or autowiring will occur between the two classes (c1 and c2) using @Autowired annotation and also through the constructor (c2) or the default case setClass Function (interface the Interface).

5. What is the importance of the annotation @Primary

This annotation is used on a class that needs to be taken by spring on a primary basis. For instance, if ClassX is @Component annotated and is dependent on both Class1 and Class2 (both @Component annotated) then the compiler would report an error. To show the primary class between Class1 and Class2 we use @Primary.

6. What is Dependency Injection?

Dependency Injection is where Spring searches for beans;

once the appropriate bean is found, it autowires the bean to the dependent class. Dependency Injection is the process where Spring framework looks for the beans and identifies the dependencies, and creates the instances of beans and autowires them.

7. Explain Inversion of Control (IOC).

In Tight Coupling the dependent class takes the responsibility of creating its dependency. Whereas, in Loose Coupling, we use @Autowired annotation over the dependency class (or reference) and Spring takes control of creating the instance and injects the dependency.

8. What are the roles of an IOC (Inversion of Control) Container?

IOC Container does the following things-

(i) Find Beans

(ii) Identify their dependencies and wire the dependencies

(iii) Manage Lifecycle of the Bean (creation, processing, and destruction)

Discover the Difference Between Java And Javascript

9. What is Application Context?

It is an advanced version of IOC Container. It provides all the functionalities of Bean Factory and also provides things like AOP, Internationalization capabilities, web application context (request, session, etc).

10. Explain the process of creating an ApplicationContext in Spring.

The ApplicationContext can be defined in two ways (i) using XML, (ii) using @Configuration. Once the configuration is done in any of the ways defined above, the ApplicationContext is created using new ClassPathXmlApplicationContext. The ClassPathXmlApplicationContext looks for the XML files,

using this is one of the two ways. The other way is to use AnnotationConfigApplicationContext.

11. Explain Component Scan.

Component Scan is one method of asking Spring to detect Spring-managed components, the input for this search is the packages. Two methods are available to define a Component Scan-

(i) Java Configuration; wherein, we use the @Component annotation to which we specify all the packages, for which Spring does the search.

(ii) XML Configuration- we use <context:component-scan base-package="com.demo.compscanex"/>

12. How do you perform the same (above question) in Spring Boot?

In Spring Boot the annotation used to perform the scan is @SpringBootApplication. This annotation on a class would automatically initiate the component scan on the package where they are in.

13. Differentiate @Component, @Repository and @Service and @Controller?

Typically a web application is developed in layers like the controller (which is the initial point of client communication), business (where the actual code or logic of the application is written) and DAO (where the database connections and interaction happens). In such an architecture web application, @Component can be used in any of the layers. Whereas, the @Controller is used in the controller/web layer. @Service is used in the business layer and @Repository is used in the DAO layer.

14. List out the different scopes of Bean.

(i) Singleton: throughout the spring context only one instance is created.

(ii) Prototype: a new bean is created whenever requested.

(iii) Request: Every HTTP Request creates a bean.

(iv) Session: A bean for every HTTP Session.

15. List out the types of Dependency Injection.

The types of Dependency Injection–

(i) Setter Injection and (ii) Constructor Injection.

16. What is the difference between the Configuration types XML and Annotation?

These are the two ways of setting up the configuration, and they perform in the say way. Though, when the annotation approach is taken very less amount of code is written and the result would be the same as compared to the XML approach.

17. List out the ways Autowiring is done.

(i) byType
(ii) byName
(iii) Constructor (same as byType, but through constructor)

You may also like: Top 5 Skills That Make You A Sure Shot Programmer

18. What is Dirty Read?

When a transaction (t1) is meant to read the changes that are performed by another transaction (t2) and provided transaction t2 is not committed yet; then in such a situation, the transaction t1 is called Dirty Read transaction.

19. List out the new features available in Spring Framework 4.0 and Spring Framework 5.0?

Spring 4.0 is the first to support Java features. Spring 5.0 has the support for Reactive Programming and Kotlin.

20. What is a FrontController?

In FrontController, the Servlet will not get the first request; the first request would go to FrontController and the request is passed on to the right servlet. In other words, DispatcherServlet is the front controller which intercepts all the requests from the client and then dispatches to appropriate controllers.

21. What is a ViewResolver?

ViewResolver enables a web application to select its view (such as JSP) dynamically. ViewResolver gets a name which is appended by /WEB-INF/views and a .jsp. All the display on the content is done in an HTML page.

22. List out all the concepts that are available in the MVC Architecture?

(i) The browser sends a request to DispatcherServlet

(ii) DispatcherServlet knows the HanderMapping and can find the appropriate controllers

(iii) Controllers execute the request and put the data in the model and return back the view name to the DispatcherServlet.

(iv) DispatcherServlet uses the view name and ViewResolver to map to the view.

23. Explain Model Attribute?

The annotation @ModelAttribute is decorated on a method typically present inside a Controller. This will help the method to be available in all other methods available in the controller.

24. What is a Session Attribute?

The annotation @SessionAttributes (“argument”) is decorated on class (Controller). The attribute (argument) that is present in Model is available in the session.

25. Explain the @InitBinder?

This annotation is decorated on a method in which a date format is declared, and throughout the class, the defined date format is used. Whenever the binding happens with a date field @InitBinder; annotation says to use the CustomDateEditor, which in return uses the date format mentioned in @InitBinder.

26. Define @ControllerAdvice?

This annotation is used when logic needs to be implemented commonly in multiple classes (Controllers). For instance, if an Exception or its subclasses, or when an exception is raised in the classes, it will be handled by a method annotated with @ExceptionHandler. Whenever an exception occurs in any of the controllers, the exception is handled by the method annotated with @ExceptionHandler.

27. Why Spring Boot?

Spring-based applications have a lot of configuration (boiler-plate code). In Spring MVC, a lot of configuration is required (like component scan, dispatcher servlet, view resolver, etc). Whereas, in Spring Boot the boiler-plate code is not required.

28. Spring vs Spring MVC vs Spring Boot?

Spring: the most important feature of Spring is Dependency Injection or Inversion of Control.

Spring MVC: provides a decoupled approach in developing web applications. Concepts like DispatcherServlet, ModelAndView, ViewResolver makes web application development easy.

Spring Boot: makes the configuration very easy and automatic using a feature called Auto Configuration, in which the DispatcherServlet is done by Spring internally.

29. What is the role of @SpringBootApplication?

This annotation is used to launch up the entire application. Internally, @SpringBootApplication does the

following,

@SpringBootConfiguration: same as @Configuration in a Spring Application.

@EnableAutoConfiguration: auto-configures the classes available in the classpath.

@ComponentScan: all the classes available under a package will be scanned when this annotation is applied.

30. What does an Embedded Server mean in Spring Boot?

To deploy any web application a server like Tomcat is required. In Spring Boot a server (like Tomcat) is available as part of the application in a jar. The concept of Embedded Server makes the deployment of application very easy and independent.

31. Why do we use application.properties?

The file application.properties is used to configure things like database details, log generation, security (username/ password), serialization, etc.

32. What is Spring JDBC?

Spring JDBC uses methods like update (query), execute (query) and query (SQL, resultSetExtractor) to interact with the database.

33. What is the difference between JDBC and Spring JDBC?

In JDBC, the checked exceptions need to be written; whereas, in Spring JDBC those exceptions are made into Runtime Exceptions. Which means, exception handling is not manually done in Spring JDBC.

34. What is JPA?

Java Persistence API (JPA) defines the mapping from Java Object to a Database Table. The procedure to map a Java object to a row in a database table is defined in JPA. JPA provides a lot of useful annotations, using which the

relationship between classes and tables are defined.

35. What is Hibernate?
Once the mapping is done, Hibernate (a JPA Implementation) will help us create query under the hood and interact with the database.

36. Describe the cases in which the Dependency Injection is done through Constructors and Setters?

When the dependencies are required/mandatory, the Constructor approach is selected. And when the dependencies are optional then the Setters approach is used.

37. What is the importance of POM.XML file?

Project Object Model (POM) is an XML formatted file in which all the configuration for a maven project is defined. The most commonly used tags in POM.XML are <groupid>, <artifactId>, <version>, <packaging> and a few more.

38. What does the @RequestParam annotation do?

This allows the server side to read from data and automatically bind it to a parameter coming into the method.

39. What is Spring Security?

Spring Security provides security services to J2EE applications. Spring Security is implemented using Servlet Filters under the hood. Servlet Filters are used to pre-process or post-process web requests.

40. What is CSRF?

Cross-Site Request Forgery (CSRF) is a security attack where a fraudulent website tricks the user into performing an event on the web application that he/she is logged into. For instance, if the user is logged into the online banking account, this attack tricks the user into transferring the money to an unknown person.

****
****

Question 1. What Is Java Collections Framework? What Are The Benefits Of Using Collections Framework?

Answer :

Java Collections Framework provides a standard way to handle a group of objects. Benefits of the Collections Framework are –

High performance, implementation of the Collection classes (ArrayList, LinkedList, HashSet etc.) are very efficient and used as-is in most of the cases.
Reduced effort as framework already provides several implementations for most of the scenarios that can be used as-is.
Provides interoperability, as exp. if you are using List interface any implementation that implements List interface can be swapped with the existing one.
If you want to extend any collection that can be easily done using the standard interfaces provided with in the Collections frameworks.
Question 2. Describe The Collection Framework Hierarchy?

Answer :

At the root of the Collections framework is Collection interface, it must be implemented by any class that defines a collection. This interface declares the core methods that every collection will have, if any class doesn't implement any of the method then it can throw UnsupportedOperationException.

Then there are List and Set interfaces that extend Collection interface and provided some of its own behaviour that will be further implemented by the classes that implement List and Set interfaces respectively.

There is also a Queue interface that extends collection to provide behaviour of a queue. On the other hand there is

Map interface which provides core methods for the Map implementations.

Question 3. Name The Core Collection Classes?

Answer :

AbstractCollection – Implements most of the Collection interface.
AbstractList – Extends AbstractCollection and implements List interface.
AbstractSequentialList – Extends AbstractList to provide sequential access.
ArrayList – Extends AbstractList to provide dynamic array implementation.
LinkedList – Extends AbstractSequentialList to implement a linked list.
AbstractQueue – Extends AbstractCollection and implements Queue interface.
ArrayDeque – Resizable-array implementation of the Deque interface.
PriorityQueue – Extends AbstractQueue to provide a priority-based queue.
AbstractSet – Extends AbstractCollection and implements Set interface.
HashSet – Extends AbstractSet backed by a HashTable (actually a HashMap instance).
LinkedHashSet – Extends HashSet to provide insertion order interation.
TreeSet – Extends AbstractSet (and implements NavigableSet interface) to provide sorted set.
EnumSet – Extends AbstractSet for use with Enum elements.

Question 4. How Generics Changed The Collection Framework?

Answer :

With JDK5 the Collection framework was reengineered to add generics. That was done to add "Type Safety" to the collections. Prior to JDK5 (and generics) elements were stored in collections as Object references, which brought the danger of accidentally storing incompatible types in Collection because for Collections everything was Object reference. Which would have resulted in run time error when

trying to retrieve elements and casting them to the desired type.

With the introduction of generics now we can explicitly tell which data type is to be stored in the collections, which helps in avoiding the run time error. As Collection will throw error at compile time itself for incompatible data type. As exp.

```
Map<String, String> cityTemperatureMap = new
LinkedHashMap<String, String>();
```

Here we are explicitly stating that this LinkedHashMap can only store string as both key and value.

Question 5. What Is For-each Style Loop?

Answer :

Another feature which was added to Collection with JDK 5 is for-each style loop. Any collection which wants to be the target of the "for-each loop" statement has to implement iterable interface.

Using for-each loop is easier than constructing the iterator to iterate over the collection and can be used in most of the cases rather than using the iterator loop.

If you have a list of Strings, that can be iterated using for-each loop like this.

```
for(String city : cityList){
System.out.println("Name " + city);
}
```

Question 6. What Is A Diamond Operator?

Answer :

Diamond operator let the compiler infer the type arguments for the generic classes. It is added in Java 7.

As Example - Before JDK7 if we had to define a Map using

String as both Key and Value we had to write it like this -

```
Map<String, String> cityMap = new HashMap<String,
String>();
```

In Java SE 7, you can substitute the parameterized type of the constructor with an empty set of type parameters (<>) known as diamond operator.

```
Map<String, String> cityMap = new HashMap<>();
```

Question 7. What Are The Changes In Collections Framework In Java 8?

Answer :

There are several changes in Collections Framework in Java 8 mostly influenced by the inclusion of Lambda expression in Java 8 -

Stream API - Stream can be obtained for Collection via the stream() and parallelStream() methods;

As exp if you have a list of cities in a List and you want to remove the duplicate cities from that list it can be done as

```
cityList =
cityList.stream().distinct().collect(Collectors.toList());
```

ForEach loop which can be used with Collection.
Spliterators which are helpful in parallel processing where several threads can itearate/process part of the collection.
New methods are added to Collections like replaceAll, getOrDefault, putIfAbsent in Map.
HashMap, LinkedHashMap and ConcurrentHashMap.implementation is changed to reduce hash collisions. Instead of linked list a balanced tree is used to store the entries after a certain threshold is reached.
Question 8. What Is Foreach Statement Added In Java 8?

Answer :

Java 8 has added a functional style lopping to the Collections. Real advantage of this loop is when it is used

on a stream with the chain of functional methods. If you have a Map<String, String> then it can be looped using ForEach statement like this –

```
Set<Map.Entry<String, String>> valueSet =
cityMap.entrySet();

valueSet.forEach((a)->System.out.println("Key is " +
a.getKey() + " Value is " + a.getValue()));
```

Question 9. What Is Randomaccess Interface?

Answer :

RandomAccess interface is a marker interface used by List implementations to indicate that they support fast (generally constant time) random access.

Question 10. Which Collection Classes Implement Random Access Interface?

Answer :

In Java.util package the classes that implement random access interface are – ArrayList, CopyOnWriteArrayList, Stack, Vector.

Question 11. What All Collection Classes Are Inherently Thread-safe?

Answer :

In the initial Collection classes like Vector, HashTable and Stack all the methods were synchronized to make these classes thread safe.

Later implementations starting from Java 1.2 were not synchronized.

Classes in java.util.concurrent package like ConcurrentHashMap, CopyOnWriteArrayList also provides thread safety but with a different implementation that doesn't require making all the methods synchronized.

Question 12. How Can We Synchronize A Collection Or How To Make A Collection Thread-safe?

Answer :

Collections class has several static methods that return synchronized collections. Each of the six core collection interfaces -Collection, Set, List, Map, SortedSet, and SortedMap - has one static factory method.

```
public static <T> Collection<T>
synchronizedCollection(Collection<T> c);
public static <T> Set<T> synchronizedSet(Set<T> s);
public static <T> List<T> synchronizedList(List<T> list);
public static <K,V> Map<K,V> synchronizedMap(Map<K,V> m);
public static <T> SortedSet<T>
synchronizedSortedSet(SortedSet<T> s);
public static <K,V> SortedMap<K,V>
synchronizedSortedMap(SortedMap<K,V> m);
```

Each of these methods returns a synchronized (thread-safe) Collection backed up by the specified collection.

Question 13. How To Make A Collection Class Immutable?

Answer :

Collections class provides static method to make a Collection unmodifiable. Each of the six core collection interfaces -Collection, Set, List, Map, SortedSet, and SortedMap - has one static factory method.

```
public static <T> Collection<T>
unmodifiableCollection(Collection<? extends T> c);

public static <T> Set<T> unmodifiableSet(Set<? extends T>
s);

public static <T> List<T> unmodifiableList(List<? extends
T> list);

public static <K,V> Map<K, V> unmodifiableMap(Map<? extends
K, ? extends V> m);
```

```
public static <T> SortedSet<T>
unmodifiableSortedSet(SortedSet<? extends T> s);

public static <K,V> SortedMap<K, V>
unmodifiableSortedMap(SortedMap<K, ? extends V> m);
```

Question 14. What Is List Interface?

Answer :

List interface extends the root interface Collection and adds behaviour for the collection that stores a sequence of elements. These elements can be inserted or accessed by their position in the list using a zero-based index.

Question 15. Which Collection Classes Implement List Interface?

Answer :

Some of the Collection classes that implement List interface are ArrayList, CopyOnWriteArrayList, LinkedList, Stack, Vector.

Question 16. Why Arraylist Is Called Dynamically Growing Array? How That Dynamic Behaviour Is Achieved?

Answer :

In the case of ArrayList you don't have to anticipate in advance, like in the case of array, how many elements you are going to store in the arraylist. As and when elements are added list keeps growing. That is why ArrayList is called dynamically growing array.

For ArrayList, data structure used for storing elements is array itself. When ArrayList is created it initializes an array with an initial capacity (default is array of length 10). When that limit is crossed another array is created which is 1.5 times the original array and the elements from the old array are copied to the new array.

Question 17. How Arraylist Works Internally In Java?

Answer :

Basic data structure used by ArrayList to store objects is an array of Object class, which is defined like –

transient Object[] elementData;

When we add an element to an ArrayList it first verifies whether it has that much capacity in the array to store new element or not, in case there is not then the new capacity is calculated which is 50% more than the old capacity and the array is increased by that much capacity (Actually uses Arrays.copyOf which returns the original array increased to the new length)

Question 18. How To Add Elements To An Arraylist?

Answer :

List provides a method add(E e) which appends specified element to the end of the list. Using add(E e) method will mean keep adding elements sequentially to the list.
Another add method – add(int index, E element) inserts the specified element at the specified position in this list.
Third method addAll(int index, Collection<? extends E> c) inserts all of the elements in the specified collection into this list, starting at the specified position.
Question 19. Does Arraylist Allow Duplicate Elements?

Answer :

Yes. List allows duplicate elements to be added.

Question 20. Does Arraylist Allow Null?

Answer :

In ArrayList any number of nulls can be added.

Question 21. How To Join Two Or More Arraylists?

Answer :

List provides a method addAll to join two or more lists in Java.

If you have one list cityList and another List secondCityList then you can join them using addAll like this –

```
cityList.addAll(secondCityList);
```

Question 22. How To Remove Elements From An Arraylist?

Answer :

ArrayList provides several methods to remove elements from the List. Since ArrayList internally uses array to store elements, one point to note is that when an element is removed from the List the remaining elements have to be shifted to fill the gap created in the underlying array.

clear() – Removes all of the elements from this list.
remove(int index) – Removes the element at the specified position in this list.
remove(Object o) – Removes the first occurrence of the specified element from this list, if it is present.
removeAll(Collection<?> c) – Removes from this list all of its elements that are contained in the specified collection.
removeIf(Predicate<? super E> filter) – Removes all of the elements of this collection that satisfy the given predicate.

Question 23. How To Remove Duplicate Elements From An Arraylist?

Answer :

We can use a HashSet to do the job of removing the duplicate elements. HashSet only stores unique elements and we'll use that feature of HashSet to remove duplicates.

If you have a List called cityList you can create a HashSet using this list –

```
Set<String> citySet = new HashSet<String>(cityList);
```

```
then add this set back to the cityList
cityList.clear();
cityList.addAll(citySet);
```

That will remove all the duplicate elements from the given list. Note that insertion order won't be retained if a HashSet is used. In case insertion order is to be retained use LinkedHashSet.

Question 24. How To Loop/iterate An Arraylist In Java?

Answer :

There are many ways to loop/iterate an arrayList in Java. Options are –

for loop
for-each loop
iterator
list iterator
Java 8 forEach loop
for-each loop is the best way to iterate a list if you just need to traverse sequentially through a list.

Question 25. What Is Listiterator In Java?

Answer :

ListIterator provides the functionality to iterate a list in both directions. The interesting point about list iterator is that it has no current element. Its current cursor position always lies between the element that would be returned by a call to previous() and the element that would be returned by a call to next().

Question 26. How To Sort Arraylist In Java?

Answer :

Collections class has a static method sort which can be used for sorting an arraylist.

There are two overloaded versions of sort method –

public static <T extends Comparable<? super T>> void sort(List<T> list) – Sorts the specified list into ascending order, according to the natural ordering of its elements.
public static <T> void sort(List<T> list, Comparator<? super T> c) – Sorts the specified list according to the order induced by the specified comparator.
If you have a List called cityList it can be sorted like this –

```
Collections.sort(cityList);
```

Question 27. How To Sort Arraylist Of Strings In Descending Order?

Answer :

Collections.sort() method will sort the ArrayList of String in ascending order.

To have a descending order either comparator has to be provided or reverseOrder method of the Collections class can be used.

```
Collections.sort(cityList, Collections.reverseOrder());
```

Question 28. How And Why To Synchronize Arraylist In Java?

Answer :

In order to have better performance most of the Collections are not synchronized. Which means sharing an instance of arrayList among many threads where those threads are modifying (by adding or removing the values) the collection may result in unpredictable behaviour.

To synchronize a List Collections.synchronizedList() method can be used.

Question 29. How To Convert Arraylist To Array In Java?

Answer :

Within Collection interface there are two versions of

toArray() which can be used to convert ArrayList to array.

Object[] toArray()
<T> T[] toArray(T array[])
The first version returns an array of Object.

The second version returns an array containing the elements of the list. Normally we go with the second version because it returns the array of the same type as List.

If you have a List called cityList, it can be converted to an array like this –

```
String cityArray[] = new String[cityList.size()];
cityArray = cityList.toArray(cityArray);
```

Question 30. How To Convert Array To Arraylist In Java?

Answer :

Arrays class contains a static factory method asList() that allows arrays to be viewed as lists.

```
public static <T> List<T> asList(T... a)
```

As Exp.

```
String cityArray[] = {"Delhi", "Mumbai", "Bangalore",
"Hyderabad", "Chennai"};

//Converting array to List

List<String> cityList = Arrays.asList(cityArray);
```

Question 31. How Is The Performance Of An Arraylist?

Answer :

Adding an element – If you are adding at the end using add(E e) method it is O(1). In the worst case it may go to O(n). That will happen if you add more elements than the capacity of the underlying array.
Retrieving an element – Since ArrayList internally uses an array to store elements so get(int index) means going to

that index directly in the array. So, for ArrayList get(int index) is O(1).
Removing an element – If you are removing using the remove(int index) method then, in case of ArrayList getting to that index is fast but removing will mean shuffling the remaining elements to fill the gap created by the removed element with in the underlying array. Thus it can be said remove(int index) operation is O(n – index) for the arraylist.

Question 32. What Is Linkedlist Class In Java? How Is It Implemented?

Answer :

LinkedList class in Java implements List and Deque interfaces and LinkedList implements it using doubly linkedlist.

Within the LinkedList implementation there is a private class Node which provides the structure for a node in a doubly linked list. It has item variable for holding the value and two reference to Node class itself for connecting to next and previous nodes.

Question 33. What Is Deque Interface In Java?

Answer :

Java.util.Deque is an interface in Java which extends Queue interface and provides support for element insertion and removal at both ends. The name deque is short for "double ended queue" and is usually pronounced "deck".

Some of the implementing classes of the Deque interface are LinkedList, ConcurrentLinkedDeque, LinkedBlockingDeque. Note that addFirst() and addLast() methods provided in the LinkedList implementation are specified by the Deque interface.

Question 34. Difference Between Arraylist And Linkedlist In Java?

Answer :

In Java collections framework ArrayList and LinkedList are two different implementations of List interface

LinkedList is implemented using a doubly linked list concept where as ArrayList internally uses an array of Objects which can be resized dynamically
For LinkedList add(e Element) is always O(1) where as for ArrayList add(e Element) operation runs in amortized constant time, that is, adding n elements requires O(n) time.
For LinkedList get(int index) is O(n) where as for ArrayList get(int index) is O(1).
If you are removing using the remove(int index) method then for LinkedList class it will be O(n). In case of ArrayList getting to that index is fast but removing will mean shuffling the remaining elements to fill the gap created by the removed element with in the underlying array.
Question 35. Difference Between Arraylist And Vector In Java?

Answer :

ArrayList is not synchronized whereas Vector is synchronized.
Performance wise ArrayList is fast in comparison to Vector as ArrayList is not synchronized.
ArrayList, by default, grows by 50% in case the initial capacity is exhausted. In case of Vector the backing array's size is doubled by default.
For traversing an ArrayList iterator is used. For traversing Vecor Iterator/Enumerator can be used. Note that Iterator is fail-fast for both Vector and ArrayList. Enumerator which can be used with Vector is not fail-fast.
Question 36. Difference Between Array And Arraylist In Java?

Answer :

Array is fixed in size which is provided at the time of creating an Array. ArrayList grows dynamically and also known as dynamically growing array.
Array can store primitive types as well as objects. In ArrayList only Objects can be stored.
Arrays can be multi-dimensional whereas ArrayList is always

unidimensional.

As Exp. a two dimensional array can be created as -

```
Integer myArray[][] = new Integer[4][3];
```

Performance wise both Array and ArrayList are almost same as internally ArrayList also uses an Array. But there is overhead of resizing the array and copying the elements to the new Array in case of ArrayList.

Question 37. What Is An Iterator?

Answer :

Iterator is an interface which is part of the Java Collections framework. An Iterator enables you to iterate a collection. Iterators also allow the caller to remove elements from the underlying collection during the iteration.

Collection classes provide iterator method which returns an iterator. This iterator can be used to traverse a collection.

As exp. if there is a set called citySet, an iterator on this set can be obtained as -

```
Iterator<String> itr = citySet.iterator();
while (itr.hasNext()) {
String city = (String) itr.next();
System.out.println("city " + city);
}
```

Question 38. Difference Between Iterator And Listiterator?

Answer :

Iterator can be obtained on any Collection class like List or Set. But ListIterator can only be used to traverse a List.
Iterator only moves in one direction using next() method. ListIterator can iterate in both directions using next() and previous() methods.
Iterator always start at the beginning of the collection.

ListIterator can be obtained at any point.

As Exp. If you have a List of integers numberList, you can obtain a ListIterator from the third index of this List.
ListIterator<Integer> ltr = numberList.listIterator(3);

ListIterator provides an add(E e) method which is not there in Iterator. add(E e) inserts the specified element into the list.
ListItearator also provides set method. void set(E e) replaces the last element returned by next() or previous() with the specified element
Question 39. Why Iterator Doesn't Have Add Method Whereas Listiterator Has Add Method?

Answer :

Iterator can be obtained on any Collection class like List or Set so contract for Iterator makes no guarantees about the order of iteration.

But ListIterator can only be used to traverse a List so it does guarantee the order of the iteration. That is why ListIterator provides an add operation.

Question 40. What Is Fail Fast Iterator?

Answer :

An iterator is considered fail-fast if it throws a ConcurrentModificationException under either of the following two conditions:

In multi-threaded environment, if one thread is trying to modify a Collection while another thread is iterating over it.
Even with single thread, if a thread modifies a collection directly while it is iterating over the collection with a fail-fast iterator, the iterator will throw this exception.
fail-fast iterator will throw a ConcurrentModificationException if the underlying collection is structurally modified in any way except through the iterator's own remove or add (if applicable as

in list-iterator) methods.

Note that structural modification is any operation that adds or deletes one or more elements; merely setting the value of an element (in case of list) or changing the value associated with an existing key (in case of map) is not a structural modification.

Question 41. What Is A Fail-safe Iterator?

Answer :

An iterator is considered fail-safe if it does not throw ConcurrentModificationException.
ConcurrentModificationException is not thrown as the fail-safe iterator makes a copy of the underlying structure and iteration is done over that snapshot.

Since iteration is done over a copy of the collection so interference is impossible and the iterator is guaranteed not to throw ConcurrentModificationException.

Question 42. Difference Between Fail-fast Iterator And Fail-safe Iterator?

Answer :

fail-fast iterator throws a ConcurrentModificationException if the underlying collection is structurally modified whereas fail-safe iterator doesn't throw ConcurrentModificationException.
fail-fast iterator doesn't create a copy of the collection whereas fail-safe iterator makes a copy of the underlying structure and iteration is done over that snapshot.
fail-fast iterator provides operations like remove, add and set (in case of ListIterator) whereas in case of fail-safe iterators element-changing operations on iterators themselves (remove, set and add) are not supported. These methods throw UnsupportedOperationException.
Question 43. Can We Iterate Through A Map?

Answer :

Though we can't iterate a Map as such but Map interface has

methods which provide a set view of the Map. That set can be iterated. Those two methods are -

Set<Map.Entry<K, V>>entrySet() - This method returns a set that contains the entries in the map. The entries in the set are actually object of type Map.Entry.
Set<K>keySet() - This method returns a set that contains the keys of the map.
There is also a values() method in the Map that returns a Collection view of the values contained in this map.

