

# Top 22 Java Design Patterns Interview Questions & Answers

## 1) Explain what is Java Design Pattern?

A design pattern is a language independent strategies for solving common object oriented design problem. It describes how to structure classes to meet a given requirement.

## 2) Explain what is creational design patterns and Factory pattern?

**Creational design pattern:** This pattern is used to define and describe how objects are created at class instantiation time.

**Factory pattern:** The factory pattern is used to create an object without exposing the creation logic to the client and refer to a newly created object using a common interface.

## 3) Which design pattern is used to get a way to access the elements of a collection object in sequential manner?

Iterator pattern is used to get a way to access the elements of a collection object in sequential manner.

## 4) When service locator pattern is used?

When we want to locate various services using JNDI we use service locator pattern.

## 5) Mention in how many ways can you create singleton pattern?

To create single objects there are two famous ways

- Lazy loading
- Eager loading



## **6) Mention which pattern is used when we need to decouple an abstraction from its implementation?**

When we want to decouple an abstraction from its implementation in order that two can vary independently we use bridge pattern.

## **7) Mention which design pattern will be helpful to add new functionality to an existing object?**

A decorator pattern allows a user to add new functionality to an existing object without changing its structure.

## **8) Explain how can you create a Singleton class in Java?**

It is two step process,

- First make the constructor private so that new operator cannot be used to instantiate the class
- Return an object of the object if not null otherwise create the object and return the same via a method.

## **9) Is it possible to write thread safe singleton in Java?**

To write thread safe singleton in Java there are multiple ways for example by using static singleton instance initialized during class loading, by writing singleton using double checked locking. Java Enum is the simplest way to create thread safe singleton.

## **10) Mention how one should describe a design pattern?**

To describe a design pattern, following things need to be taken care of

- Pattern name and classification
- Problem and solution
- Consequences : Variation and language dependent alternatives should also be addressed
- Know Uses: Identify the uses in the real systems and its efficiency

## **11) Mention why access to the non-static variable is not allowed from static method in Java?**

You cannot access non-static data from static context because non-static variable are associated with a specific instance of an object while static is not associated with any instance.

## **12) Mention which pattern is useful when one has to pass data with multiple attributes in one shot from client to server?**

Transfer Object Pattern is useful when one has to pass data with multiple attributes in one shot from client to the server.

## **13) Name some of the entities of DAO pattern?**

Some of the entities of DAO include,

- Data access object concrete class
- Data access object interface
- Model object or value object

#### **14) Mention when can you use the Intercepting pattern?**

Intercepting pattern is used when you have to do some pre-processing or post processing with request or response of the application.

#### **15) Mention when to use a Factory Pattern?**

Factory pattern can be used,

- When a class does not know which class of objects needs to create
- When class specifies its sub-classes to specify which objects to create
- In programming language, you can use factory pattern where you have to create an object of any one of sub-classes depending on the given data

#### **16) Explain in singleton pattern whether it is better to make the whole getInstance() method synchronized or just critical section is enough? Which one is preferable?**

Synchronization of whole getInstance() method is costly and is only needed during the initialization on singleton instance, to stop creating another instance of Singleton. Therefore it is better to only synchronize critical section and not the whole method.

#### **17) Mention in how many ways can you write singleton class in Java?**

One can write singleton class in Java in four ways

- Singleton with public static final field initialized during class loading
- Singleton generated by static nested class, also referred as singleton holder pattern
- Singleton by synchronizing get instance () method
- From Java 5 on-wards using Enums

#### **18) Explain how can you prevent creating another instance of singleton using clone() method?**

The preferred way to prevent creating another instance of a singleton is by not implementing Cloneable interface and if you do just throw an exception from clone() method “ not to create a clone of singleton class”.

#### **19) Mention what is the difference between “throw” and “throws”?**

Keyword “Throw” is used to explicitly throw as an exception, while “Throws” is utilized to handle checked exceptions for re-intimating the compiler that exceptions are being handled. The throws need to be used in the method’s definition and also while invoking the method that raises checked exceptions.

#### **20) Mention which classes in JDK uses singleton pattern?**

Java.lang.Runtime classes uses singleton pattern in JDK.

#### **21) Mention what is the limitation of using singleton pattern?**

The singleton pattern ensures that a class has only one instance and to provide a global point of access to it. But at the same time this becomes its limitation as most classes in an application you will need to create multiple instances.

#### **22) Mention what is the difference between VO and JDO?**

The difference between JDO and VO is that the JDO is a persistent technology that compete against entity beans in enterprise application development. It enables you to create POJO (plain old java objects) and persist them to the database.

While VO stands for value objects represents an abstract design pattern used in conjunction with entity beans, jdbc and possibly even JDO to overcome commonly found isolation and transactional problems in enterprise apps.

[Guru99](#) Provides [FREE ONLINE TUTORIAL](#) on Various courses like

Java	MIS	MongoDB	BigData	Cassandra
------	-----	---------	---------	-----------

Web Services	SQLite	JSP	Informatica	Accounting
--------------	--------	-----	-------------	------------

SAP Training	Python	Excel	ASP Net	HBase
--------------	--------	-------	---------	-------

Project Management	Test Management	Business Analyst	Ethical Hacking	PMP
--------------------	-----------------	------------------	-----------------	-----

Live Project	SoapUI	Photoshop	Manual Testing	Mobile Testing
--------------	--------	-----------	----------------	----------------

Data Warehouse	R Tutorial	Tableau	DevOps	AWS
----------------	------------	---------	--------	-----

Jenkins	Agile Testing	RPA	JUnit	Software Engineering
---------	---------------	-----	-------	----------------------

Selenium	CCNA	AngularJS	NodeJS	PLSQL
----------	------	-----------	--------	-------

**Stay updated with new  
courses at Guru99  
Join our Newsletter**