Question 44. Are Maps Actually Collections Or Why Doesn't Map Extend Collection?

Answer :

Maps themselves are not collections because they don't implement Collection interface.

Question 45. What Is Map.entry In Map?

Answer :

Map.Entry is an interface in java.util, in fact Entry is a nested interface in Map interface. Nested interface must be qualified by the name of the class or interface of which it is a member, that's why it is qualified as Map.Entry.

Map.Entry represents a key/value pair that forms one element of a Map.

The Map.entrySet method returns a collection-view of the map, whose elements are of this class.

As exp. If you have a Map called cityMap then using entrySet method you can get the set view of the map whose elements are of type Map.Entry, and then using getKey and getValue methods of the Map.Entry you can get the key and value pair of the Map.

```
for(Map.Entry<String, String> entry: cityMap.entrySet()){

System.out.println("Key is " + entry.getKey() + " Value is
" + entry.getValue());
```

```
}
```

Question 46. Does Hashmap Allow Null Keys And Null Values?

Answer :

HashMap allows one null key and any number of null values. If another null key is added it won't cause any error. But the value with the new null key will override the value with old null key.

As exp. If you have a Map, cityMap and you add two values with null keys in the cityMap, Kolkata will override the New Delhi as only one null key is allowed.

```
cityMap.put(null, "New Delhi");
cityMap.put(null, "Kolkata");
```

Question 47. What Happens If Hashmap Has Duplicate Keys?

Answer :

If an attempt is made to add the same key twice, it won't cause any error but the value which is added later will override the previous value.

As Exp. If you have a Map, cityMap and you add two values with same key in the cityMap, Kolkata will override the New Delhi.

```
cityMap.put("5", "New Delhi");
cityMap.put("5", "Kolkata");
```

Question 48. What Is Hash-collision In Hash Based Collections?

Answer :

In HashMap, using the key, a Hash is calculated and that hash value decides in which bucket the particular Map.Entry object will reside.

Hash collision means more than one key having the same

calculated hash value thus stored in the same bucket. In HashMap, in that case Entry objects are stored as a linked-list with in the same bucket.

Question 49. What Is The Hashcode() Method?

Answer :

hashCode() method is present in the java.lang.Object class. This method is used to get a unique integer value for a given object. We can see it's use with hash based collections like HashTable or HashMap where hashCode() is used to find the correct bucket location where the particular (key, value) pair is stored.

Question 50. What Is Equals() Method?

Answer :

equals() method is present in the java.lang.Object class. It is used to determine the equality of two objects.

Question 51. When Do We Need To Override Hashcode() And Equals() Methods?

Answer :

The default implementation of equals() method in the Object class is a simple reference equality check.

```
public boolean equals(Object obj){
return (this == obj);
}
```

The default implementation of hashCode() in the Object class just returns integer value of the memory address of the object.

It becomes very important to override these two methods in case we are using a custom object as key in a hash based collection.

In that case we can't rely on the default implementation provided by the Object class and need to provide custom

implementation of hashCode() and equals() method.

If two objects Obj1 and Obj2 are equal according to their equals() method then they must have the same hash code too. Though the vice-versa is not true that is if two objects have the same hash code then they do not have to be equal too.

Question 52. Which Map Implementation Should Be Used If You Want To Retain The Insertion Order?

Answer :

LinkedHashMap should be used in this case.

Question 53. What Is Linkedhashmap?

Answer :

LinkedHashMap is also one of the implementation of the Map interface, apart from implementing Map interface LinkedHashMap also extends the HashMap class. So just like HashMap, LinkedHashMap also allows one null key and multiple null values.

How it differs from other implementations of the Map interface like HashMap and TreeMap is thatLinkedHashMap maintains the insertion order of the elements which means if we iterate a LinkedHashMap we'll get the keys in the order in which they were inserted in the Map.

LinkedHashMap maintains a doubly-linked list running through all of its entries and that's how it maintains the iteration order.

Question 54. Which Map Implementation Should Be Used If You Want Map Values To Be Sorted By Keys?

Answer :

TreeMap should be used in this case.

Question 55. What Is Treemap In Java?

Answer :

TreeMap is also one of the implementation of the Map interface like HashMap and LinkedHashMap. TreeMap class implements the NavigableMap interface and extends the AbstractMap class.

How it differs from other implementations of the Map interface is that objects in TreeMap are stored in sorted order. The elements are ordered using their natural ordering or a comparator can be provided at map creation time to provide custom ordering.

One important point about TreeMap is, though HashMap and LinkedHashMap allow one null as key, TreeMap doesn't allow null as key. Any attempt to add null in a TreeMap will result in a NullPointerException.

Question 56. What Is A Weakhashmap?

Answer :

WeakHashMap is a Hash table based implementation of the Map interface, with weak keys. An entry in a WeakHashMap will automatically be removed when its key is no longer in ordinary use. Which means storing only weak references allows garbage collector to remove the entry (key-value pair) from the map when its key is not referenced outside of the WeakHashMap.

In WeakHashMap both null values and the null key are supported. A WeakHashMap is created as-

```
Map weakHashMap = new WeakHashMap();
```

Question 57. What Is A Identityhashmap?

Answer :

IdentityHashMap class implements the Map interface with a hash table, using reference-equality in place of object-equality when comparing keys (and values). In other words, in an IdentityHashMap, two keys k1 and k2 are considered equal if and only if (k1==k2). Where as In normal Map

implementations (like HashMap) two keys k1 and k2 are considered equal if and only if (k1==null ? k2==null : k1.equals(k2)).

Note that This class is not a general-purpose Map implementation. While this class implements the Map interface, it intentionally violates Map's general contract, which mandates the use of the equals method when comparing objects. This class is designed for use only in the rare cases wherein reference-equality semantics are required.

Question 58. Difference Between Hashmap And Hashtable In Java?

Answer :

Though both HashTable and HashMap store elements as a (key, value) pair and use hashing technique to store elements, moreover from Java v1.2, HashTable class was retrofitted to implement the Map interface, making it a member of the Java Collections Framework. But there are certain difference between the two -

HashMap is not synchronized where as HashTable is synchronized.
HashMap allows one null value as a key and any number of null values where as HashTable does not allow null values either as key or as value.
For traversing a HashMap an iterator can be used. For traversing a HashTable either an iterator or Enumerator can be used. The iterator used for both HashMap and HashTable is fail-fast but the enumerator used with HashTable is fail-safe.
Performance wise HashMap is faster than the HashTable reason being HashMap is not synchronized.
Question 59. Hashmap Vs Linkedhashmap Vs Treemap In Java?

Answer :

HashMap makes no guarantees as to the order of the map.

LinkedHashMap maintains the insertion order of the elements which means if we iterate a LinkedHashMap we'll get the

keys in the order in which they were inserted in the Map.

TreeMap stores objects in sorted order.

HashMap as well as LinkedHashMap allows one null as key, multiple values may be null though.

TreeMap does not allow null as key.

HashMap stores elements in a bucket which actually is an index of the array.

LinkedHashMap also uses the same internal implementation, it also maintains a doubly-linked list running through all of its entries.

TreeMap is a Red-Black tree based NavigableMap implementation.

Performance wise HashMap provides constant time performance O(1) for get() and put() method.

LinkedHashMap also provides constant time performance O(1) for get() and put() method but in general a little slower than the HashMap as it has to maintain a doubly linked list.

TreeMap provides guaranteed log(n) time cost for the containsKey, get, put and remove operations.

Question 60. How Does Hashset Internally Works In Java?

Answer :

HashSet internally uses HashMap to store it's elements. But it differs from HashMap on two points.

HashSet only stores unique values i.e. no duplicates are allowed.

In HashSet we have add(E e) method which takes just the element to be added as parameter not the (key, value) pair.

Question 61. Which Set Implementation Should Be Used If You Want The Insertion Order To Be Maintained?

Answer :

LinkedHashSet should be used in this case.

Question 62. What Is Linkedhashset?

Answer :

LinkedHashSet is also one of the implementation of the Set interface. Actually LinkedHashSet class extends the HashSet and has no other methods of its own.

LinkedHashSet also stores unique elements just like other implemetations of the Set interface. How LinkedHashSet differs is that it maintains the insertion-order; that is elements in the LinkedHashSet are stored in the sequence in which they are inserted. Note that insertion order is not affected if an element is re-inserted into the set.

****

ERICSSON JAVA INTERVIEW QUESTIONS & ANSWERS

Question 1. What Is The Main Difference Between Java Platform And Other Platforms?

Answer :

The Java platform differs from most other platforms in the sense that it’s a software-based platform that runs on top of other hardware-based platforms.It has two components:

Runtime Environment

API(Application Programming Interface)

Question 2. When Abstract Methods Are Used?

Answer :

If you want a class to contain a particular method but you want the actual implementation of that method to be determined by child classes, you can declare the method in the parent class as abstract.

Question 3. What Are Wrapper Classes?

Answer :

These are classes that allow primitive types to be accessed as objects.

Example: Integer, Character, Double, Boolean etc.

Question 4. What Do You Mean By Platform Independence?

Answer :

Platform independence means that we can write and compile the java code in one platform (eg Windows) and can execute the class in any other supported platform eg (Linux,Solaris,etc).

Question 5. What Is A Pointer And Does Java Support Pointers?

Answer :

Pointer is a reference handle to a memory location. Improper handling of pointers leads to memory leaks and reliability issues hence Java doesn’t support the usage of pointers.

Question 6. What Is The Difference Between Creating String As New() And Literal?

Answer :

When we create string with new() Operator, it’s created in heap and not added into string pool while String created using literal are created in String pool itself which exists in PermGen area of heap.

String s = new String(“Test”);

does not put the object in String pool , we need to call String.intern() method which is used to put them into String pool explicitly. its only when you create String object as String literal e.g. String s = “Test” Java automatically put that into String pool. (Ericsson JAVA interview questions)

Question 7. What Is The Use Of The Finally Block? Is Finally Block In Java Guaranteed To Be Called? When Finally Block Is Not Called?

Answer :

Finally is the block of code that executes always. The code in finally block will execute even if an exception is occurred. Finally block is NOT called in following conditions

If the JVM exits while the try or catch code is being executed, then the finally block may not execute. This may happen due to System.exit() call.

if the thread executing the try or catch code is interrupted or killed, the finally block may not execute even though the application as a whole continues.

If a exception is thrown in finally block and not handled then remaining code in finally block may not be executed.

Question 8. What’s The Difference Between The Methods Sleep() And Wait()?

Answer :

The code sleep(2000); puts thread aside for exactly two seconds. The code wait(2000), causes a wait of up to two second. A thread could stop waiting earlier if it receives the notify() or notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.

Question 9. What Is The Difference Between An Interface And An Abstract Class ?

Answer :

An abstract class can have instance methods that implement a default behavior. An Interface can only declare constants and instance methods, but cannot implement default behavior and all methods are implicitly abstract. An interface has all public members and no implementation.

Question 10. Difference Between Final, Finally And Finalize ?

Answer :

Final is used to apply restrictions on class, method and variable. Final class can’t be inherited, final method can’t be overridden and final variable value can’t be changed.

Finally is used to place important code, it will be executed whether exception is handled or not.

Finalize is used to perform clean up processing just before object is garbage collected.

****

JAVA SERVLETS INTERVIEW QUESTIONS & ANSWERS

Question 1. Can We Use The Constructor, Instead Of Init(), To Initialize Servlet?

Answer :

Yes. But you will not get the Servlet specific things from constructor. The original reason for init() was that ancient versions of Java couldn’t dynamically invoke constructors with arguments, so there was no way to give the constructor a ServletConfig. That no longer applies, but servlet containers still will only call your no-arg

constructor. So you won’t have access to a ServletConfig or ServletContext.

Question 2. What Is Servlet Context?

Answer :

The Servlet context is an object that contains a information about the Web application and container. Using the context, a Servlet can log events, obtain URL references to resources, and set and store attributes that other Servlet in the context can use.

Question 3. What Is A Servlet Filter?

Answer :

Servlet filters are pluggable Web components that allow us to implement pre-processing and post-processing logic in our Web applications.

Question 4. What Is A War File?

Answer :

WAR stands for Web Archive. It is a compressed version of your web application. You can use this WAR file to deploy your web application.

Question 5. How Would You Create Deadlock On Your Servlet?

Answer :

Calling a doPost() method inside doGet() and doGet()method inside doPost() wouleate a deadlock for a servlet.

Question 6. Why Is Httpservlet Declared Abstract?

Answer :

1.The default implementations of the main service methods can not do anything and need to be overridden. This calls of the HttpServlet class to be declared as abstract.

2.With its use the developers do not need to implement all the service methods.

Question 7. Why Is A Constructor Needed In A Servlet Even If We Use The Init Method?

Answer :

1.Although the init method of the servlet initializes it, a constructor instantiates it.
2.A developer might never explicitly call the servlet's constructor but a container uses it to create an instance of the servlet.

Question 8. What Is Genericservlet Class?

Answer :

1.GenericServlet is an abstract class which implements the Servlet interface and the ServletConfig interface.
2.Other than the methods included in above two interfaces, it also provides simple versions of the lifecycle methods init and destroy, and implements the log method declared in the ServletContext interface.
3.Since this class is not specific to any protocol, it is known as generic servlet.

Question 9. How Can The Session In Servlet Be Destroyed?

Answer :

There are two ways to destroy a session:
1. Programatically : By using session.invalidate() method. It makes the container abandon the session on which the method is called.
2. When the server shuts down.

Question 10. What Is Lazy Loading?

Answer :

The servlets are not initialized by the container from the start. It happens when the servlet is requested for the first time. This is called lazy loading.

Question 11. What Are The Mechanisms Used By A Servlet Container For Maintaining Session Information?

Answer :

For maintaining session information Servlet Container uses:
. Cookies
. URL rewriting
. HTTPS protocol information

Question 12. What Is The Procedure For Initializing A Servlet?

Answer :

- To initialize a servlet init() is used.
- init() initializes a java program.
- A constructor can also be used to initialize a servlet.

Question 13. What Is The Web Container?

Answer :

A Servlet and JSP containers are collectively referred to as Web containers.

Question 14. What Are The Uses Of Servletrequest?

Answer :

The ServletRequest gives information such as the names of the parameters passed by the client, the protocol (scheme) being used by the client, and the names of the remote host that made the request and the server that received it. The input stream, ServletInputStream.

Question 15. What Are The Uses Of Servletresponse Interface?

Answer :

ServletResponse allows the servlet to set the content

length and MIME type of that response. It provides an output stream, ServletOutputStream and a Writer through which the servlet can send data.

Question 16. How Http Servlet Handles Client Requests?

Answer :

An HTTP Servlet handles client requests through its service method. The service method supports standard HTTP client requests by dispatching each request to a method designed to handle that request.

Question 17. What Is Pre Initialization Of A Servlet?

Answer :

A container doesn't initialize the servlets when it starts up. It initializes a servlet when it receives a request for that servlet first time. This is called lazy loading. The servlet specification defines the <load-on-startup> element, which can be specified in the deployment descriptor to make the servlet container load and initialize the servlet as soon as it starts up. The process of loading a servlet before any request comes in is called preloading or pre initializing a servlet.

Question 18. How Do Servlets Handle Multiple Simultaneous Requests?

Answer :

When a request comes in, the web server will start a new thread and the request is assigned to a thread, which calls a service method of the servlet.

Question 19. What Is Servlet Chaining?

Answer :

Servlet chaining is a technique in which two or more servlets can cooperate in servicing a single request. In servlet chaining, one servlet’s output is the input of next

servlet. This process continues until the last servlet is reached. Its output is then sent back to the client. We are achieving Servlet Chaining with the help of RequestDispatcher.

Question 20. How Will You Communicate From An Applet To Servlet?

Answer :

There are three ways to communicate from an applet to servlet and they are: HTTP Communication (Text-based and object-based) , Socket Communication and RMI Communication.

Question 21. How Do You Communicate Between The Servlets?

Answer :

We can communicate between servlets by using RequestDespatcher interface and servlet chaining.

Question 22. What Is The Difference Between Context Init Parameter And Servlet Init Parameter?

Answer :

Servlet init parameters are for a single servlet only. No body out side that servlet can access that. It is declared inside the <servlet> tag inside Deployment Descriptor, where as context init parameter is for the entire web application. Any servlet or JSP in that web application can access context init parameter. Context parameters are declared in a tag <context-param> directly inside the <web-app> tag. The methods for accessing context init parameter is getServletContext ().getInitParamter (“name”) where as method for accessing servlet init parameter is getServletConfig ().getInitParamter (“name”);

Question 23. What Are The Different Ways For Getting A Servlet Context?

Answer :

We will get ServletContext by calling getServletConfig

().getServletContext (). This is because a ServletConfig always hold a reference to ServletContext. By calling this.getServletContext () also we will get a ServletContext object.

Question 24. What Is Http Tunneling?

Answer :

HTTP tunneling is used to encapsulate other protocols within the HTTP or HTTPS protocols. Normally the intranet is blocked by a firewall and the network is exposed to the outer world only through a specific Web server port, that listens for only HTTP requests. To use any other protocol, that by passes the firewall, the protocol is embedded in HTTP and send as HttpRequest.

Question 25. What Are The Differences Between A Session And A Cookie?

Answer :

Session is stored in server but cookie stored in client. Session should work regardless of the settings on the client browser. There is no limit on the amount of data that can be stored on session. But it is limited in cookie. Session can store objects and cookies can store only strings. Cookies are faster than session.

Question 26. Why Should We Go For Inter Servlet Communication?

Answer :

The three major reasons to use inter servlet communication are:
a) Direct servlet manipulation – allows to gain access to the other currently loaded servlets and perform certain tasks (through the ServletContext object)
b) Servlet reuse – allows the servlet to reuse the public methods of another servlet.
c) Servlet collaboration – requires to communicate with each other by sharing specific information (through method invocation).

Question 27. What Is Client Side Refresh?

Answer :

The standard HTTP protocols ways of refreshing the page, which is normally supported by all browsers.
<META HTTP-EQUIV="Refresh" CONTENT="5; URL=/servlet/MyServlet/">
This will refresh the page in the browser automatically and loads the new data every 5 seconds.

Question 28. What Is Server Side Push?

Answer :

Server Side push is useful when data needs to change regularly on the clients application or browser, without intervention from client. The mechanism used is, when client first connects to Server, then Server keeps the TCP/IP connection open.

Question 29. What's The Servlet Interface?

Answer :

The central abstraction in the Servlet API is the Servlet interface. All servlets implement this interface, either directly or, more commonly, by extending a class that implements it such as HttpServlet.

Question 30. What Is The Difference Between Servletcontext And Servletconfig?

Answer :

The ServletConfig gives the information about the servlet initialization parameters. The servlet engine implements the ServletConfig interface in order to pass configuration information to a servlet. The server passes an object that implements the ServletConfig interface to the servlet's init() method.
The ServletContext gives information about the container. The ServletContext interface provides information to

servlets regarding the environment in which they are running. It also provides standard way for servlets to write events to a log file.

****

JAVA DEVELOPER INTERVIEW QUESTIONS & ANSWERS

Question 1. Described Heavy Weight Components Mean In Java Programming?

Answer :

Heavy weight components like Abstract Window Toolkit (AWT), depend on the local windowing toolkit. For example, java.awt.Button is a heavy weight component, when it is running on the Java platform for Unix platform, it maps to a real Motif button. In this relationship, the Motif button is called the peer to the java.awt.Button. If you create two Buttons, two peers and hence two Motif Buttons are also created. The Java platform communicates with the Motif Buttons using the Java Native Interface.

For each and every component added to the application, there is an additional overhead tied to the local windowing system, which is why these components are called heavyweight.

Question 2. Can You Please Explain The Difference Between Boolean & Operator And The && Operator In Java Programming?

Answer :

If an expression involving the Boolean & operator is evaluated, both operands are evaluated. Then the & operator is applied to the operand. When an expression involving the && operator is evaluated, the first operand is evaluated. If the first operand returns a value of true then the second operand is evaluated. The && operator is then applied to the first and second operands. If the first

operand evaluates to false, the evaluation of the second operand is skipped.

Operator & has no chance to skip both sides evaluation and && operator does.

Question 3. Does The Garbage Collection Guarantee That A Program Will Not Run Out Of Memory?

Answer :

No, it doesn't. It is possible for programs to use up memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection.

Question 4. Described The Elements Of A Gridbaglayout Organized In Java Programming?

Answer :

The elements of a GridBagLayout are organized according to a grid. However, the elements are of different sizes and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes.

Question 5. What Advantage Do Java's Layout Managers Provide Over The Traditional Windowing Systems?

Answer :

Java uses layout managers to lay out components in a consistent manner across all windowing platforms. Since Java's layout managers aren't tied to absolute sizing and positioning, they are able to accommodate platform-specific differences among windowing systems.

Question 6. How A Gui Component Handle Its Own Events In Java Programming?

Answer :

A component can handle its own events by implementing the

required event-listener interface and adding itself as its own event listener.

Question 7. Name Primitive Java Types?

Answer :

The primitive types are byte, char, short, int, long, float, double, and boolean.

Question 8. Explain The Difference Between Reader/writer Class Hierarchy And The Inputstream/outputstream Class Hierarchy In Java Programming?

Answer :

The Reader/Writer class hierarchy is character-oriented, and the InputStream/OutputStream class hierarchy is byte-oriented.

Question 9. Does A Class Inherit Constructors Of Its Superclass In Java Programming?

Answer :

A class does not inherit constructors from any of its superclasses.

Question 10. Define Map Interface In Java Programming?

Answer :

The Map interface replaces the JDK 1.1 Dictionary class and is used associate keys with values.

Question 11. If The A Class Is Declared Without Any Access Modifiers, Where May The Class Be Accessed In Java Programming?

Answer :

A class that is declared without any access modifiers is said to have package or friendly access. This means that

the class can only be accessed by other classes and interfaces that are defined within the same package.

Question 12. Which Classes Of Exceptions May Be Caught By A Catch Clause In Java Programming?

Answer :

A catch clause can catch any exception that may be assigned to the Throwable type. This includes the Error and Exception types.

Question 13. Can You Please Explain The Difference Between Preemptive Scheduling And Time Slicing In Java Programming?

Answer :

Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

Question 14. What Happens When A Thread Cannot Acquire Lock On An Object In Java Programming?

Answer :

If a thread attempts to execute a synchronized method or synchronized statement and is unable to acquire an object's lock, it enters the waiting state until the lock becomes available.

Question 15. Can You Please Explain The Difference Between Font And Fontmetrics Classes In Java Programming?

Answer :

The FontMetrics class is used to define implementation-specific properties, such as ascent and descent, of a Font object.

Question 16. What Are Peerless Components?

Answer :

The peerless components are called light weight components.

Question 17. Which Package Has The Light Weight Components In Java Programming?

Answer :

javax.Swing package. All components in Swing, except JApplet, JDialog, JFrame and JWindow are lightweight components in Java Programming.

Question 18. When An Object Reference Be Cast To An Interface Reference In Java Programming?

Answer :

An object reference be cast to an interface reference when the object implements the referenced interface.

Question 19. Can You Please Explain The Difference Between Window And A Frame In Java Programming?

Answer :

The Frame class extends Window to define a main application window that can have a menu bar.

Question 20. Can You Please Explain The Difference Between Static And A Non-static Inner Class In Java Programming?

Answer :

A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

Question 21. Described Object's Lock And Which Object's Have Locks In Java Programming?

Answer :

An object's lock is a mechanism that is used by multiple threads to obtain synchronized access to the object. A thread may execute a synchronized method of an object only after it has acquired the object's lock. All objects and classes have locks. A class's lock is acquired on the class's Class object.

Question 22. Described Abstract Method In Java Programming?

Answer :

An abstract method is a method whose implementation is deferred to a subclass in Java Programming.

Question 23. What Are High-level Thread States In Java Programming?

Answer :

The high-level thread states are ready, running, waiting, and dead.

Question 24. What Must Class Do To Implement An Interface In Java Programming?

Answer :

It must provide all of the methods in the interface and identify the interface in its implements clause.

Question 25. Described The Purpose Of The Wait(), Notify(), And Notifyall() Methods In Java Programming?

Answer :

The wait(),notify(), and notifyAll() methods are used to provide an efficient way for threads to communicate each other in Java Programming.

Question 26. What Is The Purpose Of Finally Clause Of A Try-catch-finally Statement In Java Programming?

Answer :

The finally clause is used to provide the capability to execute code no matter whether or not an exception is thrown or caught.

Question 27. Described Locale Class In Java Programming?

Answer :

The Locale class is used to tailor program output to the conventions of a particular geographic, political, or cultural region.

Question 28. What Is The Purpose Of Runtime Class In Java Programming?

Answer :

The purpose of the Runtime class is to provide access to the Java runtime system in Java Programming.

Question 29. Described The Purpose Of System Class In Java Programming?

Answer :

The purpose of the System class is to provide access to system resources in Java Programming.

Question 30. Which Container Method Is Used To Cause Container To Be Laid Out And Redisplayed In Java Programming?

Answer :

validate(); : Container method is used to cause a container to be laid out and redisplayed in Java Programming.

Question 31. Described Properties Class In Java Programming?

Answer :

The properties class is a subclass of Hashtable that can be read from or written to a stream. It also provides the capability to specify a set of default values to be used.

Question 32. Described Gregorian Calendar Class In Java Programming?

Answer :

The GregorianCalendar provides support for traditional Western calendars in Java Programming.

Question 33. What Is Simpletimezone Class In Java Programming?

Answer :

The SimpleTimeZone class provides support for a Gregorian calendar in Java Programming.

Question 34. Which Invokes A Thread's Run() Method In Java Programming?

Answer :

After a thread is started, via its start() method or that of the Thread class, the JVM invokes the thread's run() method when the thread is initially executed.

Question 35. Can Anonymous Class Be Declared As Implementing An Interface And Extending A Class In Java Programming?

Answer :

An anonymous class may implement an interface or extend a superclass, but may not be declared to do both.

Question 36. What Is Purpose Of Finalization In Java Programming?

Answer :

The purpose of finalization is to give an unreachable

object the opportunity to perform any cleanup processing before the object is garbage collected.

Question 37. Which Class Is Superclass For Every Class In Java Programming?

Answer :

Object class is the superclass for every class in Java Programming.

Question 38. Which Component Subclasses That Support Painting In Java Programming?

Answer :

The Canvas, Frame, Panel, and Applet classes support painting.

Question 39. Described Native Method In Java Programming?

Answer :

A native method is a method that is implemented in a language other than Java.

Question 40. How To Write A Loop Indefinitely In Java Programming?

Answer :

for(;;)-for loop; while(true)-always true, etc.

Question 41. How Many Bits Are Used To Represent The Unicode, Ascii, Utf-16, And Utf-8 Characters In Java Programming?

Answer :

Unicode requires 16 bits and ASCII require 7 bits. Although the ASCII character set uses only 7 bits, it is usually represented as 8 bits. UTF-8 represents characters using 8, 16, and 18 bit patterns. UTF-16 uses 16-bit and larger bit patterns.

Question 42. Can You Please Explain The Difference Between Yielding And Sleeping In Java Programming?

Answer :

When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

Question 43. Is The Size Of A Keyword In Java Programming?

Answer :

The sizeof operator is not a keyword in Java Programming.

Question 44. Described Wrapped Classes In Java Programming?

Answer :

Wrapped classes are classes that allow primitive types to be accessed as objects.

Question 45. Does Java Handle Integer Overflows And Underflows?

Answer :

It uses those low order bytes of the result that can fit into the size of the type allowed by the operation.

Question 46. What Is Vector Class In Java Programming?

Answer :

The Vector class in Java Programming provides the capability to implement a growable array of objects.

Question 47. Which Modifiers May Be Used With An Inner Class That Is A Member Of An Outer Class In Java Programming?

Answer :

A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

Question 48. Suppose If A Method Is Declared As Protected, Where May The Method Be Accessed In Java Programming?

Answer :

A protected method may only be accessed by classes or interfaces of the same package or by subclasses of the class in which it is declared.

Question 49. What Is Iterator Interface In Java Programming?

Answer :

The Iterator interface is used to step through the elements of a Collection.

Question 50. Which Containers Is Use A Border Layout As Their Default Layout?

Answer :

The window, Frame and Dialog classes use a border layout as their default layout in Java Programming.

Question 51. Which Method Is Used To Specify A Container's Layout In Java Programming?

Answer :

The setLayout() method is used to specify a container's layout in Java Programming.

Question 52. Which State Does A Thread Enter When It Terminates Its Processing In Java Programming?

Answer :

When a thread terminates its processing, it enters the dead state.

Question 53. What Is Collections Api In Java Programming?

Answer :

The Collections API is a set of classes and interfaces that support operations on collections of objects.

Question 54. Define Llist Interface In Java Programming?

Answer :

The List interface provides support for ordered collections of objects.

Question 55. Can A Lock Be Acquired On Class In Java Programming?

Answer :

Yes, a lock can be acquired on a class. This lock is acquired on the class's Class object.

Question 56. Which Containers Use Flowlayout As Their Default Layout In Java Programming?

Answer :

The Panel and Applet classes use the FlowLayout as their default layout.

Question 57. Described Transient Variable?

Answer :

A transient variable is a variable that may not be serialized.

Question 58. What's New With Stop(), Suspend() And Resume() Methods In Jdk 1.2?

Answer :

The stop(), suspend() and resume() methods have been deprecated in JDK 1.2.

Question 59. Define Preferred Size Of A Component In Java Programming?

Answer :

The preferred size of a component is the minimum component size that will allow the component to display normally.

Question 60. Which Three Ways In Which A Thread Can Enter The Waiting State In Java Programming?

Answer :

A thread can enter the waiting state by invoking its sleep() method, by blocking on I/O, by unsuccessfully attempting to acquire an object's lock, or by invoking an object's wait() method. It can also enter the waiting state by invoking its (deprecated) suspend() method.

Question 61. Described Synchronized Methods And Synchronized Statements In Java Programming?

Answer :

Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.

Question 62. Described Synchronization And Why Is It Important In Java Programming?

Answer :

With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often causes dirty data and leads to

significant errors.

Question 63. How Observer And Observable Used In Java Programming?

Answer :

Objects that subclass the Observable class maintain a list of observers. When an Observable object is updated it invokes the update() method of each of its observers to notify the observers that it has changed state. The Observer interface is implemented by objects that observe Observable objects.

****
***

AGILE METHODOLOGY INTERVIEW QUESTIONS & ANSWERS

Question 1. An Agile Approach Advocates Which Of The Following Approaches?

Answer :

Get something business-valuable delivered as quickly as possible, consistent with the right level of quality.

Question 2. What Is Agile Testing?

Answer :

Agile Testing is testing practice that follows the principles of agile software development. Agile testing involves all members of an agile team with special skills and expertise to ensure business value is delivered at frequent intervals.

Question 3. In What Way Does The Agile Testing /development Methodology Differs From The Other Testing /development Methodologies?

Answer :

Anytime applying agile methodology, the testers /developers ensure that the whole process of testing /development is broke into as small steps as possible and just a small unit of code is tested /developed in each of this steps. The team of testers /developers is communicating consistently the results of their work, and changes the short term strategy and even the development plan on the go, based on the results of agile testing. Agile methodology encourages flexible and rapid response to change which should lead to a better end result.

Question 4. How Is It Different To Traditional Waterfall Or The V Model?

Answer :

The big difference is that in agile environment, testing is not a phase; it is an activity parallel to development.

In agile environment, small features of software are delivered frequently, so testing activity should be parallel to development activity. Testing time is short as we are only testing small features.
In the waterfall model, there is a testing phase at the end of the development so, testing is a big effort done after the whole application is developed. Testing time is long as we have to test the whole application.
Question 5. What Are The Different Methodologies In Agile Testing?

Answer :

There are various methods present in agile testing such as

Scrum
Crystal Methodologies
DSDM (Dynamic Software Development Method)
Feature driven development (FDD)
Lean software development
Extreme Programming (XP)

Question 6. What Are The Benefits Of Agile Software Development?

Answer :

Agile methods grew out of the real-life project experiences of leading software professionals who had experienced the challenges and limitations of traditional waterfall development on project after project. The approach promoted by agile development is in direct response to the issue associated with traditional software development – both in terms of overall philosophy as well as specific processes.

Question 7. What Is The Agile Manifesto?

Answer :

The agile software development emphasizes on four core values:

Individual and team interactions over processes and tools.
Working software over comprehensive documentation.
Customer collaboration over contract negotiation.
Responding to change over following a plan.
Question 8. What Are Some Of The Key Features Of Agile Development?

Answer :

Some of the key features of agile development are

Collective code ownership and freedom to change.
Incremental approach (e.g. user stories are incrementally implemented). Automation (e.g. TDD -- Test Driven Development).
Customer focused (for e.g. internal and external users and business analysts are your immediate customers).
Design must be simple.
Designing is an ongoing activity with constant re-factoring to achieve the rules of code simplicity like no duplication, verified by automated tests, separation of responsibilities, and minimum number of classes, methods, and lines.
Question 9. What Is Scrum?

Answer :

Scrum is an innovative approach to getting work done in efficient way. It is iterative & incremental agile software development method. These iterations are time boxed with various iterations & each iteration is called Sprint. According to latest surveys Scrum is the most popular agile project management methodology in software development. The term Scrum is formed from Rugby.

Scrum is ideally used where highly emergent or rapidly changing requirements. Scrum is basically worked on a self-organizing, cross-functional team. In the overall scrum team there is no team leader who assign the task to team rather whole scrum members work as a team & they decides the task on which they will work on. Also the problem will be resolve by team.

Question 10. What Are The Three Main Roles In Scrum?

Answer :

The Scrum team consists of three main roles:

Product Owner: Manages the product backlog. PO is the voice of the business and create new features to be developed for the application.
Scrum Master: Responsible for managing the sprint, remove any impediments and keeps track of the progress of the project.
Scrum Team itself: Composed of developers, designers and QA. This forms the team which is responsible for delivering high quality software.

Question 11. What Is Sprint?

Answer :

Sprint is a predefined interval or the time frame in which the work has to be completed and make it ready for review or ready for production deployment. This time box usually lies between 2 weeks to 1 month. In our day to day life

when we say that we follow 1 month Sprint cycle, it simply means that we work for one month on the tasks and make it ready for review by the end of that month.

Question 12. Which Of The Following Best Describes The Approach For Determining The Iteration Length?

Answer :

The team should agree on the length of the iteration taking the size and complexity of the project into consideration.

Question 13. As Product Owner, Which Of The Listed Guidelines Should You Follow While Creating The Product Backlog?

Answer :

Split high priority customer requirements that are too big into multiple smaller stories.
Update the Product Backlog whenever the customer or development team comes up with new requirements.
Prioritize the requirements based on business value and risk.
Question 14. Mention The Key Difference Between Sprint Backlog And Product Backlog?

Answer :

Product backlog: It contains a list of all desired features and is owned by the product owner.
Sprint backlog: It is a subset of the product backlog owned by development team and commits to deliver it in a sprint. It is created in Sprint Planning Meeting.
Question 15. Which Scrum Artifact Helps In Tracking Progress?

Answer :

Sprint burn down chart.

Question 16. Scrum Master Is Responsible For?

Answer :

The Scrum process being adopted and used properly.

Question 17. Which Of These Advocates The Anti-gold-plating Mechanism Of Agile?

Answer :

Simplicity–the art of maximizing the amount of work not done–is essential.

Question 18. What Is A Test Stub?

Answer :

A test stub is a bit of code that replaces an undeveloped or fully developed component within a system being tested. The test stub is built such that it mimics the actual component by generating specific known outputs. The stub can be used as a substitute for the actual (fully developed) component for testing purposes. The stub can also be used during testing to isolate system components and troubleshoot problems. A test stub is also known as a test double.

Question 19. Describe A Situation When You Used Agile Methodology In Your Work, Or When You Belonged To The Team That Applied It Successfully?

Answer :

Mention the situation, the challenges you faced, and how agile approach helped to successfully achieve your goal. Try to talk about team work, and about the end result of your efforts–and how it benefited the employer, or the final product

If you can not find anything to talk about from your professional career, think about something from personal life–though we do not typically realize it, we apply agile methodology in many everyday situations and interactions with other people.

Question 20. Explain In Agile, Burn-up And Burn-down Chart?

Answer :

To track the project progress burn up and burn down charts are used

Burn up Chart: It shows the progress of stories done over time.
Burn down Chart: It shows how much work was left to do overtime.

Question 21. What Is Dynamic Software Development Method (dsdm)?

Answer :

DSDM is a Rapid Application Development (RAD) approach to software development and provides an agile project delivery framework. The important aspect of DSDM is that the users are required to be involved actively, and the teams are given the power to make decisions. Frequent delivery of product becomes the active focus with DSDM.

Question 22. Main Characteristics Of Agile Methodology?

Answer :

Every person can approach the question from their own view, and perhaps a software tester would give a different answer than a project manager.

Nevertheless, you can list cross-functional team composition, face-to face communication, solving problems immediately after these are identified, and working solution as a primary metric of progress, as the main characteristics of agile development.

Question 23. Can We Apply Agile Methodology To Other Areas And Projects?

Answer :

Employers try to understand if you see the real benefits of agile, and the practical application for various areas of

their business.

To say that the methodology can be (and perhaps even should be) applied anytime we have insufficient entry data, or when we work in an unknown area, or simply within a small team, or when many unpredictable variables play the role in the final outcome, would be a good answer. Bio-medicine, biochemistry or physics belong to the fields where we apply agile methodology ever more frequently.

Question 24. In What Does The Agile Testing (development) Methodology Differ From The Other Testing (development) Methodologies?

Answer :

Anytime applying agile methodology, the testers (developers) ensure that the whole process of testing (development) is broke into as small steps as possible and just a small unit of code is tested (developed) in each of this steps. The team of testers (developers) is communicating consistently the results of their work, and change the short term strategy and even the development plan on the go, based on the results of agile testing. Agile methodology encourages flexible and rapid response to change which should lead to a better end result.

Question 25. What Is Re-factoring?

Answer :

Re-factoring is modifying existing code to improve its performance, readability, extensibility etc. The code’s functionality remains as it is.

Question 26. What Are The Two Key Factors When Working As A Qa In An Agile Team?

Answer :

QA can add a lot of value to an agile team because of the different mindset. Testers can and should think about the different possible scenarios to test a story.

However the most important asset that they can bring is:

To prevent defect: A should advocate best practices along the way to prevent defects from entering the system in the first place.
To provide fast feedback: It is important for developers to know if the new functionality works as expected and if regression tests pass, and they need that feedback quite quickly. QA should provide the results of the tests to developers as soon as possible.
Question 27. What Are The Disadvantages Of Agile Models?

Answer :

In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.

There is lack of emphasis on necessary designing and documentation.
The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

****
****
ACCENTURE JAVA INTERVIEW QUESTIONS & ANSWERS

Question 1. Why Do You Want To Work In This Industry / Company?

Answer :

First you should try to convince that this company gives huge opportunity in many aspect i.e. new technologies implementation, the policy of company suits you like professionalism. Also you can mention that you are big fan of this company and its your dream company. Basically show your all positive attitude towards company.

Question 2. Which Location Do You Want To Work In And Why?

Answer :

Give your own choice. Also mention a valid reason for why you are interested for that location. The reason should be always positive and clear. Example :- you can support your family from this location,

Question 3. Describe A Problem You Faced And How You Deal With It ?

Answer :

You can describe any issue you faced during your project work in the organization. And what the solution you have implemented for that issue.

Question 4. What Are The Types Of Class Loaders In Java?

Answer :

As per my knowledge there are basically 3 types of class loader like bootstarp classloader,extension class loader and system class loader.

Bootstrap Class Loader
Bootstrap class loader loads java's core classes like java.lang, java.util etc. These are classes that are part of java runtime environment. Bootstrap class loader is native implementation and so they may differ across different JVMs.
Extensions Class Loader
JAVA_HOME/jre/lib/ext contains jar packages that are extensions of standard core java classes. Extensions class loader loads classes from this ext folder. Using the system environment propery java.ext.dirs you can add 'ext' folders and jar files to be loaded using extensions class loader
System Class Loader
Java classes that are available in the java classpath are loaded using System class loader
Question 5. How To Read And Write Image From A File ?

Answer :

You can use ImageIo.read() and ImageIO.write() method of javax.imageio package.

Question 6. How Concurrenthashmap Works?

Answer :

The basic design of ConcurrentHashMap is to handling threading. Basically it locks each of the box (by default 16) which can be locked independently and thread safe for operation. And it does not expose the internal lock process.

Question 7. Can A Static Block Throw Exception?

Answer :

Yes. We can throw checked exception.

Question 8. What Is Difference Between Iterator Access And Index Access?

Answer :

Basically iterator access process the traverse operation through each element, where index access process access direct the element by using the index.

Question 9. Why Character Array Is Better Than String For Storing Password In Java?

Answer :

Because, character array stores data in encrypted format which is not readable by human. But,the string stores the data in human readable format which is not secure.

Question 10. What Is Daemon Thread In Java ?

Answer :

A daemon thread is normally runs on background. And it does

not prevent the JVM from exiting when the program finishes but the thread is still running.

Question 11. What Is Java Reflection Api?

Answer :

Reflection is one of the most powerful api which help to work with classes, methods and variables dynamically. Basically it inspect the class attributes at runtime. Also we can say it provides a metadata about the class.

Question 12. What Is The Difference Between Serializable And Externalizable Interfaces?

Answer :

Both interfaces are used for implement serialization. But, the basic difference is Serializable interface does not have any method (it’s a marker interface) and Externalizable interface having 2 methods such as readExternal() and writeExternal(). Serializable interface is the super interface for Externalizable interface.

Question 13. What Is An Abstract Method?

Answer :

An abstract method is a method whose implementation is deferred to a subclass.

Question 14. What Value Does Read() Return When It Has Reached The End Of A File?

Answer :

The read() method returns -1 when it has reached the end of a file.

Question 15. Can A Byte Object Be Cast To A Double Value?

Answer :

No, an object cannot be cast to a primitive value.

Question 16. What Is The Difference Between A Static And A Non-static Inner Class?

Answer :

A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

Question 17. What Is An Object's Lock And Which Object's Have Locks?

Answer :

An object's lock is a mechanism that is used by multiple threads to obtain synchronized access to the object. A thread may execute a synchronized method of an object only after it has acquired the object's lock. All objects and classes have locks. A class's lock is acquired on the class's Class object.

Question 18. What Is The % Operator?

Answer :

It is referred to as the modulo or remainder operator. It returns the remainder of dividing the first operand by the second operand.

Question 19. When Can An Object Reference Be Cast To An Interface Reference?

Answer :

An object reference be cast to an interface reference when the object implements the referenced interface.

Question 20. Which Class Is Extended By All Other Classes?

Answer :

The Object class is extended by all other classes.

Question 21. Which Non-unicode Letter Characters May Be Used As The First Character Of An Identifier?

Answer :

The non-Unicode letter characters $ and _ may appear as the first character of an identifier.

Question 22. What Restrictions Are Placed On Method Overloading?

Answer :

Two methods may not have the same name and argument list but different return types.

Question 23. What Is Transient Variable?

Answer :

Transient variable can’t be serialize. For example if a variable is declared as transient in a Serializable class and the class is written to an ObjectStream, the value of the variable can’t be written to the stream instead when the class is retrieved from the ObjectStream the value of the variable becomes null.

Question 24. What Is Collection Api?

Answer :

The Collection API is a set of classes and interfaces that support operation on collections of objects. These classes and interfaces are more flexible, more powerful, and more regular than the vectors, arrays, and hashtables if effectively replaces.

Example of classes: HashSet, HashMap, ArrayList, LinkedList, TreeSet and TreeMap.

Example of interfaces: Collection, Set, List and Map.

Question 25. What Is Casting?

Answer :

There are two types of casting, casting between primitive numeric types and casting between object references. Casting between numeric types is used to convert larger values, such as double values, to smaller values, such as byte values. Casting between object references is used to refer to an object by a compatible class, interface, or array type reference.

Question 26. What Is The Return Type Of A Program’s Main() Method?

Answer :

void.

Question 27. If A Variable Is Declared As Private, Where May The Variable Be Accessed?

Answer :

A private variable may only be accessed within the class in which it is declared.

Question 28. What Do You Understand By Private, Protected And Public?

Answer :

These are accessibility modifiers. Private is the most restrictive, while public is the least restrictive. There is no real difference between protected and the default type (also known as package protected) within the context of the same package, however the protected keyword allows visibility to a derived class in a different package.

Question 29. What Is Downcasting ?

Answer :

Downcasting is the casting from a general to a more

specific type, i.e. casting down the hierarchy.

Question 30. What Modifiers May Be Used With An Inner Class That Is A Member Of An Outer Class?

Answer :

A (non-local) inner class may be declared as public, protected, private, static, final, or abstract.

Question 31. How Many Bits Are Used To Represent Unicode, Ascii, Utf-16, And Utf-8 Characters?

Answer :

Unicode requires 16 bits and ASCII require 7 bits. Although the ASCII character set uses only 7 bits, it is usually represented as 8 bits. UTF-8 represents characters using 8, 16, and 18 bit patterns. UTF-16 uses 16-bit and larger bit patterns.

Question 32. What Restrictions Are Placed On The Location Of A Package Statement Within A Source Code File?

Answer :

A package statement must appear as the first line in a source code file (excluding blank lines and comments).

****
****

MONGODB INTERVIEW QUESTIONS & ANSWERS

Question 1. What Are Nosql Databases? What Are The Different Types Of Nosql Databases?

Answer :

A NoSQL database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases (like SQL, Oracle, etc.).

Types of NoSQL databases:

Document Oriented
Key Value
Graph
Column Oriented
Question 2. What Kind Of Nosql Database Mongodb Is?

Answer :

MongoDB is a document oriented database. It stores data in the form of BSON structure based documents. These documents are stored in a collection.

Question 3. Which Are The Most Important Features Of Mongodb?

Answer :

Flexible data model in form of documents
Agile and highly scalable database
Faster than traditional databases
Expressive query language
Question 4. What Is A Namespace In Mongodb?

Answer :

A Namespace is the concatenation of the database name and collection name. For e.g. school.students with school as the database and students as the collection.

Question 5. Which All Languages Can Be Used With Mongodb?

Answer :

Currently, MonggoDB provides official driver support for C, C++, C#, Java, Node.js, Perl, PHP, Python, Ruby, Scala, Go and Erlang. MongoDB can easily be used with any of these languages. There are some other community supported drivers too but the above mentioned ones are officially provided by MongoDB.

Question 6. Compare Sql Databases And Mongodb At A High Level.?

Answer :

SQL databases store data in form of tables, rows, columns and records. This data is stored in a pre-defined data model which is not very much flexible for today's real-world highly growing applications. MongoDB in contrast uses a flexible structure which can be easily modified and extended.

Question 7. How Is Mongodb Better Than Other Sql Databases?

Answer :

MongoDB allows a highly flexible and scalable document structure. For e.g. one data document in MongoDB can have five columns and the other one in the same collection can have ten columns. Also, MongoDB database are faster as compared to SQL databases due to efficient indexing and storage techniques.

Question 8. Compare Mongodb And Couchdb At High Level.?

Answer :

Although both of these databases are document oriented, MongoDB is a better choice for applications which need dynamic queries and good performance on a very big database. On the other side, CouchDB is better used for applications with occasionally changing queries and pre-defined queries.

Question 9. Does Mongodb Support Foreign Key Constraints?

Answer :

No. MongoDB does not support such relationships.

Question 10. Does Mongodb Support Acid Transaction Management And Locking Functionalities?

Answer :

No. MongoDB does not support default multi-document ACID transactions. However, MongoDB provides atomic operation on a single document.

Question 11. How Can You Achieve Primary Key – Foreign Key Relationships In Mongodb?

Answer :

By default MongoDB does not support such primary key – foreign key relationships. However, we can achieve this concept by embedding one document inside another. Foe e.g. an address document can be embedded inside customer document.

Question 12. Does Mongodb Need A Lot Of Ram?

Answer :

No. MongoDB can be run even on a small amount of RAM. MongoDB dynamically allocates and de-allocates RAM based on the requirements of other processes.

Question 13. Does Mongodb Pushes The Writes To Disk Immediately Or Lazily?

Answer :

MongoDB pushes the data to disk lazily. It updates the immediately written to the journal but writing the data from journal to disk happens lazily.

Question 14. Explain The Structure Of Objectid In Mongodb.?

Answer :

ObjectID is a 12-byte BSON type with:

4 bytes value representing seconds
3 byte machine identifier
2 byte process id
3 byte counter

Question 15. Mongodb Uses Bson To Represent Document Structures. True Or False?

Answer :

True

Question 16. If You Remove A Document From Database, Does Mongodb Remove It From Disk?

Answer :

Yes. Removing a document from database removes it from disk too.

Question 17. Mention The Command To Insert A Document In A Database Called School And Collection Called Persons.?

Answer :

```
use school;
db.persons.insert( { name: "kadhir", dept: "CSE" } )
```

Question 18. What Are Indexes In Mongodb?

Answer :

Indexes support the efficient execution of queries in MongoDB. Without indexes, MongoDB must perform a collection scan, i.e. scan every document in a collection, to select those documents that match the query statement. If an appropriate index exists for a query, MongoDB can use the index to limit the number of documents it must inspect.

Question 19. How Many Indexes Does Mongodb Create By Default For A New Collection?

Answer :

By default, MongoDB created the _id collection for every collection.

Question 20. Can You Create An Index On An Array Field In

Mongodb? If Yes, What Happens In This Case?

Answer :

Yes. An array field can be indexed in MongoDB. In this case, MongoDB would index each value of the array.

Question 21. What Is A Covered Query In Mongodb?

Answer :

A covered query is the one in which:

fields used in the query are part of an index used in the query, and the fields returned in the results are in the same index.

Question 22. Why Is A Covered Query Important?

Answer :

Since all the fields are covered in the index itself, MongoDB can match the query condition as well as return the result fields using the same index without looking inside the documents. Since indexes are stored in RAM or sequentially located on disk, such access is a lot faster.

Question 23. Does Mongodb Provide A Facility To Do Text Searches? How?

Answer :

Yes. MongoDB supports creating text indexes to support text search inside string content. This was a new feature which can introduced in version 2.6.

Question 24. What Happens If An Index Does Not Fit Into Ram?

Answer :

If the indexes do not fit into RAM, MongoDB reads data from disk which is relatively very much slower than reading from RAM.

Question 25. Mention The Command To List All The Indexes On A Particular Collection.?

Answer :

db.collection.getIndexes()

Question 26. At What Interval Does Mongodb Write Updates To The Disk?

Answer :

By default configuration, MongoDB writes updates to the disk every 60 seconds. However, this is configurable with the commitIntervalMs and syncPeriodSecs options.

Question 27. How Can You Achieve Transaction And Locking In Mongodb?

Answer :

To achieve concepts of transaction and locking in MongoDB, we can use the nesting of documents, also called embedded documents. MongoDB supports atomic operations within a single document.

Question 28. What Is Aggregation In Mongodb?

Answer :

Aggregations operations process data records and return computed results. Aggregation operations group values from multiple documents together, and can perform a variety of operations on the grouped data to return a single result. MongoDB provides three ways to perform aggregation: the aggregation pipeline, the map-reduce function, and single purpose aggregation methods and commands.

Question 29. What Is Sharding In Mongodb? Explain.?

Answer :

Sharding is a method for storing data across multiple

machines. MongoDB uses sharding to support deployments with very large data sets and high throughput operations.

Question 30. What Is Replication In Mongodb?

Answer :

Replication is the process of synchronizing data across multiple servers. Replication provides redundancy and increases data availability. With multiple copies of data on different database servers, replication protects a database from the loss of a single server. Replication also allows you to recover from hardware failure and service interruptions.

Question 31. What Are Primary And Secondary Replica Sets?

Answer :

Primary and master nodes are the nodes that can accept writes. MongoDB's replication is 'single-master:' only one node can accept write operations at a time.

Secondary and slave nodes are read-only nodes that replicate from the primary.

Question 32. By Default, Mongodb Writes And Reads Data From Both Primary And Secondary Replica Sets. True Or False.

Answer :

False. MongoDB writes data only to the primary replica set.

Question 33. Why Are Mongodb Data Files Large In Size?

Answer :

MongoDB preallocate data files to reserve space and avoid file system fragmentation when you setup the server.

Question 34. When Should We Embed One Document Within Another In Mongodb?

Answer :

You should consider embedding documents for:

'contains' relationships between entities
One-to-many relationships
Performance reasons
Question 35. Why Mongodb Is Not Preferred Over A 32-bit System?

Answer :

When running a 32-bit build of MongoDB, the total storage size for the server, including data and indexes, is 2 gigabytes. For this reason, do not deploy MongoDB to production on 32-bit machines.
If you're running a 64-bit build of MongoDB, there's virtually no limit to storage size.

Question 36. What Is A Storage Engine In Mongodb?

Answer :

A storage engine is the part of a database that is responsible for managing how data is stored on disk. For example, one storage engine might offer better performance for read-heavy workloads, and another might support a higher-throughput for write operations.

Question 37. Which Are The Two Storage Engines Used By Mongodb?

Answer :

MongoDB uses MMAPv1 and WiredTiger.

Question 38. What Is The Role Of A Profiler In Mongodb? Where Does The Writes All The Data?

Answer :

The database profiler collects fine grained data about MongoDB write operations, cursors, database commands on a running mongod instance. You can enable profiling on a per-database or per-instance basis.

The database profiler writes all the data it collects to the system.profile collection, which is a capped collection.

Question 39. How Does Journaling Work In Mongodb?

Answer :

When running with journaling, MongoDB stores and applies write operations in memory and in the on-disk journal before the changes are present in the data files on disk. Writes to the journal are atomic, ensuring the consistency of the on-disk journal files. With journaling enabled, MongoDB creates a journal subdirectory within the directory defined by dbPath, which is /data/db by default.

Question 40. Mention The Command To Check Whether You Are On The Master Server Or Not.?

Answer :

db.isMaster()

Question 41. Can You Configure The Cache Size For Mmapv1? How?

Answer :

No. MMAPv1 does not allow configuring the cache size.

Question 42. How Does Mongodb Provide Concurrency?

Answer :

MongoDB uses reader-writer locks that allow concurrent readers shared access to a resource, such as a database or collection, but give exclusive access to a single write operation.

Question 43. How Can You Isolate Your Cursors From Intervening With The Write Operations?

Answer :

You can use the snapshot() method on a cursor to isolate the operation for a very specific case. snapshot() traverses the index on the _id field and guarantees that the query will return each document no more than once.

Question 44. Can One Mongodb Operation Lock More Than One Databases? If Yes, How?

Answer :

Yes. Operations like copyDatabase(), repairDatabase(), etc. can lock more than onne databases involved.

Question 45. How Can Concurrency Affect Replica Sets Primary?

Answer :

In replication, when MongoDB writes to a collection on the primary, MongoDB also writes to the primary's oplog, which is a special collection in the local database. Therefore, MongoDB must lock both the collection's database and the local database.

Question 46. What Is Gridfs?

Answer :

GridFS is a specification for storing and retrieving files that exceed the BSON-document size limit of 16MB. Instead of storing a file in a single document, GridFS divides a file into parts, or chunks, and stores each of those chunks as a separate document.

Question 47. Can You Run Multiple Javascript Operations In A Single Mongod Instance?

Answer :

Yes. The V8 JavaScript engine added in 2.4 allows multiple JavaScript operations to run at the same time.

Question 48. Which Command Can Be Used To Provide Various

Information On The Query Plans Used By A Mongodb Query?

Answer :

The explain() command can be used for this information. The possible modes are: 'queryPlanner', 'executionStats', and 'allPlansExecution'.

****

SQL INTERVIEW QUESTIONS & ANSWERS

Question 1. Explain What Is A Database?

Answer :

A database is a collection of information in an organized form for faster and better access, storage and manipulation. It can also be defined as a collection of tables, schema, views and other database objects.

Question 2. Explain What Is Dbms?

Answer :

Database Management System is a collection of programs that enables a user to store, retrieve, update and delete information from a database.

Question 3. Explain What Is Rdbms?

Answer :

RDBMS stands for Relational Database Management System. RDBMS is a database management system (DBMS) that is based on the relational model. Data from relational database can be accessed using Structured Query Language (SQL)

Question 4. What Are The Popular Database Management Systems In The It Industry?

Answer :

Oracle, MySQL, Microsoft SQL Server, PostgreSQL, Sybase, MongoDB, DB2, and Microsoft Access etc.,

Question 5. Explain What Is Sql?

Answer :

SQL stands for Structured Query Language. It is an American National Standard Institute (ANSI) standard. It is a standard language for accessing and manipulating databases. Using SQL, some of the action we could do are to create databases, tables, stored procedures (SP's), execute queries, retrieve, insert, update, delete data against a database.

Question 6. Explain What Is Table In A Database?

Answer :

A table is a database object used to store records in a field in the form of columns and rows that holds data.

Question 7. Explain What Is A Field In A Database And Record In A Database?

Answer :

A field in a Database table is a space allocated to store a particular record within a table.
A record (also called a row of data) is an ordered set of related data in a table.
Question 8. What Is The Use Of Nvl Function?

Answer :

NVL function is used to convert the null value to its actual value.

Question 9. Explain What Is A Column In A Table?

Answer :

A column is a vertical entity in a table that contains all information associated with a specific field in a table.

Question 10. What Are The Different Types Of Sql Commands?

Answer :

SQL commands are segregated into following types:

DDL – Data Definition Language
DML – Data Manipulation Language
DQL – Data Query Language
DCL – Data Control Language
TCL – Transaction Control Language

Question 11. What Are The Different Ddl Commands In Sql?

Answer :

DDL commands are used to define or alter the structure of the database.

CREATE: To create databases and database objects

ALTER: To alter existing database objects

DROP: To drop databases and databases objects

TRUNCATE: To remove all records from a table but not its database structure

RENAME: To rename database objects

Question 12. What Are The Different Dml Commands In Sql?

Answer :

DML commands are used for managing data present in the database.

SELECT: To select specific data from a database

INSERT: To insert new records into a table

UPDATE: To update existing records

DELETE: To delete existing records from a table

Question 13. What Are The Different Dcl Commands In Sql?

Answer :

DCL commands are used to create roles, grant permission and control access to the database objects.

GRANT: To provide user access

DENY: To deny permissions to users

REVOKE: To remove user access

Question 14. What Are The Different Tcl Commands In Sql?

Answer :

TCL commands are used to manage the changes made by DML statements.

COMMIT: To write and store the changes to the database

ROLLBACK: To restore the database since the last commit

Question 15. Explain What Is An Index?

Answer :

An index is used to speed up the performance of queries. It makes faster retrieval of data from the table. The index can be created on one column or a group of columns.

Question 16. Explain What Is A View?

Answer :

A view is like a subset of a table which is stored logically in a database. A view is a virtual table. It contains rows and columns similar to a real table. The

fields in the view are fields from one or more real tables. Views do not contain data of their own. They are used to restrict access to the database or to hide data complexity.

Question 17. Explain What Is A Subquery ?

Answer :

A Subquery is a SQL query within another query. It is a subset of a Select statement whose return values are used in filtering the conditions of the main query.

Question 18. What Is The Difference Between Rename And Alias?

Answer :

‘Rename’ is a permanent name given to a table or column

‘Alias’ is a temporary name given to a table or column.

Question 19. What Is A Join?

Answer :

Join is a query, which retrieves related columns or rows from multiple tables.

Question 20. What Are The Different Types Of Joins?

Answer :

Types of Joins are as follows:

INNER JOIN: It is also known as SIMPLE JOIN which returns all rows from BOTH tables when it has at least one column matched

Syntax: SELECT column_name(s)

FROM table_name1

INNER JOIN table_name2

ON column_name 1=column_name 2;

LEFT JOIN (LEFT OUTER JOIN): This join returns all rows from a LEFT table and its matched rows from a RIGHT table.

Syntax: SELECT column_name(s)

FROM table_name1

LEFT JOIN table_name2

ON column_name 1=column_name 2;

RIGHT JOIN (RIGHT OUTER JOIN): This joins returns all rows from the RIGHT table and its matched rows from a LEFT table.

Syntax: SELECT column_name(s)

FROM table_name1

RIGHT JOIN table_name2

ON column_name1=column_name2;

FULL JOIN (FULL OUTER JOIN): This joins returns all when there is a match either in the RIGHT table or in the LEFT table.

Syntax: SELECT column_name(s)

FROM table_name1

FULL OUTER JOIN table_name2

ON column_name1=column_name2;

Question 21. What Are Sql Constraints?

Answer :

SQL constraints are the set of rules that enforced some restriction while inserting, deleting or updating of data

in the databases.

Question 22. What Are The Constraints Available In Sql?

Answer :

Some of the constraints in SQL are : Primary Key, Foreign Key, Unique Key, SQL Not Null, Default, Check and Index constraint.

Question 23. What Is A Unique Key And Primary Key And Foreign Key?

Answer :

A UNIQUE KEY constraint is used to ensure that there are no duplication values in the field/column.
A PRIMARY KEY constraint uniquely identifies each record in a database table. All columns participating in a primary key constraint must not contain NULL values.
A FOREIGN KEY is a key used to link two tables together. A FOREIGN KEY in a table is linked with the PRIMARY KEY of another table.
Question 24. What Is The Difference Between Unique And Primary Key Constraints?

Answer :

There should be only one PRIMARY KEY in a table whereas there can be any number of UNIQUE Keys.

PRIMARY KEY doesn’t allow NULL values whereas Unique key allows NULL values.

Question 25. What Is A Null Value?

Answer :

A field with a NULL value is a field with no value. A NULL value is different from a zero value or a field that contains spaces. A field with a NULL value is one that has been left blank during record creation. Assume, there is a field in a table is optional and it is possible to insert a record without adding a value to the optional field then

the field will be saved with a NULL value.

Question 26. What Is Normalization?

Answer :

Normalization is the process of table design to minimize the data redundancy.

There are different types of Noramalization forms in SQL:-

First Normal Form (1NF): It removes all duplicate columns from the table. Creates table for related data and identifies unique column values
First Normal Form (2NF): Follows 1NF and creates and places data subsets in an individual table and defines relationship between tables using primary key
Third Normal Form (3NF): Follows 2NF and removes those columns which are not related through primary key
Fourth Normal Form (4NF): Follows 3NF and do not define multi-valued dependencies. 4NF also known as BCNF
Question 27. What Is Stored Procedure?

Answer :

A Stored Procedure is a collection of SQL statements that have been created and stored in the database to perform a particular task. The stored procedure accepts input parameters and processes them and returns a single value such as a number or text value or a result set (set of rows).

Question 28. What Is A Trigger?

Answer :

A Trigger is a SQL procedure that initiates an action in response to an event (Insert, Delete or Update) occurs. When a new Employee is added to an Employee_Details table, new records will be created in the relevant tables such as Employee Payroll, Employee Time Sheet etc.,

Question 29. List Out The Acid Properties And Explain?

Answer :

Following are the four properties of ACID. These guarantees that the database transactions are processed reliably.

Atomicity
Consistency
Isolation
Durability

Question 30. What Is The Difference Between Delete, Truncate And Drop Command?

Answer :

The difference between the Delete, Truncate and Drop command is:

Delete command is a DML command, it is used to delete rows from a table. It can be rolled back.
Truncate is a DDL command, it is used to delete all the rows from the table and free the space containing the table. It cant be rolled back.
Drop is a DDL command, it removes the complete data along with the table structure(unlike truncate command that removes only the rows). All the tables' rows, indexes, and privileges will also be removed.

Question 31. What Is The Difference Between Having And Where Clause?

Answer :

Where clause is used to fetch data from a database that specifies particular criteria whereas a Having clause is used along with 'GROUP BY' to fetch data that meets particular criteria specified by the Aggregate functions. Where clause cannot be used with Aggregate functions, but the Having clause can.

Question 32. What Are Aggregate Functions In Sql?

Answer :

SQL aggregate functions return a single value, calculated from values in a column.

Some of the aggregate functions in SQL are as follows:

AVG() : This function returns the average value
COUNT() : This function returns the number of rows
MAX() : This function returns the largest value
MIN() : This function returns the smallest value
ROUND() : This function rounds a numeric field to the number of decimals specified
SUM() : This function returns the sum

Question 33. What Are String Functions In Sql?

Answer :

SQL string functions are used primarily for string manipulation.

Some of the widely used SQL string functions are:

LEN() : It returns the length of the value in a text field
LOWER() : It converts character data to lower case
UPPER() : It converts character data to upper case
SUBSTRING() : It extracts characters from a text field
LTRIM() : It is to remove all whitespace from the beginning of the string
RTRIM() : It is to remove all whitespace at the end of the string
CONCAT() : Concatenate function combines multiple character strings together
REPLACE() :To update the content of a string.

Question 34. Explain The Working Of Sql Privileges?

Answer :

SQL GRANT and REVOKE commands are used to implement privileges in SQL multiple user environments. The administrator of the database can grant or revoke privileges to or from users of database object like SELECT, INSERT, UPDATE, DELETE, ALL etc.

GRANT Command: This command is used provide database access to user apart from an administrator.

Syntax: GRANT privilege_name

ON object_name

TO {user_name|PUBLIC|role_name}

[WITH GRANT OPTION];

In above syntax WITH GRANT OPTIONS indicates that the user can grant the access to another user too.

REVOKE Command: This command is used provide database deny or remove access to database objects.

Syntax: REVOKE privilege_name

ON object_name

FROM {user_name|PUBLIC|role_name};

Question 35. How Many Types Of Privileges Are Available In Sql?

Answer :

There are two types of privileges used in SQL, such as

System Privilege: System privileges deal with an object of a particular type and specifies the right to perform one or more actions on it which include Admin allows a user to perform administrative tasks, ALTER ANY INDEX, ALTER ANY CACHE GROUP CREATE/ALTER/DELETE TABLE, CREATE/ALTER/DELETE VIEW etc.

Object Privilege: This allows to perform actions on an object or object of another user(s) viz. table, view, indexes etc. Some of the object privileges are EXECUTE, INSERT, UPDATE, DELETE, SELECT, FLUSH, LOAD, INDEX, REFERENCES etc.

Question 36. What Is Sql Injection?

Answer :

SQL Injection is a type of database attack technique where

malicious SQL statements are inserted into an entry field of database such that once it is executed the database is opened for an attacker. This technique is usually used for attacking Data-Driven Applications to have an access to sensitive data and perform administrative tasks on databases.

Question 37. What Is The Difference Between Clustered And Non-clustered Indexes?

Answer :

One table can have only one clustered index but multiple nonclustered indexes.
Clustered indexes can be read rapidly rather than non-clustered indexes.
Clustered indexes store data physically in the table or view and non-clustered indexes do not store data in table as it has separate structure from data row
Question 38. What Is Relationship? How Many Types Of Relationship Are There?

Answer :

The relationship can be defined as the connection between more than one tables in the database.

There are 4 types of relationships:

One to One Relationship
Many to One Relationship
Many to Many Relationship
One to Many Relationship
Question 39. What Is Collation?

Answer :

Collation is set of rules that check how the data is sorted by comparing it. Such as Character data is stored using correct character sequence along with case sensitivity, type, and accent.

Question 40. What Is Database White Box Testing And Black Box Testing?

Answer :

Database White Box Testing involves:

Database Consistency and ACID properties
Database triggers and logical views
Decision Coverage, Condition Coverage, and Statement Coverage
Database Tables, Data Model, and Database Schema
Referential integrity rules
Database Black Box Testing involves:

Data Mapping
Data stored and retrieved
Use of Black Box techniques such as Equivalence Partitioning and Boundary Value Analysis (BVA)
Question 41. What Are The Advantages Of Views?

Answer :

Advantages of Views:

Views restrict access to the data because the view can display selective columns from the table.

Views can be used to make simple queries to retrieve the results of complicated queries. For example, views can be used to query information from multiple tables without the user knowing.

Question 42. What Is Schema?

Answer :

A schema is a collection of database objects of a User.

Question 43. What Is The Difference Between Sql And Pl/sql?

Answer :

SQL is a structured query language to create and access databases whereas PL/SQL comes with procedural concepts of programming languages.

Question 44. What Is The Difference Between Sql And Mysql?

Answer :

SQL is a structured query language that is used for manipulating and accessing the relational database, on the other hand, MySQL itself is a relational database that uses SQL as the standard database language.

Question 45. What Is Sql Sandbox In Sql Server?

Answer :

SQL Sandbox is the safe place in SQL Server Environment where untrusted scripts are executed. There are 3 types of SQL sandbox, such as

Safe Access Sandbox: Here a user can perform SQL operations such as creating stored procedures, triggers etc. but cannot have access to the memory and cannot create files.

External Access Sandbox: User can have access to files without having a right to manipulate the memory allocation.

Unsafe Access Sandbox: This contains untrusted codes where a user can have access to memory

****
****

Question 1. Assuming That You Are Selected, What Will Be Your Strategy For Next 60 Days?
Answer :
If I am selected for this position, I ’ll use my initial 60 days in understanding my role carefully in terms of the contribution to the business and increasing the overall profitability. I ’ll sit with my line manager and other juniors to understand what has already been done and what its impact has been. From there on, I ’ll formulate my strategy to growth in close conjunction with managers and see that it is properly implemented.
Question 2. How Would You Improve Upon Our Product/

Company?
Answer :
Since I ’d be coming from an altogether new environment, I am bound to possess a new perspective towards everything here including the company, product, customers, environment, strategy etc. This will enable me to constructively question things which anyone else here might not do. This will help in improving the things and making the product & company better.Having worked closely with product development team I understand how the research for product development is carried out and how is customer requirement analysed; I ’d be able to provide a value addition there too.
Question 3. Don't You Think, You Are Overqualified For This Position?
Answer :
You might feel that I possess more degrees than you require for this position. But, I believe that I grow everyday when I talk to my staff, customers and superiors. So, basically the learning process continues through out the life – I don ’t think I am over qualified.
Question 4. Is There Anything That You Do Not Like About Your Last Or Current Job?
Answer :
I was quite enthusiastic while joining my last job. Towards the end, the number of challenges and opportunity to grow further started diminishing. A challenge loving and growth oriented person like me doesn ’t enjoy this.
Question 5. How Long Can You Commit To Work With Us?
Answer :
I like new challenges and a chance to grow. As long I keeping getting these, I don ’t think I ’ll need to switch ove
Question 6. You Seem To Be Drawing A Good Salary. Will You Be Ok In Taking A Salary Cut?
Answer :
I believe that at one point of time in career salary becomes secondary and self actualisation become more important. While taking up any new job, it will be my priority to ensure that the work culture, chances to contribute and grow are sufficient along with the money I am paid. I also believe that any good company who cares about its employees ensures that they are paid well.
Question 7. What Do You Do To Improve Your Knowledge?

Answer :
The field of IT is very revolutionary. It is extremely important to keep yourself abreast with the new technological developments and this needs you to take some time out of your work schedule so that you can keep sharpening your saw.
Question 8. Can You Perform Under Pressure?
Answer :
Most of the times, the job of software development is that of working under pressure. Sometimes, it will be the pressure of delivering on time while it can be that of a bug that has sprung all of a sudden in your code.So, expect pressure in everything you do. It is important to maintain your performance and develop strategies to deliver under pressure. You can then go ahead an talk about your way of dealing with pressure and performing under it.
Question 9. What Irritates You About Co-workers?
Answer :
The purpose of this question is to see how well you can fit into a team. Basically, you should not have a problem with a person, although you can have a problem with the style of working.
Question 10. Is There Any Particular Kind Of Person You Can Not Work With?
Answer :
For the reason given in the above questions, the answer to this question should be a “No ”.
Question 11. What Motivates You At Work?
Answer :
you can mention things like – new challenges, good environment which all employers think that they offer.
Question 12. Will You Be Happy To Work In Night Shifts Or Over The Weekends?
Answer :
You need to answer this question taking into consideration what is suitable for you. Say that you can work in the night shifts, only if you can really do it.
Question 13. Have You Ever Committed A Mistake At Work?
Answer :
To err is human. So, it is perfectly OK if you committed a mistake at work but before answering the question analyse the magnitude of mistake you did and the effect it had on the company.
Question 14. What Are The Most Important Things For You As

A Manager?
Answer :
The two things which should be most important for a manager to succeed in his role are:
His team should be happy and keep performing
The project he is working on with his team is successfully finished with minimum problems.
Question 15. What Kind Of A Salary Are You Looking For?
Answer :
Try to put the ball back in interviewer ’s court by asking him about the salary they offer for a position like this. Most of the big companies will have a fixed remuneration for each level.
However, if this is negotiable, you will have more negotiation power if you have some work experience. So, know your lower limit (amount below which you can ’t go) and also know the maximum salary in the industry for the position then put forward a figure which is not very exact. It is better to mention a range. For e.g. if you are expecting something around 55 K, say that you expect something in mid fifties. Don ’t keep the range to broad otherwise you will be offered something towards the lower end.
If you are a fresher, most of the times you will have to accept the company ’s offering for the position. However, if you find it too less, you can definitely discuss that during the interview.
Question 16. Can You Work Independently?
Answer :
Yes, I can work independently without supervision or support from a team
Do not emphasize on working independently as that will be seen as an inability to work with others.
Question 17. What Is More Important To You Money Or Success?
Answer :
This is tricky question, as money and success both are important and you cannot outweigh the importance of one over the other. Personally you might prefer money over success or success over money, but it is better to be neutral when answering this question in an interview:
You can say, that money and success both are important for you, but if you have to choose you would choose success. The reason being, if one is successful money often follows

and you need not focus on money over succes
Question 18. What Are Your Greatest Strengths?
Answer :
Number of answers are good to give, but in positive manner. Some good instances are:
About your problem solving skills, ability to work hard, professional expertise, leadership skills, positive attitude etc.
Question 19. Do You Have Any Blind Spots?
Answer :
Disguise your strength as a weakness. Beware this is an eliminator question, designed to shorten the candidate list.
Question 20. What Makes You Angry?
Answer :
Avoid giving any specific reason for you getting angry. Say that it's a general tendency of a human being and you are no exception. You do get angry but you don't go out of control.
Question 21. May I Contact Your Present Employer For A Reference?
Answer :
You can say that you have no objection for this but express your concern that it would be better if you could do it after arriving some sort of agreement.
You can even be honest in stating that your present employer is not aware of you hunting and you would appreciate to have any contact with them after having arrived some sort of agreement.
Question 22. What Was The Toughest Decision You Ever Had To Make?
Answer :
Be prepared with a good example, explaining why the decision was difficult, the process you followed in reaching it and the courageous or effective way you carried it out.

****
****

Java
****

1) Difference between String, StringBuffer and StringBuilder in Java? (detailed answer)
String is immutable while both StringBuffer and StringBuilder is mutable, which means any change e.g. converting String to upper case or trimming white space will produce another instance rather than changing the same instance. On later two, StringBuffer is synchronized while StringBuilder is not, in fact its a ditto replacement of StringBuffer added in Java 1.5.

2) Difference between extends Thread vs implements Runnable in Java? (detailed answer)
Difference comes from the fact that you can only extend one class in Java, which means if you extend Thread class you lose your opportunity to extend another class, on the other hand if you implement Runnable, you can still extend another class.

3) Difference between Runnable and Callable interface in Java? (detailed answer)
Runnable was the only way to implement a task in Java which can be executed in parallel before JDK 1.5 adds Callable. Just like Runnable, Callable also defines a single call() method but unlike run() it can return values and throw exceptions.

4) Difference between ArrayList and LinkedList in Java? (detailed answer)
In short, ArrayList is backed by array in Java, while LinkedList is just collection of nodes, similar to linked list data structure. ArrayList also provides random search if you know the index, while LinkedList only allows sequential search. On other hand, adding and removing element from middle is efficient in LinkedList as compared to ArrayList because it only require to modify links and no other element is rearranged.

5) What is difference between wait and notify in Java? (detailed answer)
Both wait and notify methods are used for inter thread communication, where wait is used to pause the thread on a condition and notify is used to send notification to waiting threads. Both must be called from synchronized

context e.g. synchronized method or block.

6) Difference between HashMap and Hashtable in Java? (detailed answer)
Though both HashMap and Hashtable are based upon hash table data structure, there are subtle difference between them. HashMap is non synchronized while Hashtable is synchronized and because of that HashMap is faster than Hashtable, as there is no cost of synchronization associated with it. One more minor difference is that HashMap allows a null key but Hashtable doesn't.

7) Difference between TreeSet and TreeMap in Java? (detailed answer)
Though both are sorted collection, TreeSet is essentially a Set data structure which doesn't allow duplicate and TreeMap is an implementation of Map interface. In reality, TreeSet is implemented via a TreeMap, much like how HashSet is implemented using HashMap.

8) Write a Java program to print Fibonacci series? (solution)
Fibonacci series is a series of number on which a number is equal to sum of previous two numbers i.e. f(n) = f(n-1) + f(n-2). This program is used to teach recursion to students but you can also solve it without recursion. Check out the solution for both iterative and recursive solution of this problem. In telephonic interview, this question is not that common but sometime interviewer also wants to check your problem solving skill using such questions.

9) Write a Java program to check if a number is Prime or not? (solution)
A number is said prime if it is not divisible by any other number except itself. 1 is not considered prime, so your check must start with 2. Simplest solution of this to check every number until the number itself to see if its divisible or not. When Interviewer will ask you to improve, you can say that checking until square root of the number. If you can further improve the algorithm, you will more impress your interviewer. check out the solution for how to do this in Java

10) How to Swap two numbers without using temp variable?

(solution)
This question is ages old. I have first seen this question way back in 2005 but I am sure its even older than that. Good thing about this problem is that except XOR trick all solution has some flaws, which is used to test whether candidate really knows his stuff or not. Check out the solution for all three possible solution and drawback of each.

11) How to check if linked list contains loop in Java? (solution)
This is another problem solving question which is very popular in telephonic and screening round. This is a great question to test problem solving skill of candidate, especially if he has not seen this question before. It also has a nice little followup to find the starting of the loop. See the solution for a Java program which finds loop in singly linked.

12) Write Java program to reverse String without using API? (solution)
One more question to test problem solving skill of candidate. You wouldn't expect these kind of question in telephonic round of Java interview but these questions have now become norms. All interviewer is looking it for logic, you don't need to write the code but you should be able to think of solution.

13) Difference between Serializable and Externalizable in Java? (detailed answer)
Serializable is a marker interface with no methods defined it but Externalizable interface has two methods defined on it e.g. readExternal() and writeExternal() which allows you to control the serialization process. Serializable uses default serialization process which can be very slow for some application.

14) Difference between transient and volatile in Java? (detailed answer)
transient keyword is used in Serialization while volatile is used in multi-threading. If you want to exclude a variable from serialization process then mark that variable transient. Similar to static variable, transient variables are not serialized. On the other hand, volatile variables

are signal to compiler that multiple threads are interested on this variable and it shouldn't reorder its access. volatile variable also follows happens-before relationship, which means any write happens before any read in volatile variable. You can also make non atomic access of double and long variable atomic using volatile.

15) Difference between abstract class and interface? (detailed answer)
From Java 8 onwards difference between abstract class and interface in Java has minimized, now even interface can have implementation in terms of default and static method. BTW, in Java you can still extend just one class but can extend multiple inheritance. Abstract class is used to provide default implementation with just something left to customize, while interface is used heavily in API to define contract of a class.

16) Difference between Association, Composition and Aggregation? (detailed answer)
Between association, composition and aggregation, composition is strongest. If part can exists without whole than relationship between two class is known as aggregation but if part cannot exists without whole than relationship between two class is known as composition. Between Inheritance and composition, later provides more flexible design.

17) What is difference between FileInputStream and FileReader in Java? (detailed answer)
Main difference between FileInputStream and FileReader is that former is used to read binary data while later is used to read text data, which means later also consider character encoding while converting bytes to text in Java.

18) How do you convert bytes to character in Java? (detailed answer)
Bytes are converted to character or text data using character encoding. When you read binary data from a file or network endpoint, you provide a character encoding to convert those bytes to equivalent character. Incorrect choice of character encoding may alter meaning of message by interpreting it differently.

19) Can we have return statement in finally clause? What will happen? (detailed answer)
Yes, you can use return statement in finally block, but it will not prevent finally block from being executed. BTW, if you also used return statement in try block then return value from finally block with override whatever is returned from try block.

20) Can you override static method in Java? (detailed answer)
No, you cannot override static method in Java because they are resolved at compile time rather than runtime. Though you can declare and define static method of same name and signature in child class, this will hide the static method from parent class, that's why it is also known as method hiding in Java.

21) Difference between private, public, package and protected in Java? (detailed answer)
All four are access modifier in Java but only private, public and protected are modifier keyword. There is no keyword for package access, its default in Java. Which means if you don't specify any access modifier than by default that will be accessible inside the same package. Private variables are only accessible in the class they are declared, protected are accessible inside all classes in same package but only on sub class outside package and public variables e.g. method, class or variables are accessible anywhere. This is highest level of access modifier and provide lowest form of encapsulation.

22) 5 Coding best practices you learned in Java? (detailed answer)
If you are developing on a programming language for couple of years, you sure knows lots of best practices, by asking couple of them, Interviewer just checking that you know your trade well. Here are my 5 Java best practices :
- Always name your thread, this will help immensely in debugging.
- Use StringBuilder for string concatenation
- Always specify size of Collection, this will save lot of time spent on resizing
- Always declare variable private and final unless you have good reason.

- Always code on interfaces instead of implementation
- Provide dependency to method instead they get it by themselves, this will make your code unit testable.

23) Write a Program to find maximum and minimum number in array? (solution)
This is another coding question test problem solving ability of candidate. Be ready for couple of follow up as well depending upon how you answer this question. Simplest way which comes in mind is to sort the array and then pick the top and bottom element. For a better answer see the solution.

24) Write a program to reverse Array in place? (solution)
Another problem solving question for Java programmers. Key point here is that you need to reverse the array in place, which means you cannot use additional buffer, one or two variable will be fine. Note you cannot use any library code, you need to create your own logic.

25) Write a program to reverse a number in Java? (solution)
This is an interesting program for a very junior programmer, right from the college but can sometime puzzle developer with couple of years of experience as well. Most of these developer does very little coding so they found these kind of questions challenging. Here the trick is to get the last digit of the number by using modulus operator (%) and reducing number in each go by using division operator (/). See the solution for how to do that in Java.

26) Write a Program to calculate factorial in Java? (solution)
Another beginners coding problem, good for telephonic interview because you can differentiate a guy who can write program to the guy who can't. It's also good to see if developer is familiar with both recursive and iterative algorithm and pros and cons of each. You can also ask lots of follow up e.g. how to improve performance of algorithm? Since factorial of a number is equal to number multiplied by factorial of previous number, you can cache those value instead of recalculating them, this will impress your interviewer a bit. See the solution for full code example.

27) What is difference between calling start() and run()

method of Thread? (detailed answer)
You might have heard this question before, if calling start() method calls the run() method eventually then why not just call the run() method? Well the difference is, start method also starts a new thread. If you call the run method directly then it will run on same thread not on different thread, which is what original intention would be.

28) Write a Program to solve Producer Consumer problem in Java? (solution)
A very good question to check if candidate can write inter thread communication code or not. If a guy can write producer consumer solution by hand and point out critical section and how to protect, how to communicate with thread then he is good enough to write and maintain your concurrent Java program. This is the very minimum requirement for a Java developer and that's why I love this question, it also has several solution e.g. by using concurrent collections like blocking queue, by using wait and notify and by using other synchronizers of Java 5 e.g. semaphores.

29) How to find middle element of linked list in one pass? (solution)
Another simple problem solving question for warm up. In a singly linked list you can only traverse in one direction and if you don't know the size then you cannot write a loop to exactly find out middle element, that is the crux of the problem. One solution is by using two pointers, fast and slow. Slower pointer moves 1 step when faster pointer move to 2 steps, causing slow to point to middle when fast is pointing to end of the list i.e. null. Check out solution for Java code sample.

30) What is equlas() and hashCode() contract in Java? Where does it used? (detailed answer)
One of the must ask question in Java telephonic interview. If a guy doesn't know about equals() and hashCode() then he is probably not worth pursuing further because its the core of the Java fundamentals.

The key point of contract is that if two objects are equal by equals() method then they must have same hashcode, but

unequal object can also have same hashcode, which is the cause of collision on hash table based collection e.g HashMap. When you override equals() you must remember to override hashCode() method to keep the contract valid.

31) Why wait and notify methods are declared in Object class? (detailed answer)VVVVIMP
This question is more to find out how much experience you really have and what is your thinking towards Java API and its design decision. Similar question is why String is immutable in Java? Well, true answer can only be given by Java designers but you can reason something. For example, wait and notify methods are associated with locks which is owned by object not thread, and that's why it make sense to keep those method on java.lang.Object class. See the detailed answer for more discussion and reasoning.

32) How does HashSet works in Java? (detailed answer)

HashSet is internally implemented using HashMap in Java and this is what your interviewer wants to hear. He could then quiz you with some common sense based question e.g. how can you use HashMap because its needs two object key and value? what is the value in case of HashSet? Well, in case of HashSet a dummy object is used as value and key objects are the actual element on Set point of view. Since HashMap doesn't allow duplicate key it also follows contract of set data structure to not allow duplicates. See detailed answer for more analysis and explanation.

33) What is difference between synchronize and concurrent Collection in Java? (detailed answer)
There was time, before Java 1.5 when you only have synchronized collection if you need them in a multi-threaded Java program. Those classes were plagued with several issue most importantly performance because they lock the whole collection or map whenever a thread reads or writes. To address those issue, Java released couple of Concurrent collection classes e.g. ConcurrentHashMap, CopyOnWriteArrayList and BlockingQueue to provide more scalability and performance.

34) What is difference between Iterator and Enumeration in Java? (detailed answer)

Main difference is that Iterator was introduced in place of Enumeration. It also allows you to remove elements from collection while traversing which was not possible with Enumeration. The methods of Iterator e.g. hasNext() and next() are also more concise then corresponding methods in Enumeration e.g. hasMoreElements(). You should always use Iterator in your Java code as Enumeration may get deprecated and removed in future Java release.

35) What is difference between Overloading and Overriding in Java? (detailed answer)
Another frequently asked question from telephonic round of Java interviews. Though both overloading and overriding are related with methods of same names but they have different characteristics e.g.overloaded methods must have different method signature than original method but overridden method must have same signature. Also, overloaded methods are resolved at compiled time while overridden methods are resolved at runtime. See the detailed answer for more analysis and differences.

36) Difference between static and dynamic binding in Java? (detailed answer)
This is usually asked as follow-up of previous question, static binding is related to overloaded method and dynamic binding is related to overridden method. Method like private, final and static are resolved using static binding at compile time but virtual methods which can be overridden are resolved using dynamic binding at runtime.

37) Difference between Comparator and Comparable in Java? (detailed answer)
This is one more basic concept, I expect every Java candidate to know. You will deal with them on every Java project. Several core classes in Java e.g. String, Integer implements Comparable to define their natural sorting order and if you define a value class or a domain object then you should also implement Comparable and define natural ordering of your object. Main difference between these two is that, you could create multiple Comparator to define multiple sorting order based upon different attribute of object. Also, In order to implement Comparable you must have access of the class or code, but you can use Comparator without having source code of class, all you

need is the JAR file of particular object. That's why Comparator is very powerful to implement custom sorting order and from Java 8 you can do it even more elegantly, as seen here.

38) How do you sort ArrayList in descending order? (solution)
You can use Collections.sort() method with reverse Comparator, which can sort elements in the reverse order of their natural order e.g.

```
List<String> listOfString = Arrays.asList("London", "Tokyo", "NewYork");
Collections.sort(listOfString, Collections.reverseOrder());
System.out.println(listOfString); //[Tokyo, NewYork, London]
```

39) What is difference between PATH and CLASSPATH in Java? (detailed answer)
PATH is an environment variable which points to Java binary which is used to run Java programs. CLASSPATH is another environment variable which points to Java class files or JAR files. If a class is not found in CLASSPATH then Java throws ClassNotFoundException.

40) What is difference between Checked and Unchecked Exception in Java? (detailed answer)
Checked exception ensures that handling of exception is provided and its verified by compiler also, while for throwing unchecked exception no special provision is needed e.g. throws clause. A method can throw unchecked exception without any throw cla.

****

JAVA 8 INTERVIEW QUESTIONS & ANSWERS

Question 1. What Are The New Features Introduced In Java 8?

Answer :

There are dozens of features added to Java 8, the most significant ones are mentioned below –

Lambda expression - Adds functional processing capability to Java.
Method references - Referencing functions by their names instead of invoking them directly. Using functions as parameter.
Default method - Interface to have default method implementation.
New tools - New compiler tools and utilities are added like 'jdeps' to figure out dependencies.
Stream API - New stream API to facilitate pipeline processing.
Date Time API - Improved date time API.
Optional - Emphasis on best practices to handle null values properly.
Nashorn, JavaScript Engine - A Java-based engine to execute JavaScript code.
Along with these new featuers, lots of feature enhancements are done under-the-hood, at both compiler and JVM level.

Question 2. How Will You Sort A List Of String Using Java 8 Lambda Expression?

Answer :

Following code sorts a list of string using Java 8 lambda expression:

```
//sort using java 8
private void sortUsingJava8(List<String> names){
Collections.sort(names, (s1, s2) -> s1.compareTo(s2));
}
```

Question 3. What Are The Characteristics Of A Java 8 Lambda Expression?

Answer :

A lambda expression is characterized by the following syntax - parameter -> expression body

Following are the important characteristics of a lambda expression -

Optional type declaration – No need to declare the type of a parameter. The compiler can inference the same from the value of the parameter.
Optional parenthesis around parameter – No need to declare a single parameter in parenthesis. For multiple parameters, parentheses are required.
Optional curly braces – No need to use curly braces in expression body if the body contains a single statement.
Optional return keyword – The compiler automatically returns the value if the body has a single expression to return the value. Curly braces are required to indicate that expression returns a value.
Question 4. Why Lambda Expression Is To Be Used?

Answer :

Lambda expressions are used primarily to define inline implementation of a functional interface, i.e., an interface with a single method only. In the above example, we've used various types of lambda expressions to define the operation method of MathOperation interface. Then we have defined the implementation of sayMessage of GreetingService.

Lambda expression eliminates the need of anonymous class and gives a very simple yet powerful functional programming capability to Java.

Question 5. What Kind Of Variable You Can Access In An Lambda Expression?

Answer :

Using lambda expression, you can refer to final variable or effectively final variable (which is assigned only once). Lambda expression throws a compilation error, if a variable is assigned a value the second time.

Question 6. What Are Method References?

Answer :

Method references help to point to methods by their names.

A method reference is described using :: (double colon) symbol. A method reference can be used to point the following types of methods -

Static methods
Instance methods
Constructors using new operator (TreeSet::new)
Question 7. Explain The System.out::println Expression?

Answer :

System.out::println method is a static method reference to println method of out object of System class.

Question 8. What Are Functional Interfaces?

Answer :

Functional interfaces have a single functionality to exhibit. For example, a Comparable interface with a single method 'compareTo' is used for comparison purpose. Java 8 has defined a lot of functional interfaces to be used extensively in lambda expressions.

Question 9. What Is The Purpose Of Biconsumer<t,u> Functional Interface?

Answer :

It represents an operation that accepts two input arguments, and returns no result.

Question 10. What Is The Purpose Of Bifunction<t,u,r> Functional Interface?

Answer :

It represents a function that accepts two arguments and produces a result.

Question 11. What Is The Purpose Of Binaryoperator<t> Functional Interface?

Answer :

It represents an operation upon two operands of the same type, producing a result of the same type as the operands.

Question 12. What Is The Purpose Of Bipredicate<t,u> Functional Interface?

Answer :

It represents a predicate (Boolean-valued function) of two arguments.

Question 13. What Is The Purpose Of Booleansupplier Functional Interface?

Answer :

It represents a supplier of Boolean-valued results.

Question 14. What Is The Purpose Of Consumer<t> Functional Interface?

Answer :

It represents an operation that accepts a single input argument and returns no result.

Question 15. What Is The Purpose Of Doublebinaryoperator Functional Interface?

Answer :

It represents an operation upon two double-valued operands and producing a double-valued result.

Question 16. What Is The Purpose Of Doubleconsumer Functional Interface?

Answer :

It represents an operation that accepts a single double-valued argument and returns no result.

Question 17. What Is The Purpose Of Doublefunction<r> Functional Interface?

Answer :

It represents a function that accepts a double-valued argument and produces a result.

Question 18. What Is The Purpose Of Doublepredicate Functional Interface?

Answer :

It represents a predicate (Boolean-valued function) of one double-valued argument.

Question 19. What Is The Purpose Of Doublesupplier Functional Interface?

Answer :

It represents a supplier of double-valued results.

Question 20. What Is The Purpose Of Doubletointfunction Functional Interface?

Answer :

It represents a function that accepts a double-valued argument and produces an int-valued result.

Question 21. What Is The Purpose Of Doubletolongfunction Functional Interface?

Answer :

It represents a function that accepts a double-valued argument and produces a long-valued result.

Question 22. What Is The Purpose Of Doubleunaryoperator Functional Interface?

Answer :

It represents an operation on a single double-valued operand that produces a double-valued result.

Question 23. What Is The Purpose Of Function<t,r> Functional Interface?

Answer :

It represents a function that accepts one argument and produces a result.

Question 24. What Is The Purpose Of Intbinaryoperator Functional Interface?

Answer :

It represents an operation upon two int-valued operands and produces an int-valued result.

Question 25. What Is The Purpose Of Intconsumer Functional Interface?

Answer :

It represents an operation that accepts a single int-valued argument and returns no result.

Question 26. What Is The Purpose Of Intfunction<r> Functional Interface?

Answer :

It represents a function that accepts an int-valued argument and produces a result.

Question 27. What Is The Purpose Of Intpredicate Functional Interface?

Answer :

It represents a predicate (Boolean-valued function) of one int-valued argument.

Question 28. What Is The Purpose Of Intsupplier Functional Interface?

Answer :

It represents a supplier of int-valued results.

Question 29. What Is The Purpose Of Inttodoublefunction Functional Interface?

Answer :

It represents a function that accepts an int-valued argument and produces a double-valued result.

Question 30. What Is The Purpose Of Inttolongfunction Functional Interface?

Answer :

It represents a function that accepts an int-valued argument and produces a long-valued result.

Question 31. What Is The Purpose Of Intunaryoperator Functional Interface?

Answer :

It represents an operation on a single int-valued operand that produces an int-valued result.

Question 32. What Is The Purpose Of Longbinaryoperator Functional Interface?

Answer :

It represents an operation upon two long-valued operands and produces a long-valued result.

Question 33. What Is The Purpose Of Longconsumer Functional Interface?

Answer :

It represents an operation that accepts a single long-valued argument and returns no result.

Question 34. What Is The Purpose Of Longfunction<r> Functional Interface?

Answer :

It represents a function that accepts a long-valued argument and produces a result.

Question 35. What Is The Purpose Of Longpredicate Functional Interface?

Answer :

It represents a predicate (Boolean-valued function) of one long-valued argument.

Question 36. What Is The Purpose Of Longsupplier Functional Interface?

Answer :

It represents a supplier of long-valued results.

Question 37. What Is The Purpose Of Longtodoublefunction Functional Interface?

Answer :

It represents a function that accepts a long-valued argument and produces a double-valued result.

Question 38. What Is The Purpose Of Longtointfunction Functional Interface?

Answer :

It represents a function that accepts a long-valued argument and produces an int-valued result.

Question 39. What Is The Purpose Of Longunaryoperator Functional Interface?

Answer :

It represents an operation on a single long-valued operand that produces a long-valued result.

Question 40. What Is The Purpose Of Objdoubleconsumer<t> Functional Interface?

Answer :

It represents an operation that accepts an object-valued and a double-valued argument, and returns no result.

Question 41. What Is The Purpose Of Objintconsumer<t> Functional Interface?

Answer :

It represents an operation that accepts an object-valued and an int-valued argument, and returns no result.

Question 42. What Is The Purpose Of Objlongconsumer<t> Functional Interface?

Answer :

It represents an operation that accepts an object-valued and a long-valued argument, and returns no result.

Question 43. What Is The Purpose Of Predicate<t> Functional Interface?

Answer :

It represents a predicate (Boolean-valued function) of one argument.

Question 44. What Is The Purpose Of Supplier<t> Functional Interface?

Answer :

It represents a supplier of results.

Question 45. What Is The Purpose Of Todoublebifunction<t,u> Functional Interface?

Answer :

It represents a function that accepts two arguments and produces a double-valued result.

Question 46. What Is The Purpose Of Todoublefunction<t> Functional Interface?

Answer :

It represents a function that produces a double-valued result.

Question 47. What Is The Purpose Of Tointbifunction<t,u> Functional Interface?

Answer :

It represents a function that accepts two arguments and produces an int-valued result.

Question 48. What Is The Purpose Of Tointfunction<t> Functional Interface?

Answer :

It represents a function that produces an int-valued result.

Question 49. What Is The Purpose Of Tolongbifunction<t,u> Functional Interface?

Answer :

It represents a function that accepts two arguments and produces a long-valued result.

Question 50. What Is The Purpose Of Tolongfunction<t> Functional Interface?

Answer :

It represents a function that produces a long-valued result.

Question 51. What Is The Purpose Of Unaryoperator<t> Functional Interface?

Answer :

It represents an operation on a single operand that produces a result of the same type as its operand.

Question 52. What Are Default Methods?

Answer :

With java 8, an interface can have default implementation of a function in interfaces.

Question 53. What Are Static Default Methods?

Answer :

An interface can also have static helper methods from Java 8 onwards.

```
public interface vehicle {
default void print(){
System.out.println("I am a vehicle!");
}
static void blowHorn(){
System.out.println("Blowing horn!!!");
}
}
```

Question 54. How Will You Call A Default Method Of An Interface In A Class?

Answer :

Using super keyword along with interface name.

```
interface Vehicle {
```

```
default void print(){
System.out.println("I am a vehicle!");
}
}
class Car implements Vehicle {
public void print(){
Vehicle.super.print();
}
}
```

Question 55. How Will You Call A Static Method Of An Interface In A Class?

Answer :

Using name of the interface.

```
interface Vehicle {
static void blowHorn(){
System.out.println("Blowing horn!!!");
}
}
class Car implements Vehicle {
public void print(){
Vehicle.blowHorn();
}
}
```

Question 56. What Is Streams In Java 8?

Answer :

Stream represents a sequence of objects from a source, which supports aggregate operations.

Question 57. What Is Stream Pipelining In Java 8?

Answer :

Most of the stream operations return stream itself so that their result can be pipelined. These operations are called intermediate operations and their function is to take input, process them, and return output to the target. collect() method is a terminal operation which is normally

present at the end of the pipelining operation to mark the end of the stream.

Question 58. What Is The Difference Between Collections And Stream In Java8 ?

Answer :

Stream operations do the iterations internally over the source elements provided, in contrast to Collections where explicit iteration is required.

Question 59. What Is The Purpose Of Foreach Method Of Stream In Java 8?

Answer :

Stream has provided a new method 'forEach' to iterate each element of the stream.

Question 60. How Will You Print 10 Random Numbers Using Foreach Of Java 8?

Answer :

The following code segment shows how to print 10 random numbers using forEach.

```
Random random = new Random();
random.ints().limit(10).forEach(System.out::println);
```

Question 61. What Is The Purpose Of Map Method Of Stream In Java 8?

Answer :

The 'map' method is used to map each element to its corresponding result.

Question 62. How Will You Print Unique Squares Of Numbers In Java 8?

Answer :

The following code segment prints unique squares of numbers using map.

```
List<Integer> numbers = Arrays.asList(3, 2, 2, 3, 7, 3, 5);

//get list of unique squares

List<Integer> squaresList = numbers.stream().map( i ->
i*i).distinct().collect(Collectors.toList());
```

Question 63. What Is The Purpose Of Filter Method Of Stream In Java 8?

Answer :

The 'filter' method is used to eliminate elements based on a criteria.

Question 64. How Will You Print Count Of Empty Strings In Java 8?

Answer :

The following code segment prints a count of empty strings using filter.

```
List<String>strings = Arrays.asList("abc", "", "bc", "efg",
"abcd","", "jkl");
//get count of empty string
int count = strings.stream().filter(string ->
string.isEmpty()).count();
```

Question 65. What Is The Purpose Of Limit Method Of Stream In Java 8?

Answer :

The 'limit' method is used to reduce the size of the stream.

Question 66. How Will You Print 10 Random Numbers In Java 8?

Answer :

The following code segment shows how to print 10 random numbers.

```
Random random = new Random();
random.ints().limit(10).forEach(System.out::println);
```

Question 67. What Is The Purpose Of Sorted Method Of Stream In Java 8?

Answer :

The 'sorted' method is used to sort the stream.

Question 68. How Will You Print 10 Random Numbers In A Sorted Order In Java 8?

Answer :

The following code segment shows how to print 10 random numbers in a sorted order.

```
Random random = new Random();
random.ints().limit(10).sorted().forEach(System.out::println);
```

Question 69. What Is Parallel Processing In Java 8?

Answer :

parallelStream is the alternative of stream for parallel processing. Take a look at the following code segment that prints a count of empty strings using parallelStream.

```
List<String> strings = Arrays.asList("abc", "", "bc",
"efg", "abcd","", "jkl");
//get count of empty string
int count = strings.parallelStream().filter(string ->
string.isEmpty()).count();
//It is very easy to switch between sequential and parallel
streams.
```

Question 70. What Are Collectors In Java 8?

Answer :

Collectors are used to combine the result of processing on the elements of a stream. Collectors can be used to return a list or a string.

```
List<String>strings = Arrays.asList("abc", "", "bc", "efg", "abcd","", "jkl");
List<String> filtered = strings.stream().filter(string -> !string.isEmpty()).collect(Collectors.toList());
System.out.println("Filtered List: " + filtered);
String mergedString = strings.stream().filter(string -> !string.isEmpty()).collect(Collectors.joining(", "));
System.out.println("Merged String: " + mergedString);
```

Question 71. What Are Statistics Collectors In Java 8?

Answer :

With Java 8, statistics collectors are introduced to calculate all statistics when stream processing is being done.

Question 72. How Will You Get The Highest Number Present In A List Using Java 8?

Answer :

Following code will print the highest number present in a list.

```
List numbers = Arrays.asList(3, 2, 2, 3, 7, 3, 5);
IntSummaryStatistics stats = integers.stream().mapToInt((x) -> x).summaryStatistics();
System.out.println("Highest number in List : " + stats.getMax());
```

Question 73. How Will You Get The Lowest Number Present In A List Using Java 8?

Answer :

Following code will print the highest number present in a list.

```
List numbers = Arrays.asList(3, 2, 2, 3, 7, 3, 5);
IntSummaryStatistics stats = integers.stream().mapToInt((x)
-> x).summaryStatistics();
System.out.println("Lowest number in List : " +
stats.getMin());
```

Question 74. How Will You Get The Sum Of All Numbers Present In A List Using Java 8?

Answer :

Following code will print the sum of all numbers present in a list.

```
List numbers = Arrays.asList(3, 2, 2, 3, 7, 3, 5);
IntSummaryStatistics stats = integers.stream().mapToInt((x)
-> x).summaryStatistics();
System.out.println("Sum of all numbers : " +
stats.getSum());
```

Question 75. How Will You Get The Average Of All Numbers Present In A List Using Java 8?

Answer :

Following code will print the average of all numbers present in a list.

```
List numbers = Arrays.asList(3, 2, 2, 3, 7, 3, 5);
IntSummaryStatistics stats = integers.stream().mapToInt((x)
-> x).summaryStatistics();
System.out.println("Average of all numbers : " +
stats.getAverage());
```

Question 76. What Is Optional In Java8?

Answer :

Optional is a container object which is used to contain not-null objects. Optional object is used to represent null with absent value. This class has various utility methods to facilitate code to handle values as 'available' or 'not available' instead of checking null values. It is

introduced in Java 8 and is similar to what Optional is in Guava.

Question 77. What Is Nashorn In Java8?

Answer :

With Java 8, Nashorn, a much improved javascript engine is introduced, to replace the existing Rhino. Nashorn provides 2 to 10 times better performance, as it directly compiles the code in memory and passes the bytecode to JVM. Nashorn uses invokedynamics feature, introduced in Java 7 to improve performance.

Question 78. What Is Jjs In Java8?

Answer :

For Nashorn engine, JAVA 8 introduces a new command line tool, jjs, to execute javascript codes at console.

Question 79. Can You Execute Javascript Code From Java 8 Code Base?

Answer :

Yes! Using ScriptEngineManager, JavaScript code can be called and interpreted in Java.

Question 80. What Is Local Datetime Api In Java8?

Answer :

Local - Simplified date-time API with no complexity of timezone handling.

Question 81. What Is Zoned Datetime Api In Java8?

Answer :

Zoned - Specialized date-time API to deal with various timezones.

Question 82. What Is Chromounits In Java8?

Answer :

java.time.temporal.ChronoUnit enum is added in Java 8 to replace the integer values used in old API to represent day, month, etc.

Question 83. How Will You Get The Current Date Using Local Datetime Api Of Java8?

Answer :

Following code gets the current date using local datetime api –

```
//Get the current date
LocalDate today = LocalDate.now();
System.out.println("Current date: " + today);
```

Question 84. How Will You Add 1 Week To Current Date Using Local Datetime Api Of Java8?

Answer :

Following code adds 1 week to current date using local datetime api –

```
//add 1 week to the current date
LocalDate today = LocalDate.now();
LocalDate nextWeek = today.plus(1, ChronoUnit.WEEKS);
System.out.println("Next week: " + nextWeek);
```

Question 85. How Will You Add 1 Month To Current Date Using Local Datetime Api Of Java8?

Answer :

Following code adds 1 month to current date using local datetime api:

```
//add 1 month to the current date
LocalDate today = LocalDate.now();
LocalDate nextMonth = today.plus(1, ChronoUnit.MONTHS);
System.out.println("Next month: " + nextMonth);
```

Question 86. How Will You Add 1 Year To Current Date Using Local Datetime Api Of Java8?

Answer :

Following code adds 1 year to current date using local datetime api –

```
//add 1 year to the current date
LocalDate today = LocalDate.now();
LocalDate nextYear = today.plus(1, ChronoUnit.YEARS);
System.out.println("Next year: " + nextYear);
```

Question 87. How Will You Add 10 Years To Current Date Using Local Datetime Api Of Java8?

Answer :

Following code adds 10 years to current date using local datetime api –

```
//add 10 years to the current date
LocalDate today = LocalDate.now();
LocalDate nextDecade = today.plus(1, ChronoUnit.DECADES);
System.out.println("Date after ten year: " + nextDecade);
```

Question 88. How Will You Get Next Tuesday Using Java8?

Answer :

Following code gets next tuesday using java8 –

```
//get the next tuesday
LocalDate today = LocalDate.now();
LocalDate nextTuesday =
today.with(TemporalAdjusters.next(DayOfWeek.TUESDAY));
System.out.println("Next Tuesday on : " + nextTuesday);
```

Question 89. How Will You Get Second Saturday Of Next Month Using Java8?

Answer :

Following code gets second saturday of next month using java8 –

```
//get the second saturday of next month

LocalDate firstInYear =
LocalDate.of(date1.getYear(),date1.getMonth(), 1);

LocalDate secondSaturday =
firstInYear.with(TemporalAdjusters.nextOrSame(DayOfWeek.SAT
URDAY)).with(TemporalAdjusters.next(DayOfWeek.SATURDAY));

System.out.println("Second Saturday on : " +
secondSaturday);
```

Question 90. How Will You Get The Instant Of Current Date In Terms Of Milliseconds Using Java8?

Answer :

Following code gets the instant of current date in terms of milliseconds –

```
//Get the instant of current date in terms of milliseconds

Instant now = currentDate.toInstant();
```

Question 91. How Will You Get The Instant Of Local Date Time Using Time In Of Milliseconds Using Java8?

Answer :

Following code gets the instant of local date time using time in of milliseconds –

```
Instant now = currentDate.toInstant();
ZoneId currentZone = ZoneId.systemDefault();
LocalDateTime localDateTime = LocalDateTime.ofInstant(now,
currentZone);
System.out.println("Local date: " + localDateTime);
```

Question 92. How Will You Get The Instant Of Zoned Date Time Using Time In Of Milliseconds Using Java8?

Answer :

Following code gets the instant of zoned date time using time in of milliseconds −

```
Instant now = currentDate.toInstant();
ZoneId currentZone = ZoneId.systemDefault();
ZonedDateTime zonedDateTime = ZonedDateTime.ofInstant(now, currentZone);
System.out.println("Zoned date: " + zonedDateTime);
```

Question 93. Which Class Implements A Decoder For Decoding Byte Data Using The Base64 Encoding Scheme In Java8?

Answer :

static class Base64.Decoder − This class implements a decoder for decoding byte data using the Base64 encoding scheme as specified in RFC 4648 and RFC 2045.

Question 94. Which Class Implements An Encoder For Encoding Byte Data Using The Base64 Encoding Scheme In Java8?

Answer :

static class Base64.Encoder − This class implements an encoder for encoding byte data using the Base64 encoding scheme as specified in RFC 4648 and RFC 2045.

Question 95. How Will You Create A Base64 Decoder?

Answer :

getDecoder() method of Base64 class returns a Base64.Decoder that decodes using the Basic type base64 encoding scheme.

Question 96. How Will You Create A Base64 Encoder?

Answer :

getEncoder() method of Base64 class returns a Base64.Encoder that encodes using the Basic type base64 encoding scheme.

Question 97. How Will You Create A Base64 Decoder That Decodes Using The Mime Type Base64 Encoding Scheme?

Answer :

getMimeDecoder() method of Base64 class returns a Base64.Decoder that decodes using the MIME type base64 decoding scheme.

Question 98. How Will You Create A Base64 Encoder That Encodes Using The Mime Type Base64 Encoding Scheme?

Answer :

getMimeEncoder() method of Base64 class returns a Base64.Encoder that encodes using the MIME type base64 encoding scheme.

Question 99. How Will You Create A Base64 Decoder That Decodes Using The Url And Filename Safe Type Base64 Encoding Scheme?

Answer :

getUrlDecoder() method of Base64 class returns a Base64.Decoder that decodes using the URL and Filename safe type base64 encoding scheme.

Question 100. How Will You Create A Base64 Encoder That Encodes Using The Url And Filename Safe Type Base64 Encoding Scheme?

Answer :

getUrlEncoder() method of Base64 class returns a Base64.Encoder that encodes using the URL and Filename safe type base64 encoding scheme.

****

JAVA STRING INTERVIEW QUESTIONS & ANSWERS

Question 1. What Is String In Java?

Answer :

In Java String class represents character strings which means; Strings in Java are objects and all strings are instances of the String class. Internally in String class Strings are stored as character array.

Question 2. In How Many Ways String Object Can Be Created?

Answer :

Since strings are objects so strings can of course be created using new operator. String class has more than 10 constructors to create Strings which ranges from taking nothing as parameter to taking char array, StringBuffer, StringBuilder, another String as argument.

Another and more preferred way to create Strings is to assign String literal directly to a String reference as you will do for any primitive type. For every string literal Java will automatically constructs a String object.

As example – String str = “abc”;

Question 3. If String Can Be Created Using String Str = “test” Then String Is A Primitive Data Type.yes/no?

Answer :

No. For every string literal Java automatically constructs a String object.

Question 4. What Is String Pool? Where Is It Created In Memory?

Answer :

When String literals are created they are stored in a String pool and that is a common pool; which means if there are two strings literals having the same content then those string will share the space in the pool.

When String object is created by assigning a string literal, pool will be checked to verify if there is any existing object with the same content if there is then that existing reference is used, no new object is created in that case. If no object is found with the same content then this new literal will be added in the pool.

String pool is stored in the heap.

Question 5. What Is Immutable Object? Is String Object Immutable?

Answer :

An immutable object is an object that would not be able to change its state after creation. Thus immutable object can only be in one state and that state can not be changed after creation of the object.

Yes String object is immutable. Once you create a String object the content of that string cannot be modified.

Question 6. Why Is String Class Immutable?

Answer :

Since Java maintains a string pool where String references are shared thus changing content of any of the String will also affect the other strings sharing the same references that’s one reason why string is immutable.

Question 7. Why Is String Class Final In Java?

Answer :

Since String is immutable, whenever you perform any operation on string which alters its content a new string object is created containing the modified string. Which means all the methods of the String class that modify the content in any way return a new String object with the modified content.

Now, What if you can override the method of the String class so that it modifies and return the original string reference itself? In that case all the other strings having the same data in the string pool will also get affected as the reference is shared for the String literals having the same content.

To avoid these kind of scenarios String class is declared as final and it can’t be overridden.

Question 8. Which Operator Is Overloaded For String?

Answer :

‘+’ operator is overloaded in Java for String. It is used for concatenating two strings.

Question 9. How Many Objects Will Be Created If Two Strings Are Created This Way?

```
String S1 = “test”;
String S2 = “test”;
```

Answer :

Since s1 and s2 are string literals and having the same content object reference will be shared by them in the string pool. Therefore only one object is created.

Question 10. How Many Object Will Be Created If Strings Are Created This Way?

```
String S1 = “test”;
String S2 = New String(“test”);
String S3 = New String(“test”).intern();
```

Answer :

s1 will go to string pool, for s2 new object is created. S3, though created using new will still search in the string pool for any reference having the same content as intern() method is used. So two objects will be created.

Question 11. What Is Intern() Method In String?

Answer :

Using intern() method you can still get string object from the pool (if it exists) even if new operator is used to create a string.

When the intern method is invoked, if the pool already contains a string equal to this String object as determined by the equals(Object) method, then the string from the pool is returned. Otherwise, this String object is added to the pool and a reference to this String object is returned.

Question 12. Is String Thread Safe In Java?

Answer :

Yes string is thread safe in Java as String is immutable.

Question 13. What Is Stringbuffer In Java?

Answer :

StringBuffer class is the companion class of String. StringBuffer is a mutable(modifiable) sequence of characters which is in contrast to String class which is an immutable sequence of characters. Thus in case of StringBuffer length and content of the sequence can be changed through certain method calls.

Since StringBuffer is mutable a new String object is not created every time string is modified, which in turn results in less memory consumptions and not having lots of intermediate String object for garbage collection.

Question 14. What Is Stringbuilder In Java?

Answer :

StringBuilder class (Added in Java 5),just like StringBuffer, is a mutable(modifiable) sequence of characters which is in contrast to String class which is an immutable sequence of characters. Thus in case of

StringBuilder length and content of the sequence can be changed through certain method calls.

Question 15. Differences Among String, Stringbuffer And Stringbuilder In Java?

Answer :

String is immutable where as both StringBuffer and StringBuilder are mutable.

String and StringBuffer are thread safe where as StringBuilder is not thread safe.

Question 16. Is Stringbuffer Class Also Immutable In Java?

Answer :

No, StringBuffer is not immutable.

Question 17. Is Stringbuffer Class Also Final In Java?

Answer :

Yes, StringBuffer class is final in Java.

Question 18. Is Stringbuffer Class Thread Safe?

Answer :

Yes StringBuffer class is thread safe. Methods in StringBuffer class are synchronized.

Question 19. Is Stringbuilder Class Thread Safe?

Answer :

No StringBuilder class is not thread safe. That makes it faster than StringBuffer.

Question 20. Is Stringbuilder Class Also Final In Java?

Answer :

Yes StringBuilder class is final in Java.

Question 21. How To Compare Two Strings In Java?

Answer :

equals() method can be used for comparing two strings in Java. If you want to ignore case then you can use equalsIgnoreCase(String anotherString) method.

There are also compareTo() and compareToIgnoreCase() methods for comparing two strings lexicographically. Returns an integer indicating whether this string is greater than (result is > 0), equal to (result is = 0), or less than (result is < 0) the argument.

You can also use matches() method where you can pass a regular expression for matching strings.

Question 22. What Will Happen If “==” Operator Is Used To Compare Two Strings In Java?

Answer :

“==” operator will compare the references of the strings not the content.

```
String str1 = "abc";
String str4 = new String("abc");
```

Comparing these two strings using “==” operator

```
if(str1 == str4)
```

will return false as the references are different.

Question 23. How To Get Characters And Substrings By Index With In A String?

Answer :

You can get the character at a particular index within a string by invoking the charAt() accessor method.

```
String str = "Example String";
char resChar = str.charAt(3);
```

Will give char ‘m’. If you want to get more than one consecutive character from a string, you can use the substring method. The substring method has two versions –

String substring(int beginIndex, int endIndex) – Returns a new string that is a substring of this string.
String substring(int beginIndex) – Returns a new string that is a substring of this string.

Question 24. How Can You Find Characters Or Substrings Within A String?

Answer :

To find characters or substrings with in a string indexOf() and lastIndexOf() methods can be used.

You can also use contains() method

public boolean contains(CharSequence s) – Returns true if and only if this string contains the specified sequence of char values. Otherwise it returns false.

Question 25. How Can You Split A String In Java?

Answer :

String provides a split method in order to split the string into one or more substring based on the given regular expression.

As example If you have a string where one (or more) spaces are used and you want to split it around those spaces.

```
String str1 = "split example program";
String[] strArray = str1.split("\s+");
```

Question 26. How Can You Join Strings In Java?

Answer :

With Java 8 join() method has been added in the String class which makes it very easy to join the multiple strings.

join method has two overloaded versions –

public static String join(CharSequence delimiter, CharSequence... elements) – Returns a new String composed of copies of the CharSequence elements joined together with a copy of the specified delimiter.
public static String join(CharSequence delimiter, Iterable<? extends CharSequence> elements) – Here elements is an Iterable that will have its elements joined together and delimiter is a sequence of characters that is used to separate each of the elements in the resulting String.
Question 27. Can We Use String In Switch Case Statement?

Answer :

Yes from Java 7 string can be used in switch case statement.

****
****

What is difference between fail-fast and fail-safe Iterator in Java?
==========
==========
CTS
==========
==========
Core-java
****
1 .Tell me the internal flow of Set implementation class with one example
2 . In HashMap if hashing collision occure then how to resolve it.
3 . can we add duplicate in set and map if yes why write one code
4 . Read data from file find the duplicate word and count them and sort them in desending order

5 . where to use Comparable and where to use Comparator did you ever used in ur project
6 . what is bubble sort can you write one programme. .?
7 . can I write try block single means without using try-catch or try-finally
8 . what is Executor framework
9 . how many way we can create thread and which one best approach and why
10 . jdk version u r using in ur project and why (be care on that question coz they indirectly ask u the advantages of version or latest features added in New version)
*Jdbc:-
****
1 . difference between Statement and PreparedStatement
2.they give one db schema and ask me to retrieve data from DB by passing id
*Jsp:-
****
1.Life cycle
2. list down the implicitly object
3.what is the use of c tag library
*Spring :-
****
1.What is RowMapper when we have to use it write sample code not completely just give sm hints with flow
2.what is ResultSet Extractor where exactly we have to use
3.if in my Spring bean configuration file I configure same bean with same id 2 times then what is the problem and how to resolve it (contender)
4 . Spring Mvc Flow as per your Project
5.Spring transaction, why nd how to work on it
6. How u handle Exception in ur Project just give some brief idea on it with annotation
*Webservices :-
****
1 . WSDL ,what are the elements and just explain the role of each section verbally
2.what is Rest,
3 . difference between Soap and Rest
4.Write one Resource method using Http method Post
5.which Response u provide to presentation layer and how to bind Json Response
6.Difference between @QuerryParm and @PathPatm which one best and where to use. ..

==========
==========
DELOITE
==========
==========
1.tell me when to use interface and when to use abstract?
2.some questions from overloading and overring?
3.what is week refference?
4.what is ConcurrantModoficationException?
5.what is thow,thows,throwable?
6. what is static,final and how to use in project?
7.servlet lifecycle?
8.diff b/w requestdispatcher.forward(-,-) anf
include,sendRedirect
9.what is <load-on-start-up>
10.difference b/w ClassNotFoundException ans
NoClassDefFound Error?
11.what is bean scope?
12.how 2 configure 2 database in spring mvc manually?
==========
==========
TCS PUNE
==========
==========
1.explain your project
2.selling features of java
3.what are the oops principles
4.what is encapsulation,abstraction
5.difference b/w dynamic polypharphism and static
polymarphism
6. what is inheritence
7.when you go for interface and abstract class
8.scenario on inheritence
9.can i sort arraylist ? how?
10.can i override static method
10.when we can go for comparable and comparator
11.can i decrease the access modifier visibility in
overriding ? if no why?
12.scenario on throwing exception
13.draw exception hierarchy
14.diff b/w checked and unchecked Exceptions
15.try ,catch is mandatory for checked exceptions.
16.can i have try with out catch and finally
17.when the finally block will execute

18.scenario on throwing multiple exceptions.
19.how to create customized exceptions
20.diff b/w throw and throws
20.what is the flow if i placed return in catch and finally block
21.scenario on placing Systrm.exit(1) in try/catch block
22.shall we create unchecked exception ?
23.what is syncronization? what is the need
24.difference b/w wait and sleep.
25.what are the collections you used in yuor project
26.diff b/w set and list
27.diff b/w arraylist and linked list
28.when we can use arraylist and linkedlist
29.how you sort arraylist
30. how union two lists.
31. how to intersect two lists.
32. how to find the duplicates in list
33.diff b/w map and list
34.how hashmap works
35.what is hash collision how it resolve
36.how the hashset find duplicates.
-------- SPRING-------
37.what is DI
38.what is IOC
39.types of injections
40.which is prefer when ?
41.autowireing
42.MVC flow
------hibernate-------
43.tell tags in configuration file and mapping file.
44.diff b/w get and load
------webservices-----
45.how you consume a webservice.
46.diff b/w SOAP and REST
-------designpattern------
47.singleton,strategy
==========
==========
BitWise Software solutions
==========
==========
1.what are selling features of java
2.diff b/w stringbuffer and builder? When we use buffer and builder.

3.how to perform search on collection
4.how to sort the collection 5.diff b/w throws and throw.
6.try,catch is mandatory for checked exceptions.
7.explain exception hierarchy.
8.what is polymorphism
9.diff between overloading and overriding.
10.what is inheritance.
11.explain singleton design pattern.
12.explain strategy design pattern.
13.explain service locator design pattern.
14.what is Spring AOP and why?explain some logics that can managed by AOP.
15.what is transaction.
16.how hashmap works.
17.can I store two values with same key in map.
18.diff b/w hashmap and treemap.
19.for searching which collection is preferable.
20.diff b/w arraylist and linkedlist.
21.diff b/w SAOP and rest.
22.servelt is threadsafe or not,how you make it threadsafe.
23.how many object are created for servlet.
24.what are the configuration tags used for AOP.
25.what is aspect.
PM round:
Explain about you and your project.
1.explain difference between WaterFall methodology and Agile
2.who is assign the task to you in scrum.
3.where the story points are placed.
4.what is the duration of stadup meetings.
5.what is product backlog
6.how you know that the new sprint is started.
7.which tool is using in scrum to assign the issue.
8.where you checkin your code.
9.what is JUnit?why programmer has to do testing.
10.how the tester know that the build is completed.
11.while using hibernate have experienced any performance issue.
12.how you test WebService that you developed.
13.how you managed transactions in hibernate.
14.what is your sprint duration.
15.why Agile is preferable than waterfall.
16.what is DAO why DAO.
17.when you go for mocking.

==========
1. Did u develop webservices for internal usage of u r organization?
2. difference between comparable and comparator interfaces.
3. what are other ways of acquiring lock other than sychronized keyword.
4. give me negative and positive test cases for fibonacci series from 1 to 10 and bva(boundary value analysis).
5. what is tranisitive dependency in maven
6. why maven when ant is also a build tool
7. when should i do unit testing?
8. what are the integration tools u r aware of ?(he excepts jenkins,hudson...)
9. tell me spring mvc flow?
10. why should i go for two ioc container for spring mvc?
11. what are the modules present in spring framework?
12. difference between first-level and second-level caching in hibernate.
13. difference between get() and load() in hibernate
14. why hibernate when other orm frameworks are available?
15. how to do one-many mappings in hibernate
==========

Q.tell me the annotations present in jax-rs api and where u used those annotations in u r project.
Q.what are your daily activites while developing restful/ soap based web services u
Q.difference between @Controller and @RestController
Q.what is xml? where exactly u used xml in u r project?can u define u r own xml.how do u verify u r xml is correct or wrong.
Q.benefits/usage of IOC container?
Q.how do u enable security in restful services?
Q. why should i go for restful services.
Q. how do u provide u r restful services to other person?
Q. tell me a usecase where u can suggest developing webservices rather then a web application w.r.t u r present organisation
==========
1. why webservices??
2. what is contract first approach and contract last

approach
3. how do we communicate with two remote systems/different machines without web services (or) when webservices was not available.
4. what are the different implementation for jax-ws
5. explain me the procedure to develop a provider using apache-axis2 implementation.
6. explain wsdl?
7. disadvantages of soap-based services.
8. given me a wsdl told me write sei interface
9.as well given me sei interface write corresponsing wsdl parts.
10. Difference between restful services and soap-based services.
11.Are u aware of client api in restful services?
12. when should we go for soap-based services /restful services
13. tell me some bugs u encountered while developing soap-based/restful services
==========
what is fail fast and fail safe iterator in java.
what is concurrentmodificationException?
when should we go for TreeMap and why?
explain internal structure of hashmap why should i go for quicksort when compared with bubblesort?
how to create my own immutable class.
what are the measures u will take while developing a web application?
==========
Interview Questions:(myself sandeep attended 7 interviews, myfriend ravi teja, he atteneded 11 interviews. these are all the questins they asked)
1.project architecture?
2.spring flow?
3.list out all the annotations in spring?
4.what are all the annotations you used in your project on restful services?
5.in hibernate use of cascade and inverse?
6.first level cache? second level cache in hibernate?
7.waht is diff b/w arraylist and linkedlist?
8.can you explain the internal flow of hashmap?
9.what is the diff b/w hashmap and hashtable?

10.diff b/w array and arraylist?
11.diff b/w arraylist and vector?
12.in your project where you used cuncurrent hashmap?
13.waht is java annoying?
14.diff b/w callable interface adn future interface in concurrent package?(ravi)
15.class loaders?
16.how can you take list into map?
17.how can you take map into list?
18.when you will get ClassNotFoundException and NoClassDefFoundError?
19.how you implement exception handling in your project?
20.where you implement multi-threading in your project?
21.what are all the design patterens you oberved in spring?
22.which design patters you used in your project?
23.what are all the critial situations you come across in your project?
24.why wait() placed in object class ? why not it is placed in Thread class?(ravi)
25.waht is use of intern() in spring?
26.what is diff b/w String str="sandeep"; and String str2= new String("sandeep");
27.expalin about java architectue?
28.explain about jvm architecture?
29.data base queries?
30.i have a compeny table in remote database. by using rest i need to get the table data and print into a file?
31.how to read book pages on online library by using bookid or author id(by using restful services)?
32.i have a table in remote database, how to update the data in that table using rest?
33.diff b/w rest and web(soap)?
34.agile methodolgy?
35.how to create web-services project and spring project using mavan?
36.what is diff b/w throw and throws?
37.can you tell me java8 features?
38.what are all the contents in wsdl?
39.refer regular expressions?
40.can i add elements to list , if it is defined as final ex:final List<String> list= new ArrayList<>();?
41.if you pass duplicate key to map what will happen?
42.diff b/w abstract class and interface?
43.diff b/w comparator and comparable?

44.how to compare two database tables(clue: comparator, compare(), you have to compare database objects.)?
45.how to set timeout for the browser?(clue: restful client api.)?
46.what workflow you used in your project?
47.why java? why not c & c++?
48.In written test they are asking sorting programs(bubblesort,quicksort,...)?
49.what is time complexity? if you are going to implement sorting by your own which sorting you prefer? and why?
50.what is the use of volatile and synchronized?
51.what is serialization? have you implement serialization in your project?
52.programs on io streams?
53.why we are using @qualifier?
54.diff b/w BeanFactory and ApplicationContext?
55.explain about ioc container?
56.programs on string manipulations?(they are expecting solv by using regular expressions).
57.how you are implemented polymorphism in your project?
58.how you iterate map having key and list<values>?
59.diff b/w Iterator and ListIterator and Enumarator?
60.what are all the collections are supporting ListIterator?
61.how to make non-synchronized map and list as synchronized(by using collection method)?
62.what is diff b/w collection and collections?
63.write the junit test case for the below senario..
-->read array of elements into list<>.
64.what are all the modifiers we can use inside method? (ans: only final)
65.what is diff b/w spring-jdbc and hibernate?
66.what are all the drawbacks of jdbc over hibernate?
67.what are all the problems with inheritance?
68.what is the use of hinernate session?
69.they given one query in sql and they are asking corresponding criteria api query?
70.why we are using @transient in hibernate?
71.what are all the inputs we are giving to SessionFactory?
72.what we are writing in hibernate-mapping file?
73.what we are writing in hibernate-configuration file?
74.senario: in jsp page with 2 buttons, one for addbook and another is for showListOfBooks(by using spring and hibernate)?

75.what is use of @ComponentScan?
76.what is use of dispather servlet?
77.what are all the pre-processings tasks done by DispatcherServlet?
78.how to render excel and pdf view to the enduser(using poi and itext api's)?
79.how to validate valid username and password in spring? for validating can i directly interact with dao without service?
80.by defalut servlet container will handle multi-threaded applications , then why you are implementing multi-threading in your application?

suggesions: In banglore most of the compenies conducting written test for 3+ years of experience also,
some compenies expecting knowledge on Spring bOOT?
hope so it will be helful for you all, all the best to all :)
==========
==========
1.what is a volatile keyword where it is used?.
2.diff between comparator and comparable?.
3. beanScopes?.
4.bean lifecycle?.
5.IOC?.
6.what is dependency injection?.
7.what is get() and load()?.
8.what is a hash map?.
9.what is a singleton class?..how to prfem u r class is a singleton?.
10.who is u r project client ?.
11.company address?.tell me u r official mail id?.
12.company contact no?.

1.what is a java?
2.java satisfied with oops what are they?.
3.how to defeine u r class is a serializable class?.
4.what are the hibernate API class?.
5.what is diff between overloading and overriding?.
6.what is runtime polymerphism?.
7.rumtime polymorphism usecase(my class is bank interfac and a m having two classes i,e HDFC and ICICI these two are my class how to declare these two are runtime ploymerpshim)?.

8.what is serialization?..in u r class how to prfem u r class is a serialized or not ?.(ex:a m having 5 employees data but i want declare 3 employees data should be serializable how it is possible(by using volatile key word))?.
9.1st i want create the data in database and then after that i want access that data how u can follows to access that data?.

1.tel me about u r self?.
2.what is a searilazation y we r using serialization?when u r using serialization in u r project?
3.what is a Maven plugins?.
4.how to write the test caeses in u r project?
5.expalin u r project?.
6.how to identifies the duplicates in a arraylist?.
7.what is a volite keyword?.
8.what are the core concepts of Spring?.
9. dispay 3 values by using foreach or hshmap or hashtable?.

1.what is diff between webservices and restful?
2.how many ways to create the thread?.
3.what is a diff between hashmap and hashtable?.
4.where u r using hashmap in u r project?.
5.what is IOC?.
6.what is a autowiring?.
7.what is a lazy initilizer?.
8.write a query for the inner join?.
9.i want dispaly 3 values by using hashmap?.
10.where u r using extends and implements?.

1.what is the diff between .equals and == equals how to difine it?.

1.tell me about u r self?
2.what are the Jsp tags?.
3.what is a servlet?.
4.diffrenece between webservices and restful services?.
5.what is Soap?.
6.what is a interface?.

7.how many ways to create a thread?.
8.do u know about oracle?.
9.do u know javascript?.
10.explain about u r project?.
11.diff between soap and json?.
12.what is response object in webservice or html?.
==========
==========
First technical round:-
==========
1.What is ConcurrentModificationException ,Write one Sample programme for it and how to resolve it.. 2.What is method overloading ,write sample code with real world example
3.Did u use collection in your project ,where ?
4.Write Sample dao class and configuration in spring annotation
5.Wait(),notify() write code that one thread will print even and another will print odd sequentially ? 6.Reverse an integer value without using String class ?
7.Where you are using webservices in your project architecture
8.Develope one sample Rest Resource with all the HttpMethod based on your project senario
9.How u display huge amount of records data in UI ?
10.which server you are using to develope rest api ?
11.Version no of your server?
12.Which dataSource you are using ?
13.What is Load on Start up how it works ?
14.Which response you are providing to UI through Rest
Second round technical:
==========
1.Project Description
2.What is WSDL ? which MEF u r using and why ?
3.Which approach u r using to develope Webservice and why ?
4.How u can write wsdl if i have huge amount of data read from database means Sample
Ex:-
==========
interface xyz{ public List<Product> getAll(); }
5.R u used Singletone in your project why and where ?
6.How u deploy your code to see the functionality..
7.Did u ever try to see the functionality of your project UI-->SERVICE--->DAO (End to End) how u check this one
8.Defect test which tool you are using ?

9.What is mvn and what is the advantages ?
Third round:-(P M R)
==========
1.Project Functionality (Non-technical flow)
2.How u write Junit case write a sample code
3.Which methodlogy u are using tell me ur complete work responsibility
4.Sprint duration estimate ?
5.Puzzle --- i have 8lt ,5lt nd 3lt jar ...but i have all total 8 lt water then tell me how to manage 4lt nd 4lt in first 2 jar ..u can use 3lt jar for measure and balancing
6.i have 9 ball and 1 ball is more weight and i have one measurement device/Weight balancer machine how can u find out heavy weight ball tell me within 2 step
7.If your junit test case fail then what wil u do
Fourth round (H.R):
==========
1.Why u looking for change ?
2.Current CTC ?
3.Expected CTC ?
4.Are you agree for relocate ?
5.Notice Period ?
6.How u came today ?
7.in Which zone u r working (Company address) ?
8.Are u handel both develop and testing part ?
9.Have u passport/pancard ?
10.tell me your parent company location ?
==========
==========
1. which transaction ur using? and how to do it in ur project?
2.how many types of servlets?
3.how to maintain global data in ur application?
4.diff b/w xml and xsd?
5. what is jax-B and jax-p?
6.how to send and receive xml data as part of request in rest?
7.what is AOP and where u use aop in your application?
8.use case like :
two different client has two different databases with same data, you have both the references of databases when client B change in db how to apply transactions(lets check this use case)
==========

==========

1. Difference between annotation and interface?
2. Difference between String, StringBuffer, StringBuilder?
3. Internal data structure used in AL?
4. Find duplicate numbers from an Array?
5. String anagram?
6. Occurrence of each word in String?
7. Java is what pass by value or reference, justify?
8. Oops concept?
9. How to create custom exception?
10. How to create custom error?
11. Servlet life cycle?
12. Jsp implicit objects?
13. What is IOC?
14. Different types of Injection?
15. How to make employee object as key in hashmap?
16. What is PrepareStatement?
17. What are the methods available in resultset?
18. How to traverse a resultset in reverse order?
19. How to update data in resultset?
20. What is Agile? How you are using agile in your project?
21. Difference between Agile and Waterfall?
22. MVC architecture?
23. How to integrate WS with Spring?
24. How to integrate Restful with Spring?
25. Types of IOC?
26. Difference between ApplicationContext and BeanFactory?
27. How to manage transaction if there are multiple transactional resources(database, jms)?
28. What is spring transaction?
29. How to execute store procedure in java?
30. Difference between HashMap and HashTable?
31. What happen when we are using annotation?
32. Http status codes?
33. Difference between put and post?
34. What are the different http methods?
35. What is Hibernate Dialact?
36. What is HibernateTemplate and what are the advantages of using HibernateTemplate?
37. Define your project architecture?
38. What is CI? How to use Hudson?
39. What is CDN server? Why it is used?
40. How website is hosted?
41. What is scrollable?

42. How to redirect a .htm page to a .jsp page in spring?
43. If there is a css file for a table which is included into a jsp page and if you are overwriting the stylesheet in the jsp file for the table then what will happen?Will it reflect the jsp styles or css file style?
44. How to make a class singleton? What happen if I have a custom classloader , will the singleton class will work or not?
45. Define the classloader hierarchy?
46. Bean Life cycle?
47. Bean scopes?
48. Difference between Struts and Spring?
49. Difference between SOAP and Rest?
50. When destroy method will be called in a servlet?
51. How to create a session inside a servlet?
52. What are the exceptions thrown by a DAO class in spring?
53. If there is a complex query how do you execute it? Will you take the help of hibernate or native sql query?
54. How to normalize with the help of Hibernate?
55. What are the new features added in jdk1.8, explain?
56. Annotation support is there in Hibernate3 and Hibernate4?
57. What is the percentage of code coverage in your project?
58. What are the different ways to crate a servlet?
59. Which method will be called in a servlet when a request is coming?
60. If multiple requests are coming how many objects will be crated for the servlet?
61. What is a marker interface?
62. What do you mean by persistency?
63. What is DriverManager?
64. Difference Aggregation and Composition?
65. What is caching in Hibernate? What are different levels of caching available?
66. What are the other ORM frameworks available in market? What is the difference between them and Hibernate?
67. What is Mocking? What are the libraries available in market? Which one you are using?
68. What is PrepareStatementCreator?
69. If you are using ApacheCXF and if you want to expose them as rest resources, then how to do it?
70. How did you get the offer letter from your previous

company?
71. Why are trying to change the company?
72. Why do you want to join a startup company(Asked in a startup product based company)?if you got a offer letter from Congnizant or any big company and if you got another offer letter from a startup company then which one will you join(Both are offering the same salary package)?
73. Why did you came to Bangalore?
74. How much salary you are expecting ? Justify ?
75. What do you mean by Techno function?(If you are mentioning in your resume that you are Techno function then they are asking , How do you became a Techno function within 3year)?

==========

Company Requirement is Soap Based Web Services::

What kind of approaches you fallowed in developing web services?

can u write more than one service in wsdl?(they try to confuse)

what is jax-p and jax-b?

how exactly Marshalling and Unmarshalling in web services?

if in your xml you are writing element "name", but in your dto your attribute name is "firstname"
how it will map in jaxb? how can u map this?

explain about handler in web service?

How can u handle security in web services?

if we are imposing security in provider,
then how can a client knows that provider impose security?
(i failed in answering this question. can any one post the answer for this question, if possible answer all questions)

REST : if your resource is accepting xml, but if your are passing json obj, then what will happen?

what is structure of json?

REST : i want to pass ids(1,2,3,...) to server, how can u pass to server?

REST : what is matrix parameter? what kind of additional information you can pass?

Define class and object?

why string is immutable? can u write your own immutable class? are we need make all methods are final or not?

can we put private for a class?

what is default and protected?

i have a static method in class? can u override in its sub class? if we override what will happen?

==========

1st round:

1.What is immutable and what is the use of it?why we need to make a class as immutable?
2.What is aop and explain about it.where you have used in your project?
3.What is transaction and what are the propgation levels and explain with an example?
4.What are the code coverage tools you are using?
5.What are the frameworks you are using for unit testing?
6.What is synchronization and why it is needed?
7.Explain about JMS?
8.How did you use dependencies in maven(how did you overcome the transitive dependencies)?
9.Explain about log4j?
10.What are the design patterns you know?
11.Explain about builder design pattern?
He stressed alot in JMS
In all interview they have asked so many questions on webservices(SOAP and REST).
What is wsdl?
How do consume a webservice?

==========

1.What are the core principles of java and explain with an example
2.what is overloading
3.what is overriding
4.what is interface and when do you go for interface?
5.what is exception(he covered all areas of exceptions)
6.Threads(All areas, Question: If you get a chance how do you write a scheduling algorithm)
7.Servlets(life cycle)
8.What is singleton and explain with an example.

she has asked real time scenarios in hibernate and jdbc.
==========
Tell me the annotations in Spring MVC
Tell me the Annotations in Hibernate if you want to replace mapping file with Annotations
Can we Have Multiple Dispatcher Servlets?
Where did you place spring files in (web.xml)
Tell me about SessionFactory
Tell me the Core Interfaces of Hibernate
What is Criteria Api And How to write
What is HQL how to write?
What is Connection Pool When will you use?
What is the difference between Servlet and Dispatcher Servlet?
How to make a class as Servlet?
What is the Difference between doGet and doPost?
I have so many forms in my application Which method is safe to use?
Did you faced any Out of Memory Exceptions If so How did you resolved it?
What is the difficult situation faced in your project recently and Explain?
Do you fix Production Server(Application deployed in the Production server) issues?
Do you create the on your own or will you take the help from Database developer?
What is HibernateTemplate and what is the use of it and what are the benefits?

DataBase:
Can we Have Multiple Primary Keys in a table?
Can we have Multiple Foreign keys in a table?

What is Composite key?
What is Unique Key?
Can we Have Multiple Indexes in a table?
What are the joins available?
what is the difference between LEFT JOIN and RIGHT JOIN?

Java
----

which version are you using?
what is the difference between jdk1.5 and jdk1.6?
which server are using application server or web server?
In which folder you will deploy the code in jboss and how?
How to compare objects in java?
how to compare list of objects in java?
how to compare two objcets in java?
can you please tell me about comparator and comparable?
how to sort list of objects?
what is servlet and how to make a servlet?
what is filter and what is the use of filter?
if any changes are done in schema,In which pages we need to change?
what is the default scope of a bean?
Query related to employee and department?
have you used JMS?
how to convert json to java object?
==========

JAVA 9 INTERVIEW QUESTIONS & ANSWERS

Question 1. What Features Module System Has Provided?

Answer :

With the Modules component, following enhancements has been added in Java 9 :-

A new optional phase,link time, is introduced. This phase is in-between compile time and run time. During this phase, a set of modules can be assembled and optimized, making a custom runtime image using jlink tool.
javac, jlink, and java have additional options to specify

module paths, which further locate definitions of modules.
JAR format updated as modular JAR, which contains module-info.class file in its root directory.
JMOD format introduced, a packaging format (similar to JAR)

Question 2. What Are The Significant Changes In Java 9?

Answer :

There are 90+ enhancements added to Java 8, the most significant ones are mentioned below :-

Module :- A new kind of Java programing component introduced as module, which is a named, self-describing collection of code and data.
REPL (JShell) :- Read-Eval-Print Loop (REPL) capability added to the Java platform.
HTTP 2 Client:- new HTTPClient API supporting websockets and HTTP 2 streams and server push features.
Improved JavaDocs:- Supports HTML5 output generation. Provides a search box to generated API documentation.
Multirelease JAR:- Enhances the JAR format so that multiple, Java release-specific versions of class files can coexist in a single archive.

Question 3. What Is Module In Java 9?

Answer :

In Java 9, a new kind of programming component called module has been introduced. A module is a self-describing collection of code and data and has a name to identify it.

Question 4. What Is Jshell In Java 9?

Answer :

With JShell, java has REPL capability. Using JShell, we can code and test java based logic without compiling using javac and see the result of calculations directly.

Question 5. How Will You Create Html5 Compliant Javadoc Using Java 9?

Answer :

Run the javadoc tool of jdk 9 with -html5 flag to generate new type of documentation.

Question 6. Source Code Of A Module Lies In Which Folder?

Answer :

By convention, the source code of a module to lie in same directory which is the name of the module.

Question 7. What Is Multi-release Jar Format Introduced In Java 9?

Answer :

In java 9, a new feature is introduced where a jar format has been enhanced to have different versions of java class or resources can be maintained and used as per the platform.

Question 8. What Is The Full Form Of Repl?

Answer :

REPL:- Read-Eval-Print Loop.

Question 9. Can You Create A Multi-release Jar For Different Version Of Java? Give An Example?

Answer :

Yes! Following command will create a multi-release jar for java 7 and java 9 version.

```
$ jar -c -f test.jar -C java7 . --release 9 -C java9 .
```

Question 10. What Are The Methods Added To Collections In Java 9?

Answer :

With java 9, following methods are added to List, Set and Map interfaces along with their overloaded counterparts.

```
static <E> List<E> of(E e1, E e2, E e3);

static <E> Set<E> of(E e1, E e2, E e3);

static <K,V> Map<K,V> of(K k1, V v1, K k2, V v2, K k3, V
v3);

static <K,V> Map<K,V> ofEntries(Map.Entry<? extends K,?
extends V>... entries)
```

note:

For List and Set interfaces, of(...) method is overloaded to have 0 to 10 parameters and one with var args parameter.
For Map interface, of(...) method is overloaded to have 0 to 10 parameters.
In case of more than 10 paramters for Map interface, ofEntries(...) method can be used accepting var args parameter.

Question 11. Collections Are Enhanced In Java 9. What Are The Changes And Explain?

Answer :

With Java 9, new factory methods are added to List, Set and Map interfaces to create immutable instances. These factory methods are convenience factory methods to create a collection in less verbose and in concise way.

Question 12. How Can You Run A Multi Release Jar On Different Version Of Java? Give An Example?

Answer :

Syntax is same on both java versions, result will be different. Run with JDK 7.

```
C:JAVA > java -cp test.jar com.tutorialspoint.Tester

Inside Java 7
```

Run with JDK 9.

C:JAVA > java -cp test.jar com.tutorialspoint.Tester

Inside Java 9

Question 13. How Many Kind Of Variables/methods An Interface Supports In Java 9?

Answer :

With Java 9 interfaces can have following type of variables/methods:

Constant variables
Abstract methods
Default methods
Static methods
Private methods
Private Static methods

Question 14. What Is Default Project Structure For A Web Based Application?

Answer :

The following is the default project structure:-

The database scripts are stored in the db folder.
The java source code is stored in the src folder.
The images, js, META-INF, styles (css) are stored in the war folder.
The JSPs are stored in the jsp folder.
The third party jar files are stored in the lib folder.
The java class files are stored in the WEB-INFclasses folder.

Question 15. How Can You Check A System Process Details In Java 9?

Answer :

In Java 9 Process API which is responsible to control and manage operating system processes has been improved considerably. ProcessHandle Class now provides processes native processes ID, start time, accumulated CPU time,

arguments, command, user, parent process, and descendants.

Question 16. Explain The Use Of Dropwhile Method In Stream?

Answer :

dropWhile method throw away all the values at the start until the predicate returns true. It returns, in case of ordered stream, a stream consisting of the remaining elements of this stream after dropping the longest prefix of elements matching the given predicate.

Question 17. Can You Perform Some Function If A Process Exits?

Answer :

ProcessHandle class provides method to check processes' liveness and to destroy processes. It has onExit method, the CompletableFuture class can perform action asynchronously when process exits.

Question 18. What Are The Changes Made To Stream From Java 8 In Java 9?

Answer :

Streams were introduced in Java to help developers perform aggregate operations from a sequence of objects. With Java 9, few more methods are added to make streams better.

takeWhile
dropWhile
iterate
ofNullable

Question 19. What Are The Changes In Iterate Method Of Stream?

Answer :

iterate method now has hasNext predicate as parameter which stops the loop once hasNext predicate returns false.

Question 20. Explain The Use Of Takewhile Method In Stream?

Answer :

takeWhile method takes all the values until the predicate returns false. It returns, in case of ordered stream, a stream consisting of the longest prefix of elements taken from this stream matching the given predicate.

Question 21. Explain The Purpose Of Try-with-resource Statement?

Answer :

The try-with-resources statement is a try statement with one or more resources duly declared. Here resource is an object which should be closed once it is no more required. The try-with-resources statement ensures that each resource is closed after the requirement finishes. Any object implementing java.lang.AutoCloseable or java.io.Closeable, interface can be used as a resource.

Question 22. Explain The Use Of Ofnullable Method In Stream?

Answer :

ofNullable method is introduced to prevent NullPointerExceptions and to avoid null checks for streams. This method returns a sequential Stream containing single element, if non-null, otherwise returns an empty Stream.

Question 23. What Are The Changes Made To @deprecated Annotation In Java 9?

Answer :

With Java 9, two new enhancements are made to @Deprecated annotation:-

forRemoval:- Indicates whether the annotated element is subject to removal in a future version. The default value is false.
since:- Returns the version in which the annotated element

became deprecated. The default value is the empty string.
Question 24. What Changes Are Made To Optional Class In Java 9?

Answer :

Optional Class was introduced in Java 8 to avoid null checks and NullPointerException issues. In java 9, three new methods are added to improve its functionality.

stream()
ifPresent OrElse()
or()
Question 25. What Are The Changes Made To Diamond Operator In Java 9?

Answer :

In java 9, it can be used with anonymous class as well to simplify code and improves readability.

Question 26. Explain Purpose Of Completablefuture?

Answer :

CompletableFuture class was introduced in Java 8 to represent the Future which can be completed by setting its value and status explicitly. It can be used as java.util.concurrent.CompletionStage. It supports dependent functions and actions which got triggered upon the future's completion. In java 9 CompletableFuture API has been enhanced further. Following are the relevant changes done to the API.

Support for delays and timeouts.
Improved support for subclassing.
New factory methods added.
Question 27. What Is Multi-resolution Image Api In Java 9?

Answer :

With Java 9, a new multi-resolution image API has been introduced which supports multiple images with different resolution variants. This API allows a set of images with

different resolution to be used as a single multi-resolution image.

Following are major operations of multi-resolution image:-

Image set Resolution Variant(double destImageWidth, double destImageHeight):- Gets a specific image which is best variant to represent this logical image at the indicated size.
List get Resolution Variants():- Gets a readable list of all resolution variants.

****

SPRING BATCH INTERVIEW QUESTIONS & ANSWERS

Question 1. Explain Spring Batch Framework.

Answer :

Spring Batch frame work, a collaborative effort from Accenture and SpringSource, is a lightweight, comprehensive framework that facilitates the development of batch applications that helps the day to day activities of enterprise systems. Batch application or processing refers to automated offline systems that performs bulk data processing, periodic updates and delegated processing.

Examples include loading csv file data to database, process feed file once received and push daily transactions to the upstream or downstream systems.

Question 2. List Out Some Of The Practical Usage Scenario Of Spring Batch Framework?

Answer :

Reading large number of records from a database, file, queue or any other medium, process it and store the processed records into medium, for example, database.

Concurrent and massively parallel processing.
Staged, enterprise message-driven processing.

Sequential processing of dependent steps.
Whole-batch transaction.
Scheduled and repeated processing.
Question 3. Technical Advantages Of Using Spring Batch Framework From A Developer Perspective?

Answer :

Batch framework leverages Spring programming model thus allows developers to concentrate on the business logic or the business procedure and framework facilitates the infrastructure.

Clear separation of concerns between the infrastructure, the batch execution environment, the batch application and the different steps/proceses within a batch application.

Provides common scenario based, core execution services as interfaces that the applications can implement and in addition to that framework provides its default implementation that the developers could use or partially override based on their business logic.

Easily configurable and extendable services across different layers.

Provides a simple deployment model built using Maven.

Question 4. Explain The Spring Batch Framework Architecture?

Answer :

Spring Batch exhibit a layered architecture and it comprises of three major high level components:

Application, Core and Infrastructure.

The application layer contains all the batch job configurations, custom codes for business logic and job meta information developed by Application developers.

The Batch Core has the core runtime classes necessary to launch and control any batch job. Some of the core runtime

classes include JobLauncher, Job, and Step implementations.

The infrastructure contains API for common readers and writers, and services for retrying on failure, repeat jobs etc. The infrastructure layer are used both by application developers(ItemReader and ItemWriter) and the core framework itself for controlling the batch job such as Retry, repeat. Thus Batch Core and Application layers are built on top of Infrastructure layer.

Question 5. How Do You Categorize Batch Applications Based On Input Source In Spring Batch?

Answer :

Database-driven applications are driven by rows or values received from the database.

File-driven applications are driven by records or values retrieved from a file.

Message-driven applications are driven by messages retrieved from a message queue.

Question 6. What Are The Typical Processing Strategies In Spring Batch?

Answer :

Normal processing during offline.

Concurrent batch or online processing.

Parallel processing of many different batch or jobs at the same time.

Partitioning (processing of many instances of the same job at the same time).

Question 7. How Do I Start A Spring Batch Job?

Answer :

A Job Launcher can be used to execute a Spring Batch Job. Also a batch job can be launched/scheduled using a web container as well.

Execution of a job is termed as Job Instance. Each Job Instance is provided with an execution id which can be used to restart the job if required.

Job can be configured with parameters which is passed to it from the Job Launcher.

Question 8. What Are The Important Features Of Spring Batch?

Answer :

Restorability: Restart a batch program from where it failed.

Different Readers and Writers : Provides great support to read from text files, csv, JMS, JDBC, Hibernate, iBatis etc. It can write to JMS, JDBC, Hibernate, files and many more.

Chunk Processing : If we have 1 Million records to process, these can be processed in configurable chunks (1000 at a time or 10000 at a time).

Easy to implement proper transaction management even when using chunk processing.

Easy to implement parallel processing. With simple configuration, different steps can be run in parallel.

Question 9. Explain Normal Processing Strategy In Spring Batch Framework?

Answer :

Normal processing refers to the batch processes that runs in a separate batch window, the data being updated is not required by on-line users or other batch processes, where concurrency would not be a concern and a single commit can be done at the end of the batch run.

Single commit point may be a concern in terms of scaiability and volume of data it could handle, it is always a good practice to have restart recovery options .

Question 10. Explain Concurrent Batch On-line Processing In Spring Batch Framework?

Answer :

Concurrent/on-line batch processing refers to the batch process that handles data being concurrently used/updated by online users so the data cannot be locked in database or file as the online users will need it. Also the data updates should be commited frequently at the end of few transactions to minimize the portion of data that is unavailable to other processes and the elapsed time the data is unavailable.

Question 11. Explain Parallel Processing In Spring Batch Framework?

Answer :

Parallel processing enables multiple batch runs jobs to run in parallel to reduce the total elapsed batch processing time. Parallel processing is simpler as long as the same file or database table is not shared among the processes otherwise the processes should process partitioned data.

Another approach would be using a control table for maintaining interdependencies and to track each shared resource in use by any process or not.

Other key issues in parallel processing include load balancing and the availability of general system resources such as files, database buffer pools etc. Also note that the control table itself can easily become a critical resource.

Question 12. Explain Partitioning In Spring Batch Framework?

Answer :

Partitioning faciliates multiple large batch applications to run concurrently that minimize the elapsed time required to process long batch jobs. Processes which can be successfully partitioned are those where the input file can be split and/or the main database tables partitioned to allow the application to run against different sets of data.

Processes which are partitioned must be designed to only process their assigned data set.

Question 13. What Is Tasklet In Spring Batch Framework?

Answer :

The Tasklet is an interface which performs any single task such as setup resource, running a sql update, cleaning up resources etc.

Question 14. How Do I Configure A Job In Spring Batch Framework?

Answer :

A Job in Spring Batch contains a sequence of one or more Steps. Each Step can be configured with the list of parameters/attribute required to execute each step.

next : next step to execute

tasklet: task or chunk to execute. A chunk can be configured with a Item Reader, Item Processor and Item Writer.

decision : Decide which steps need to executed.

Question 15. What Are Spring Batch Metadata Schema?

Answer :

The Spring Batch Meta-Data tables are used to persist batch domain objects such as JobInstance, JobExecution,

JobParameters, and StepExecution for internally managing the Batch Jobs.

The JobRepository is responsible for saving and storing each Java object into its correct table

Question 16. Can We Create A Spring Batch With No Step?

Answer :

No. There must exists at least one step or flow or split configuration within a Spring Batch job.

Question 17. What Are The Different Bean Scope In Spring Batch 3.0?

Answer :

Step scope– there is only one instance of such a bean per executing step.

<bean id="..." class="..." scope="step">

Job scope– there is only one instance of such a bean per executing Job.

<bean id="..." class="..." scope="job">

Question 18. How Do We Track The Number Of Item Processed By The Itemreader In Spring Batch?

Answer :

The item mapping bean can implement org.springframework.batch.item. ItemCountAware, a marker interface to have the item position tracked.

Question 19. Define A Job In Spring Batch?

Answer :

A Job is an entity that encapsulates an entire batch process.

Job will be wired together using a XML configuration file or Java based configuration. This configuration is also referred as "job configuration".

A Job is simply a container for Steps and it combines multiple steps that runs logically together in a flow.

Question 20. What Is Job Launcher In Spring Batch Framework?

Answer :

JobLauncher represents a simple interface for launching a Job with a given set of JobParameters.

Question 21. What Is Executioncontext?

Answer :

An ExecutionContext represents a collection of key/value pairs that are persisted and controlled by the framework in order to provide the developers a placeholder to store persistent state that is scoped to a StepExecution or JobExecution.

Question 22. How Do You Run Spring Batch Jobs In Production Environment?

Answer :

Usually The Java batch Job main class and its dependencies are passed to the java command and it is stored in a command line Batch file or shell script in terms of linux/ unix.

These script file can be run using scheduler like Autosys at the Production environment.

Question 23. What Is Commandlinejobrunner In Spring Batch?

Answer :

CommandLineJobRunner is one of the ways to bootstrap your

Spring batch Job. The xml script launching the job needs a Java class main method as as entry point and CommandLineJobRunner helps you to start your job directly using the XML script.

The CommandLineJobRunner performs 4 tasks:

Load the appropriate ApplicationContext.

Parse command line arguments into JobParameters.

Locate the appropriate job based on arguments.

Use the JobLauncher provided in the application context to launch the job..

The CommandLineJobRunner arguments are jobPath, the location of the XML file that will be used to create an ApplicationContext and the jobName, the name of the job to be run.

bash$ java CommandLineJobRunner DailyJobConfig.xml processDailyJob

These arguments must be passed in with the path first and the name second. All arguments after these are considered to be JobParameters and must be in the format of 'name=value'.

Question 24. What Is Resourceaware Is Spring Batch?

Answer :

ResourceAware is a marker interface which will set the current resource on any item that implement this interface.

Question 25. Difference Between Spring Batch And Quartz Scheduler?

Answer :

Spring Batch and Quartz have different features and responsibility. Spring Batch provides functionality for processing large volumes of data while Quartz provides

functionality for scheduling tasks. Thus Quartz could complement Spring Batch and a common combination would be to use Quartz as a trigger for a Spring Batch job using a Cron expression.

Question 26. How Do I Schedule A Job With Spring Batch?

Answer :

Use a scheduling tool such as Quartz, Control-M or Autosys. Quartz islight weight, doesn't have all the features of Control-M or Autosys. Even the OS based Task scheduler, CRON jobs could be used to schedule Spring batch jobs.

Question 27. How Can I Make An Item Reader Thread Safe In Spring Batch?

Answer :

You can synchronize the read() method. Remember that you will lose restartability, so best practice is to mark the step as not restartable and to be safe (and efficient) you can also set saveState=false on the reader.

Question 28. What Is The Latest Version Of Spring Batch?

Answer :

The available latest version is 3.0.7.

Question 29. What Is Itemreader In Spring Batch Framework?

Answer :

ItemReader is an abstraction that represents the retrieval of input for a Step, one item/row/record at a time. When the ItemReader has exhausted the items it can provide, it will indicate this by returning null.

Question 30. What Is Itemwriter In Spring Batch Framework?

Answer :

ItemWriter is an abstraction that represents the output of

a Step, one batch or chunk of items at a time. Generally, an item writer has no knowledge of the input it will receive next, only the item that was passed in its current invocation.

Question 31. What Is Itemprocessor?

Answer :

ItemProcessor is an abstraction that represents the business processing of an item. While the ItemReader reads one item, and the ItemWriter writes them, the ItemProcessor provides access to transform or apply other business processing. If, while processing the item, it is determined that the item is not valid, returning null indicates that the item should not be written out.

Question 32. Mention The Different Itemreader And Itemwriter Implementations Available In Spring Batch?

Answer :

There are many implementations including the ones that allow read and write operations on,

Flat File.
Xml.
Hibernate Cursor.
JDBC.
JMS.
Hibernate Paging.
Stored Procedure.
Question 33. Name Few Of The Domain Buzzwords In Spring Batch?

Answer :

Job.
JobLauncher.
JobRepository.
JobInstance.
JobExecution.
JobParameters.
Question 34. What Job Configuration Consists Of?

Answer :

The job configuration contains,

The simple name of the job.
Definition and order of the Steps.
configurable global(to all steps) properties such as restartablity.
Question 35. What Is Jobinstance?

Answer :

A JobInstance represents the concept of a logical job run.

Question 36. Difference Between Step And Stepexecution?

Answer :

A Step is a domain object that encapsulates an independent and sequential phase of a batch job while a StepExecution represents a single attempt to execute a step.

Question 37. Define Executioncontext?

Answer :

An ExecutionContext represents a collection of key-value pairs that are persisted and controlled by the framework in order to allow developers a place to store persistent state that is scoped to a StepExecution or JobExecution.

Question 38. What Are The Required Dependencies For Configuring A Job?

Answer :

There are 3 required dependencies:

Job name,

JobRepository,

and one or more steps.

Question 39. How Do I Setup Spring Batch Job Without Using Xml?

Answer :

Spring 3 enables the ability to configure applications using java instead of XML and from Spring Batch 2.2.0, batch jobs can be configured using the same java config.

There are 2 components for the java based configuration:

the @EnableBatchConfiguration annotation and two builders.

@EnableBatchProcessing provides a base configuration for building batch jobs.

The core interface for this configuration is the BatchConfigurer. The default implementation provides the beans to be autowired such as JobRepository, JobLauncher.

Question 40. Explain The Role Of Jobrepository In Spring Batch?

Answer :

The JobRepository is used for basic CRUD operations of the various persisted domain objects within Spring Batch, such as JobExecution and StepExecution. It is required by many of the major framework features, such as the JobLauncher, Job, and Step.

Question 41. What Is The Default Isolation Level Of Spring Batch Transactions?

Answer :

It is SERIALIZABLE by default to prevent the same job instance being executed concurrently.

Question 42. What Is A Cron Job?

Answer :

A cron job is a Linux command for scheduling script on your server to execute repetitive tasks automatically. Scripts executed as a cron job are typically used to modify files, databases and manage caching.

Question 43. How Cron Job Works In Linux?

Answer :

Cron is a daemon that executes scheduled commands. Cron is started automatically from /etc/init.d on entering multi-user runlevels. Cron searches its spool area (/var/spool/cron/crontabs) for crontab files (which are named after accounts in /etc/passwd); crontabs found are loaded into memory.

Cron wakes up every minute, examining all stored crontabs, checking each command to see if it should be run in the current minute. When executing commands, any output is mailed to the owner of the crontab (or to the user named in the MAILTO environment variable in the crontab, if such exists).

****

MAVEN INTERVIEW QUESTIONS & ANSWERS

1) Explain what is Maven? How does it work?

Maven is a project management tool. It provides the developer a complete build lifecycle framework. On executing Maven commands, it will look for POM file in Maven; it will run the command on the resources described in the POM.

2) List out what are the aspects does Maven Manages?

Maven handles following activities of a developer

- Build
- Documentation
- Reporting
- Dependencies

• SCMs
• Releases
• Distribution
• Mailing list

3) Mention the three build lifecycle of Maven?

• Clean: Cleans up artifacts that are created by prior builds
• Default (build): Used to create the application
• Site: For the project generates site documentation

4) Explain what is POM?

In Maven, POM (Project Object Model) is the fundamental unit of work. It is an XML file which holds the information about the project and configuration details used to build a project by Maven.

5) Explain what is Maven artifact?

Usually an artifact is a JAR file which gets arrayed to a Maven repository. One or more artifacts a maven build produces such as compiled JAR and a sources JAR.

Each artifact includes a group ID, an artifact ID and a version string.

6) Explain what is Maven Repository? What are their types?

A Maven repository is a location where all the project jars, library jars, plugins or any other particular project related artifacts are stored and can be easily used by Maven.

Their types are local, central and remote

7) Why Maven Plugins are used?

Maven plugins are used to
• Create a jar file
• Create war file
• Compile code files
• Unit testing of code

• Documenting projects
• Reporting

8) List out the dependency scope in Maven?

The various dependency scope used in Maven are:

• Compile: It is the default scope, and it indicates what dependency is available in the classpath of the project
• Provided: It indicates that the dependency is provided by JDK or web server or container at runtime
• Runtime: This tells that the dependency is not needed for compilation but is required during execution
• Test: It says dependency is available only for the test compilation and execution phases
• System: It indicates you have to provide the system path
• Import: This indicates that the identified or specified POM should be replaced with the dependencies in that POM's section

9) Mention how profiles are specified in Maven?

Profiles are specified in Maven by using a subset of the elements existing in the POM itself.

10) Explain how you can exclude dependency?

By using the exclusion element, dependency can be excluded

11) Mention the difference between Apache Ant and Maven?

Apache Ant Maven
• Ant is a toolbox – Maven is a framework
• Ant does not have formal conventions like project directory structure – Maven has conventions
• Ant is procedural; you have to tell to compile, copy and compress – Maven is declarative (information on what to make & how to build)
• Ant does not have lifecycle; you have to add sequence of tasks manually – Maven has a lifecycle
• Ant scripts are not reusable – Maven plugins are reusable

12) In Maven what are the two setting files called and what are their location?

In Maven, the setting files are called settings.xml, and the two setting files are located at

- Maven installation directory: $M2_Home/conf/settings.xml
- User's home directory: ${ user.home }/ .m2 / settings.xml

13) List out what are the build phases in Maven?

Build phases in Maven are

- Validate
- Compile
- Test
- Package
- Install
- Deploy

14) List out the build, source and test source directory for POM in Maven?

- Build = Target
- Source = src/main/java
- Test = src/main/test

15) Where do you find the class files when you compile a Maven project?

You will find the class files ${basedir}/target/classes/.

16) Explain what would the "jar: jar" goal do?

jar: jar will not recompile sources; it will imply just create a JAR from the target/classes directory considering that everything else has been done

17) List out what are the Maven's order of inheritance?

The maven's order of inheritance is

- Parent Pom
- Project Pom
- Settings
- CLI parameters

18) For POM what are the minimum required elements?

The minimum required elements for POM are project root, modelVersion, groupID, artifactID and version

19) Explain how you can produce execution debug output or error messages?

To produce execution debug output you could call Maven with X parameter or e parameter

20) Explain how to run test classes in Maven?

To run test classes in Maven, you need surefire plugin, check and configure your settings in setting.xml and pom.xml for a property named “test.”

Question 21. What Are Different Types Of Build Profiles?

Answer :

Build profiles are of three types :

Per Project – Defined in the project POM file, pom.xml.
Per User – Defined in Maven settings xml file (%USER_HOME%/.m2/settings.xml).
Global – Defined in Maven global settings xml file (%M2_HOME%/conf/settings.xml)
Question 21.How Can You Activate Profiles?

Answer :

A Maven Build Profile can be activated in various ways –

Explicitly using command console input.
Through maven settings.
Based on environment variables (User/System variables).
OS Settings (for example, Windows family).
Present/missing files.

****
****

core java logical questions

Logical interview programs

1.> wtite a program fabonici series.
2.> how to convert binary to decimal and decimal to binary conversion.
3.> write a program to swap two numbers without using third variable.
4.> write a program factorial number.
5.> write a program to check the given number is palindrome or not.
6.> write a program a given number is prime or not.
7.> write a program to check number is armstrong or not.
8.> how to write custom exception in java.
9.> how to create immutable class.
10.> how to create singleton class.
11.> how to create deadlock between two threads.
12.> write a program to implement hashcode() and equals().
13.> how sort element using comparable and comparator.

String interview programs

1.> how to reverse a string in java without using any API.
2.> how to find duplicate charcater(no of occurance) in a string in java.
2.> how to count occurance of each character in a string in java.
3.> how do you remove all white spaces from a string in java.
4.> how to check given String is palindrome or not.
5.> how to check if two String are anargms.
6.> how to find duplicate character in a String.

Array interview programs

1.> write a java program to find duplicate element in an array.
2.> write a java program to find second largest number in an array.
3.> how to find missing number in the array.
4.> how to find number occuring odd number of times in an

array.
5.> Write a java program to count occurrences of each element in an array.
7.> How to find the missing element in integer array of 1 to 7.
8.> How to cut or remove an element from the array.
9.> How to get largest and smallest number in an array.

Collections interview programs

Arraylist

1.> basic arraylist operation.
2.> how to read all element in arraylist by using iterator.
3.> how to remove duplicate element in arraylist in java.
4.> how to delete all element in array list.
5.> how to sort arraylist using comparator.
6.> how to convert array to arrylist and arraylist to array in java.
7.> write a java program to sort an arraylist.

Linkedlist

2.> how to reverse linked list in java.
3.> how to find middle element of linked list in java.
4.> how to find nth element from end of linked list in java.
5.> how to reverse linked list in pairs.
6.> basic operation of linked list.
7.> how to remove element from linked list.
8.> how to add element in last position in linked list.
9.> how to add element in first position in linked list.

Hashmap

1.> basic hashmap operation.
2.> how to iterate hashmap.
3.> how to search key in hashmap.
4.> how to search value in hashmap.
5.> how to delete all elements from hashmap.
6.> create and add obj in hashmap.
7.> retrieving value from hashmap.

Linked hashmap.

1.> basic linked hashmap operation.
2.> how to iterate through linked hashmap.
3.> how to delete all element from linked hashmap.
4.>

Hashtable

1.> basic hastable operation.
2.> how to iterate through hashtable.
3.> how to search key in hashtable.
4.> how to search value in hashtable.
5.> how to delete all element in hashtable.

Hashset

1.> basic hashset operation.
2.> how to iterate hashset.
3.> how to delete all element in hashset.

linked hashset

1.> basic linked hashset operation.
2.> how to iterate through linked hashset.
3.> how to delete all element from linked hashset.
4.> how to delete specific element from linked hashset.
5.> how to search an element from linked hashset.

TreeSet

1.> basic treeset operation.
2.> how to create treeset with a list.
3.> how to read obj from treeset using iterator.
4.> how to create treeset using comparator.

TreeMap

-> basic treemap operation.
-> how to iterate through treemap.
-> how to search key in treemap.
-> how to search value in treemap.
-> how to delete all element in treemap.
-> how to sort keys in treemap using comparator.

-> how to get first key element from treemap(sorted map).
-> how to get last key element from treemap(sorted map).

How to find duplicate element element in an array

--
-> public class DuplicateElement {

public static void main(String[] args) {

String[] strArray= {"java", "spring", "hibernate", "java"};

for(int i=0;i<strArray.length-1;i++){

for(int j=i+1;j<strArray.length;j++){

if((strArray[i].equals(strArray[j]))&&(i!=j)){

System.out.println("duplicate:"+ strArray[j
}}}}}.

How to iterate array in java (using normal for loop).

-> public class Array{

public static void main(String[] args){

int[] a1 = new int[]{45, 12, 78, 34, 89, 21}; //Iterating
over an array using normal for loop

for (int i = 0; i < a1.length; i++) {

System.out.println(a1[i]);

}}}

using for each loop

-> public class Array{

public static void main(String[] args){

int[] a2 = new int[]{45, 12, 78, 34, 89, 21}; //Iterating
over an array using extended for loop

```
for (int i: a2){

System.out.println(i);

}}}
```

How to find missing number an integer array.

-> Find the sum of n number using formula n=n*(n+1)/2.
-> Find the sum of elements present in given array.
-> Substract (sum of n numbers – sum of elements present in the array).

eg.

```
int[] arr1={7,5,6,1,4,2};
Missing numner : 3
int[] arr2={5,3,1,2};
Missing numner : 4

public class MissingNumber {

public static void main(String[] args) {

int[] arr1={7,5,6,1,4,2};
System.out.println("Missing number from array arr1:
"+missingNumber(arr1));
int[] arr2={5,3,1,2};
System.out.println("Missing number from array arr2:
"+missingNumber(arr2));

}

public static int missingNumber(int[] arr)
{
int n=arr.length+1;
int sum=n*(n+1)/2;
int restSum=0;
for (int i = 0; i < arr.length; i++) {
restSum+=arr[i];
}
int missingNumber=sum-restSum;
return missingNumber;
}
}
```

How to find second largest number in array.

-> int[] arr1={7,5,6,1,4,2};
Second largest element in the array : 6

-> Initialize highest and secondHighest with minimum possible value.
-> Iterate over array.
-> If current element is greater than highest.
-> Assign secondHighest = highest.
-> Assign highest = currentElement.
-> Else if current element is greater than secondHighest.
-> Assign secondHighest =current element.

```
public class FindSecondLargestMain {
public static void main(String args[])
{
int[] arr1={7,5,6,1,4,2};
int secondHighest=findSecondLargestNumberInTheArray(arr1);
System.out.println("Second largest element in the array :
"+ secondHighest);
}

public static int findSecondLargestNumberInTheArray(int
array[])
{
// Initialize these to the smallest value possible
int highest = Integer.MIN_VALUE;
int secondHighest = Integer.MIN_VALUE;

// Loop over the array
for (int i = 0; i < array.length; i++) {

// If current element is greater than highest
if (array[i] > highest) {

// assign second highest element to highest element
secondHighest = highest;

// highest element to current element
highest = array[i];
} else if (array[i] > secondHighest)
// Just replace the second highest
```

```
secondHighest = array[i];
}


// After exiting the loop, secondHighest now represents the
second
// largest value in the array
return secondHighest;
}
}
output- Second largest element in the array : 6.

How to get largest and smallest number in an array.
-----------------------------------------------------------
-
-> We use two variables to store largest and smallest
number.
-> First, we initialize largest with Integer.MIN_VALUE and.
-> Next, we initialize smallest with Integer.MAX_VALUE.
-> In each iteration of the for loop, we will compare
present number with largest and smallest number, and we
will update.
-> If a number is larger than largest, then it cannot be
smaller than smallest. That means no need to check if the
first condition is true,
-> We will use the if-else code block, where else part will
only execute if the first condition is false means not
true.

public class MaximumMinimumArrayExample{

public static void largestAndSmallest(int[] numbers) {

int largest = Integer.MIN_VALUE;

int smallest = Integer.MAX_VALUE;

for (int number : numbers) {

if (number > largest) {

largest = number;
}
else if (number < smallest) {
```

```
smallest = number;

} }
System.out.println("Largest is : " + largest);

System.out.println("Smallest is : " + smallest); } }
```

****
****

Question 1. What Is The Base Class Of All Classes?

Answer :

java.lang.Object

Question 2. What Do You Think Is The Logic Behind Having A Single Base Class For All Classes?

Answer :

Casting
Hierarchical and object oriented structure.
Question 3. Why Most Of The Thread Functionality Is Specified In Object Class?

Answer :

Basically for interthread communication.

Question 4. Is String A Wrapper Class Or Not?

Answer :

No. String is not a Wrapper class.

Question 5. How Will You Find Length Of A String Object?

Answer :

Using length () method of String class.

Question 6. How Many Objects Are In The Memory After The Exaction Of Following Code Segment?

Answer :

```
String str1 = "ABC";

String str2 = "XYZ";

String str1 = str1 + str2;
```

There are 3 Objects.

Question 7. What Is The Difference Between An Object And Object Reference?

Answer :

An object is an instance of a class. Object reference is a pointer to the object. There can be many references to the same object.

Question 8. What Will Trim () Method Of String Class Do?

Answer :

Trim () eliminate spaces from both the ends of a string.

Question 9. What Is The Use Of Java.lang. Class Class?

Answer :

The java.lang. Class class is used to represent the classes and interfaces that are loaded by a java program.

Question 10. What Is The Possible Runtime Exception Thrown By Sub String () Method?

Answer :

ArrayIndexOutOfBoundsException.

Question 11. What Is The Difference Between String And String Buffer?

Answer :

Object's of String class is immutable and object's of StringBuffer class is mutable moreover String buffer is faster in concatenation.

Question 12. What Is The Use Of Math Class?

Answer :

Math class provides methods for mathematical functions.

Question 13. Can You Instantiate Math Class?

Answer :

No. It cannot be instantiated. The class is final and its constructor is private. But all the methods are static, so we can use them without instantiating the Math class.

Question 14. What Will Math.abs () Do?

Answer :

It simply returns the absolute value of the value supplied to the method, i.e. gives you the same value. If you supply negative value it simply removes the sign.

Question 15. What Will Math. Ceil() Do?

Answer :

This method returns always double, which is not less than the supplied value. It returns next available whole number.

Question 16. What Will Math. Floor () Do?

Answer :

This method returns always double, which is not greater than the supplied value.

Question 17. What Will Math.max () Do?

Answer :

The max () method returns greater value out of the supplied values.

Question 18. What Will Math.min () Do?

Answer :

The min () method returns smaller value out of the supplied values.

Question 19. What Will Math. Random () Do?

Answer :

The random () method returns random number between 0.0 and 1.0. It always returns double.

****
****

Question 1. What Is Thread In Java?

Answer :

A thread is a lightweight sub process. It is an independent path of execution. A thread is executed inside a process and one process can have multiple threads. All the threads inside a process shares a common memory area. Since threads are independent, if an exception occurs in one thread, it doesn’t affect other threads. Threads are used to take advantage of multiple CPU cores available in a machine.

Question 2. What Is The Difference Between Thread And Process In Java?

Answer :

A process is a program in execution. It can also be defined as a self contained execution environment.
A Thread is a single task of execution within the process.
One process can have multiple threads.
A process has its own memory space.
A thread uses the process's memory space and share it with the other threads inside the process.
Question 3. Can We Call Run() Method Of A Thread Class?

Answer :

Yes, we can call run() method of a Thread class but it will behave like a normal method and a new thread will not be created to execute the run() method. In this case the run() method will be executed in the same thread which called the run method. To actually execute it in a new Thread, we need to start it using Thread.start() method.

Question 4. How Does Thread Communicate With Each Other?

Answer :

Threads can communicate using wait(), notify() and notifyAll() methods. Read this post to understand inter thread communication.

Question 5. Explain About Thread Priority?

Answer :

Every thread has a priority, usually higher priority thread gets precedence in execution but it depends on Thread Scheduler implementation that is OS dependent. We can specify the priority of thread using Thread's setPriority(int) method but it doesn't guarantee that higher priority thread will get executed before lower priority thread. Thread priority is an int whose value varies from 1 to 10 where 1 is the lowest priority and 10 is the highest priority.

Question 6. What Is A Daemon Thread?

Answer :

Daemon threads are non-user threads. They are typically used to carry out low-priority tasks that should not take priority over the main task of the program. They can be used to do useful work when all other user threads are blocked. The garbage collector is one example of a daemon thread.
JVM terminates itself when all non-daemon threads (user threads) finishes their execution, JVM does not care even if some Daemon threads are running. If JVM finds running daemon thread (upon completion of user threads), it terminates the thread and after that shutdowns itself. You can make a user thread to Daemon by using setDaemon() method of thread class.
A child thread created from daemon thread is also a daemon thread.

Question 7. What Is Difference Between User Thread And Daemon Thread?

Answer :

By default a thread created in a Java program is always a user thread however we can make it daemon by calling setDaemon(true) method, if needed. A daemon thread runs in the background and doesn't prevent JVM from terminating. As soon as all user thread finishes execution, Java program or JVM terminates itself, JVM doesn't wait for daemon thread to finish their execution. As soon as last non daemon thread finished, JVM terminates no matter how many Daemon thread exists or running inside JVM.

Question 8. What Is Volatile In Java?

Answer :

volatile is a special modifier which is used to indicate that a variable's value will be modified by different threads. The volatile keyword will mark a Java variable as "being stored in main memory". The value of this variable will never be cached locally: all reads and writes will go straight to "main memory". Volatile variable guarantees that a write will happen before any subsequent read. Access to the variable acts as though it is enclosed in a

synchronized block.

Question 9. What Does Yield Method Of Thread Class Do?

Answer :

yield() method causes the currently executing thread object to temporarily pause and allow other threads to execute. If there is no waiting thread or all the waiting threads have a lower priority than the current thread, then the same thread will continue its execution. When the yielded thread will get the chance for execution is decided by the thread scheduler whose behavior is platform dependent.

Question 10. What Does The Join() Method In Thread Class Do?

Answer :

The join method allows one thread to wait for the completion of another. If t is a Thread object whose thread is currently executing, t.join() causes the current thread(the thread which calls t.join(), mostly the main thread) to pause execution until t's thread terminates.

Question 11. What Is The Difference Between Notify() And Notifyall()?

Answer :

notify() method wakes up a single thread that is waiting on this object's monitor. If any threads are waiting on this object, one of them is chosen to be awakened. The choice is random and occurs at the discretion of the implementation. notifyAll() wakes up all threads that are waiting on this object's monitor. A thread waits on an object's monitor by calling one of the wait methods.

Question 12. How Can You Ensure All Threads That Started From Main Must End In Order In Which They Started?

Answer :

We can use join() method to ensure all threads that started from main will end in order in which they started and also main should end in last.

Question 13. Why Thread Communication Methods Wait(), Notify() And Notifyall() Are In Object Class?

Answer :

In Java, wait and notify methods acts as synchronization utility and are essential methods for inter thread communication. Hence these methods are defined in Object class so that every object will have access to it. Also every Object has a monitor and Locks are made available on per Object basis. This is another reason why wait and notify is declared in Object class rather then Thread class.

Question 14. Can A Constructor Be Synchronized?

Answer :

No, constructor cannot be synchronized. Constructor is used for instantiating object and when we are in constructor, object is under creation. So, until object is not instantiated it does not need any synchronization.

Question 15. How Can You Access The Current Thread In Java?

Answer :

The current thread can be accessed by calling the static method currentThread() of the java.lang.Thread class. E.g. Thread.currentThread().getName().

Question 16. What Happens When An Uncaught Exception Occurs In The Run() Method?

Answer :

When an unchecked exception has occurred in the run() method, the thread is stopped by the Java Virtual Machine. It is possible to catch this exception by registering an instance that implements the interface

UncaughtExceptionHandler as an exception handler.
The handler can be registered by invoking the static method Thread.setDefaultUncaughtExceptionHandler(Thread.UncaughtExceptionHandler) or by invoking setUncaughtExceptionHandler(Thread.UncaughtExceptionHandler) on the thread instance which tells the JVM to use the provided handler in case there was no specific handler registered on the thread.

Question 17. What Do You Mean By An Atomic Operation?

Answer :

In programming, an atomic operation is one that effectively happens all at once. An atomic operation cannot stop in the middle: it either happens completely, or it doesn't happen at all. No side effects of an atomic operation are visible until the action is complete.
In Java,

Reads and writes are atomic for reference variables and for most primitive variables (all types except long and double).
Reads and writes are atomic for all variables declared volatile (including long and double variables).
all operations of java.concurrent.Atomic* classes
Atomic actions cannot be interleaved, so they can be used without fear of thread interference. However, this does not eliminate all need to synchronize atomic actions, because memory consistency errors are still possible.

Question 18. What Happens If A Start Method Is Not Invoked And The Run Method Is Directly Invoked?

Answer :

If a thread has been instantiated but not started its is said to be in new state. Unless until a start() method is invoked on the instance of the thread, it will not said to be alive. If you do not call a start() method on the newly created thread instance thread is not considered to be alive. If the start() method is not invoked and the run() method is directly called on the Thread instance, the code

inside the run() method will not run in a separate new thread but it will start running in the existing thread.

Question 19. What Happens When Start() Is Called?

Answer :

A new thread of execution with a new call stack starts. The state of thread changes from new to runnable. When the thread gets chance to execute its target run() method starts to run.

Question 20. If Code Running Is A Thread Creates A New Thread What Will Be The Initial Priority Of The Newly Created Thread?

Answer :

When a code running in a thread creates a new thread object, the priority of the new thread is set equal to the priority of the thread which has created it.

Question 21. What Are The Different States Of A Thread's Life Cycle?

Answer :

The different states of threads are as follows:
New – When a thread is instantiated it is in New state until the start() method is called on the thread instance. In this state the thread is not considered to be alive.
Runnable – The thread enters into this state after the start method is called in the thread instance. The thread may enter into the Runnable state from Running state. In this state the thread is considered to be alive.
Running – When the thread scheduler picks up the thread from the Runnable thread’s pool, the thread starts running and the thread is said to be in Running state.
Waiting/Blocked/Sleeping – In these states the thread is said to be alive but not runnable. The thread switches to this state because of reasons like wait method called or sleep method has been called on the running thread or thread might be waiting for some i/o resource so blocked.
Dead – When the thread finishes its execution i.e. the

run() method execution completes, it is said to be in dead state. A dead state can not be started again. If a start() method is invoked on a dead thread a runtime exception will occur.

Question 22. What Is Use Of Synchronized Keyword?

Answer :

synchronized keyword can be applied to static/non-static methods or a block of code. Only one thread at a time can access synchronized methods and if there are multiple threads trying to access the same method then other threads have to wait for the execution of method by one thread. Synchronized keyword provides a lock on the object and thus prevents race condition. E.g.

```
public void synchronized method(){}
public void synchronized staticmethod(){}
public void myMethod(){
synchronized (this){
//synchronized keyword on block of code
}
}
```

Question 23. What Is The Difference When The Synchronized Keyword Is Applied To A Static Method Or To A Non Static Method?

Answer :

When a synch non static method is called a lock is obtained on the object. When a synch static method is called a lock is obtained on the class and not on the object. The lock on the object and the lock on the class donâ€™t interfere with each other. It means, a thread accessing a synch non static method, then the other thread can access the synch static method at the same time but canâ€™t access the synch non static method.

Question 24. What Is A Volatile Keyword?

Answer :

In general each thread has its own copy of variable, such

that one thread is not concerned with the value of same variable in the other thread. But sometime this may not be the case. Consider a scenario in which the count variable is holding the number of times a method is called for a given class irrespective of any thread calling, in this case irrespective of thread access the count has to be increased so the count variable is declared as volatile. The copy of volatile variable is stored in the main memory, so every time a thread access the variable even for reading purpose the local copy is updated each time from the main memory. The volatile variable also have performance issues.

Question 25. What Is The Difference Between Yield() And Sleep()?

Answer :

yield() method pauses the currently executing thread temporarily for giving a chance to the remaining waiting threads of the same priority to execute. If there is no waiting thread or all the waiting threads have a lower priority then the same thread will continue its execution. The yielded thread when it will get the chance for execution is decided by the thread scheduler whose behavior is vendor dependent. If doesn't release the lock on the objects acquired.
sleep() allows the thread to go to sleep state for x milliseconds. When a thread goes into sleep state it doesn’t releases the lock.

Question 26. What Is The Difference Between Wait() And Sleep()?

Answer :

wait() is a method of Object class. sleep() is a method of Thread class.
sleep() allows the thread to go to sleep state for x milliseconds. When a thread goes into sleep state it doesn’t release the lock. wait() allows thread to release the lock and goes to suspended state. The thread is only active when a notify() or notifAll() method is called for the same object.
Question 27. What Is Multithreading?

Answer :

The process of executing multiple threads simultaneously is known as multithreading. Java supports multithreading. The main advantage of multithreading is reducing CPU idle time and improving the CPU utilization. This makes the job to be completed in less time.

Question 28. What Is Starvation?

Answer :

Starvation is a situation when some threads acquired the shared resources for long time and therefore other threads are not able to access those resources and not able to do anything further.
For example, suppose an object provides a synchronized method that often takes a long time to return.
If one thread invokes this method frequently, other threads that also require frequent synchronized access to that object will be blocked.
In Java, Starvation can be caused by inappropriate allocation of thread priorities.
A thread with low priority can be starved by the threads of higher priority if the higher priority threads do not release shared resources time to time.
Question 29. How Do You Debug Your Application For Issues When Multiple Threads Are Being Executed?

Answer :

Following are some way to debug issues in multi-threaded applications in Java.

By using logging and print statements along with thread names. In this way we can know about the flow of thread execution.
With the use of debugging functionality available in Eclipse and JDeveloper.
We can write a thread dump of the application which will give the information about the active threads at a point of time.
This is most effective way for detecting deadlocks in

production systems.

Question 30. What Are The Methods Of The Thread Class Used To Schedule The Threads?

Answer :

The methods of the thread class used to schedule the threads are as follows:

```
public final void join() throws InterruptedException
public final void notify()
public final void notifyAll()
public static void yield()
public final void setPriority(int priority)
public static void sleep(long millis) throws
InterruptedException
public final void wait() throws InterruptedException
```

Question 31. Explain The Method Of Runnable Interface With Example.

Answer :

In this method of creating thread, we have to implement the Runnable interface and implement the run() method in our class.
We have to create an object of our class.
Then we you have to pass the reference of that object for creating a new object of Thread
Invoke the start method using this Thread object which will create a new thread of execution.
For example

```
public class MyThread implements Runnable
{
public void run()
{
// code to execute under the thread
}
public static void main(String [] args)
{
MyThread c = new NewThread();
Thread t = new Thread(c);
t.start();
}
}
```

Question 32. What Happens If We Invoke Run Method Without Calling The Start Method For A Thread Instance?

Answer :

1.If we instantiate a thread it is called in new state until the Start() method is called.
2.If we don't call a start() method for that thread instance, the thread is not called alive.
3.If we invoke run method without calling the start method for a thread instance, the code in run() method wil not be executed by a new thread but it will be executed by the existing thread only.

Question 33. What Is Thread Leak?

Answer :

Thread leak is when application does not release references of the thread object and those threads do not get garbage collected.
Number of such unused threads increases with time and it can cause issues in the application like long response time.
To overcome this problem we can do the following
1. By maintaining a log for all entry and exit point of thread.
2. Check how the new thread is created and how it is closed.
3. By using exception handling etc.

Question 34. Explain The Method Of Thread Class With Example.

Answer :

In this method of creating thread, we have to extend the Thread class and override the run() method in our class to create a Thread.
We have to create an object of the our class.
Once the object is created then we have to invoke the start() method and it will generate a new thread of execution.
For example

```
public class MyThread extends Thread

{
public void run()
{
// code to execute under the thread
}
public static void main(String [] args)
{
MyThread c = new MyThread();
c.start();
}
}
```

Question 35. What Is The Difference Between Yielding And Sleeping?

Answer :

Sleep causes the currently executing thread to sleep until the specified time is completed. The thread will resume once the specified time period is over.
Sleep causes the currently executing thread to sleep and gives a chance to other threads to execute. The thread will join the ready queue.
Thread.sleep() will moves the thread to “Wait” state.
Thread.yield() will moves the thread to “Ready” state.
Question 36. What Is The Difference Between Pre Emptive Scheduling And Time Slicing?

Answer :

In Preemptive scheduling, highest priority task will executes until it enters in waiting or dead states. It also executes, until a higher priority task enters.
In Time slicing, a task will execute for a fixed time slice and after that it will go in ready state.
At that time the scheduler will find the executable task, according to the priority and various other tasks.
In preemptive scheduling, the running task will be preempted by the higher priority task.
In time slicing methods, a task executes until the specified period of time. Once the execution of that task is complete then the higher priority task will be executed

if available.
Question 37. Can You Tell Some Ways In Which A Thread Can Enter The Waiting State?

Answer :

A thread can enter the waiting state by the following ways:

We can invoke sleep() method of the thread.
An attempt to acquire the object’s lock can put the thread in waiting mode.
We can also invoke wait() method of the thread.
A thread can also be entered in waiting state by invoking its suspend() method.
Question 38. Name The Methods Available In The Thread Class.

Answer :

isAlive()
join()
resume()
suspend()
stop()
start()
sleep()
destroy()
Question 39. What Is Time Slicing?

Answer :

Timeslicing is the method of allocating CPU time to individual threads in a priority schedule.

Question 40. Is It Possible To Perform Stream Operations In Java 8 With A Thread Pool?

Answer :

Collections provide the method parallelStream() to create a stream that is processed by a thread pool. Alternatively you can call the intermediate method parallel() on a given stream to convert a sequential stream to a parallel counterpart.

Question 41. How Can We Access The Thread Pool That Is Used By Parallel Stream Operations?

Answer :

The thread pool used for parallel stream operations can be accessed by ForkJoinPool.commonPool(). This way we can query its level of parallelism with commonPool.getParallelism(). The level cannot be changed at runtime but it can be configured by providing the following JVM parameter: -Djava.util.concurrent.ForkJoinPool.common.parallelism=5.

Question 42. What Is The Difference Between Hashmap And Hashtable Particularly With Regard To Thread-safety?

Answer :

The methods of Hashtable are all synchronized. This is not the case for the HashMap implementation. Hence Hashtable is thread-safe whereas HashMap is not thread-safe. For single-threaded applications it is therefore more efficient to use the “newer” HashMap implementation.

Question 43. Provide An Example Why Performance Improvements For Single-threaded Applications Can Cause Performance Degradation For Multi-threaded Applications.

Answer :

A prominent example for such optimizations is a List implementation that holds the number of elements as a separate variable. This improves the performance for single-threaded applications as the size() operation does not have to iterate over all elements but can return the current number of elements directly. Within a multi-threaded application the additional counter has to be guarded by a lock as multiple concurrent threads may insert elements into the list. This additional lock can cost performance when there are more updates to the list than invocations of the size() operation.

Question 44. What Is The Purpose Of The Class

Java.lang.threadlocal?

Answer :

As memory is shared between different threads, ThreadLocal provides a way to store and retrieve values for each thread separately. Implementations of ThreadLocal store and retrieve the values for each thread independently such that when thread A stores the value A1 and thread B stores the value B1 in the same instance of ThreadLocal, thread A later on retrieves value A1 from this ThreadLocal instance and thread B retrieves value B1.

Question 45. What Are Possible Use Cases For Java.lang.threadlocal?

Answer :

Instances of ThreadLocal can be used to transport information throughout the application without the need to pass this from method to method. Examples would be the transportation of security/login information within an instance of ThreadLocal such that it is accessible by each method. Another use case would be to transport transaction information or in general objects that should be accessible in all methods without passing them from method to method.

****
****

****
****

Hi Friends, Yesterday I have attended Uber client interview(Telephonic round), Interview has happened around one hour. I would like to share my interview experience with you people, I didn't remember all question and not written in proper order, whatever questions are striking in to my mind, I wrote in that order.

1)Difference between === and euqlas() ?
2)String s1="abc";
String s2=new String("abc");
i)what is the result
s1==s2;
s.equals(s2);
ii)s1==s2; I want to get the result as true then what should I add?
3)Difference between Runnable and Externalizable interface?
4)What is the difference between HashMap and HashTable?
5)What is the difference between Lock and synchronization?
6)How you are consuming Rest API's?
7)What is the difference between @RequestBody and @RequestParam?
8)What is the difference between Unused code and Unreachable code?
9)Can we lock a class using Synchronization?
10)How can we store and retrieve null as key in HashMap?
11)Give me a quick introduction about your self?
12)What is the difference between @Configuration and @Component?
13)What is the difference between @Autowire and @Resource?
14)If I call getBean(,) method twice with the same bean then how many objects will be created?
15)Did u work on spring Security?
16)How many days u will take to implement one application?
17)How your producing response and consuming request from third party API?
18)Difference between checked and unchecked exception, Give some examples?
19)What is the difference between sleep() and wait() method?
20)Did u use JMS in your project and how you have implemented?
21)How you are validating a user, which algorithm u have used?
22)Explain different types of HTTP methods?
23)What is the difference between POST and PUT?
24)What is the difference between GET and POST?
25)What is the use of PATCH method?
26)Can I update a record using POST method?
27)How u will validate request data?
28)Total how many years of experience u have?
29)Give one example for Synchronization concept and where u

have used in your project?
30)What is the difference between @Controller and @Service?
31)What is the use of @Component?
32)What is the use of @RequestMapping, Can we apply on class level?
33)What is the difference between XML and JSON, Why JSON is faster than XML?
34)What is the difference between @RestController and @Controller?
35)Which ORM tool you have used in your project?
36)Do you have an idea on JPA, Hibernate?
37)What is spring MVC?
38)Did u implement procedures and triggers in your project?
39)How many databases do you know?
40)What is the use of clusters and Joins?
41)How much rating you will give for yourself out of 10(Java, Spring, Database, SpringBoot, Microservices..etc)
42)What is the Contract between hashcode() and equals() method?
43)If two objects are equals then their hashcode also same?
44)If two objects are having the same hashcode then their state also same?
45)How your generating security token in your project?
46)What is stream in java 8?
47)What is lambda expression?
48)Did you work on Java 8?
49)How you will iterate the Hashmap and tell me the syntax?
50)if you call next() method on iterator object then what it will return?
51)What is the difference between Comparator and Comparable interface?
52)Explain System.out.println()?

****

My Interviews :

****

Tech Mahendra (28/11/2018)

1) how to differentiate springboot & Spring MVC ?

2) how to reload spring boot application any configuration required ?

3) in spring boot what is use of devtools ?
When you have the spring-boot-devtools module included, any classpath file changes will automatically trigger an application restart.

4) difference b/w @controller & @RestController ?
Ans: The @RestController annotation in Spring MVC is nothing but a combination of @Controller and @ResponseBody annotation. It was added into Spring 4.0 to make the development of RESTful Web Services in Spring framework easier.

This difference is also obvious in the @Controller and @RestController annotation. The job of @Controller is to create a Map of model object and find a view but @RestController simply return the object and object data is directly written into HTTP response as JSON or XML.

This can also be done with traditional @Controller and use @ResponseBody annotation but since this is the default behavior of RESTful Web services, Spring introduced @RestController which combined the behavior of @Controller and @ResponseBody together.

5)in spring what are the inner beans are there?
When wiring beans, if a bean element is embedded to a property tag directly, then that bean is said to the Inner Bean. The drawback of this bean is that it cannot be reused anywhere else.

6) in spring what are the scopes are there ?.

Ans: request, session, global, singleton, prototype.

7) What Are The Different Types Of Bean Injections?
Answer :

There are two types of bean injections.
1. By setter
2. By constructor

What’s the difference between @Component, @Controller, @Repository & @Service annotations in Spring?

@Component: This marks a java class as a bean. It is a generic stereotype for any Spring-managed component. The component-scanning mechanism of spring now can pick it up and pull it into the application context.

@Controller: This marks a class as a Spring Web MVC controller. Beans marked with it are automatically imported into the Dependency Injection container.

@Service: This annotation is a specialization of the component annotation. It doesn’t provide any additional behavior over the @Component annotation. You can use @Service over @Component in service-layer classes as it specifies intent in a better way.

@Repository: This annotation is a specialization of the @Component annotation with similar use and functionality. It provides additional benefits specifically for DAOs. It imports the DAOs into the DI container and makes the unchecked exceptions eligible for translation into Spring DataAccessException.

****
****

CTS:

1. how to create a custom annotation in java?

2. what is static synchronization ?

3. what is Asynch ?

sentech media:

****
1.write a program string repeated words count & display?

RestTemplet consuming steps:
****

```
@SpringBootApplication
@EnableAutoConfiguration(exclude =
{EmbeddedServletContainerAutoConfiguration.class,
WebMvcAutoConfiguration.class})
public class SpringbootResttemplateApplication {
    public static void main(String[] args) {
        RestfulClient restfulClient = new RestfulClient();
        /*
         *POST ENTITY
         */
        restfulClient.postEntity();

        /*
         * GET ENTITY
         */
        restfulClient.getEntity();

        /*
         * PUT ENTITY
         */
        restfulClient.putEntity();

        /*
         * DELETE ENTITY
         */
        restfulClient.deleteEntity();
    }
}
```