# Java

**- Why is Java called the ‘Platform Independent Programming Language’?**

Platform independence means that execution of your program does not dependent on type of operating system(it could be any : Linux, windows, Mac ..etc)

So compile code only once and run it on any System

- **Explain Final keyword in java?**

Final keyword in java is used to restrict usage of variable, class and method.

Variable: Value of Final variable is constant, you can not change it.

Method: you can’t override a Final method.

Class: you can’t inherit from Final class.

- **When is the super keyword used?**

super keyword is used to refer:

immediate parent class constructor,

immediate parent class variable,

immediate parent class method.

- **What is the difference between StringBuffer and String?**

String is an Immutable class, i.e. you can not modify its content once created.

While StringBuffer is a mutable class, means you can change its content later.

Whenever we alter content of String object, it creates a new string and refer to that,it does not modify the existing one.

This is the reason that the performance with StringBuffer is better than with String.

**Why multiple inheritance is not supported in java?**

Java supports multiple inheritance but not through classes, it supports only through its interfaces.

The reason for not supporting multiple inheritance is to avoid the conflict and complexity arises due to it and keep Java a Simple Object Oriented Language

**Can a top level class be private or protected?**

Top level classes in java can’t be private or protected, but inner classes in java can.

The reason for not making a top level class as private is very obvious, because nobody can see a private class and thus they can not use it.

Declaring a class as protected also doesn’t make any sense

**What is the difference between ‘throw’ and ‘throws’ in Java Exception Handling?**

throw keyword is used to throw Exception

throws is used to indicate that which Exception can possibly be thrown by this method

**Difference in Set and List interface?**

Set and List both are child interface of Collection interface. There are following two main differences between them

* List can hold duplicate values but Set doesn’t allow this.
* In List interface data is present in the order you inserted but in the case of Set insertion order is not preserved.

**Can we**[**Overload or Override static methods in java**](https://www.geeksforgeeks.org/can-we-overload-or-override-static-methods-in-java/)**?**

***Overriding* :** Overriding is related to run-time polymorphism.  A subclass (or derived class) provides a specific implementation of a method in superclass (or base class) at runtime.

***Overloading*:** Overloading is related to compile time (or static) polymorphism. This feature allows different methods to have same name, but different signatures, especially number of input parameters and type of input paramaters.

**Can we overload static methods?**

The answer is **‘Yes’**. We can have two ore more static methods with same name, but differences in input parameters

**Can we Override static methods in java?**

We can declare static methods with same signature in subclass, but it is not considered overriding as there won’t be any run-time polymorphism. Hence the answer is **‘No’**

**What happens if you remove static modifier from the main method?**

Program compiles successfully . But at runtime throws an error “NoSuchMethodError”.

**What is the**[**scope of variables**](https://www.geeksforgeeks.org/variable-scope-in-java/)**in Java in following cases?**

* *Member Variables* (Class Level Scope) : The member variables must be declared inside class (outside any function). They can be directly accessed anywhere in class
* *Local Variables*(Method Level Scope) : Variables declared inside a method have method level scope and can’t be accessed outside the method.
* *Loop Variables*(Block Scope) : A variable declared inside pair of brackets “{” and “}” in a method has scope withing the brackets only.

**What is an**[**abstract class**](https://www.geeksforgeeks.org/abstract-classes-in-java/)**? How abstract classes are similar or different in Java from C++?**

Abstract classes are classes that contain one or more abstract methods.

An abstract method is a method that is declared, but contains no implementation.

Abstract classes may not be instantiated, and require subclasses to provide implementations for the abstract methods.

* Like C++, in Java, an instance of an abstract class cannot be created, we can have references of abstract class type though.
* Like C++, an abstract class can contain constructors in Java. And a constructor of abstract class is called when an instance of a inherited class is created
* In Java, we can have an abstract class without any abstract method. This allows us to create classes that cannot be instantiated, but can only be inherited.
* Abstract classes can also have final methods (methods that cannot be overridden). For example, the following program compiles and runs fine.

**Can we override private methods in Java?**

No, a private method cannot be overridden since it is not visible from any other class.

**What is**[**blank final variable**](https://www.geeksforgeeks.org/blank-final-in-java/)**?**

A **blank final** variable in Java is a [final](https://www.geeksforgeeks.org/g-fact-48/) variable that is not initialized during declaration. Below is a simple example of blank final.

A final variable in Java can be assigned a value only once, we can assign a value either in declaration or later.

**What is**[**static variable in Java**](https://www.geeksforgeeks.org/static-class-in-java/)**?**

The static keyword in java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class than instance of the class.

The static can be:

* variable (also known as class variable)
* method (also known as class method)
* block
* nested class

**Differences between**[**HashMap and HashTable in Java**](http://quiz.geeksforgeeks.org/differences-between-hashmap-and-hashtable-in-java/)**.**

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

**Garbage collection**

* Garbage Collector is a [Daemon thread](https://www.journaldev.com/1072/daemon-thread-in-java) that keeps running in the background. Basically, it frees up the [heap memory](https://www.journaldev.com/4098/java-heap-space-vs-stack-memory) by destroying the unreachable objects.
* Unreachable objects are the ones that are no longer referenced by any part of the program.

What is structure of Java Heap ? What is Perm Gen space in Heap ?

Heap is divided into different generation e.g. new generation, old generation and PermGen space.PermGen space is used to store class’s metadata and filling of PermGen space can cause [java.lang.OutOfMemory:PermGen space](http://javarevisited.blogspot.sg/2012/01/tomcat-javalangoutofmemoryerror-permgen.html). Its also worth noting to remember [JVM option to configure PermGen](http://javarevisited.blogspot.sg/2011/09/javalangoutofmemoryerror-permgen-space.html) space in Java.

Data Encapsulate

* Technically in encapsulation, the variables or data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared.
* As in encapsulation, the data in a class is hidden from other classes, so it is also known as **data-hiding**.
* Encapsulation can be achieved by: Declaring all the variables in the class as private and writing public methods in the class to set and get the values of variables.

The **volatile** keyword basically says to the JVM “Warning, this variable may be modified in another Thread”.

**How is**[**inheritance in C++  different from Java?**](https://www.geeksforgeeks.org/comparison-of-inheritance-in-c-and-java/)

1. In Java, all classes inherit from the Object class directly or indirectly. Therefore, there is always a single inheritance tree of classes in Java, and Object class is root of the tree.
2. In Java, members of the grandparent class are not directly accessible. See [this G-Fact](https://www.geeksforgeeks.org/accessing-grandparents-member-in-java-using-super/)for more details.
3. Java uses extends keyword for inheritance. Unlike C++, Java doesn’t provide an inheritance specifier like public, protected or private. Unlike C++, in Java, we don’t have to remember those rules of inheritance which are combination of base class access specifier and inheritance specifier.
4. In Java, methods are virtual by default. In C++, we explicitly use virtual keyword. See [this G-Fact](https://www.geeksforgeeks.org/g-fact-43/) for more details.

**Difference between Abstract Class and Interface in Java**

* ***Type of methods*: Interface can have only abstract methods. Abstract class can have abstract and non-abstract methods. From Java 8, it can have default and static methods also.**
* ***Final Variables*: Variables declared in a Java interface are by default final. An abstract class may contain non-final variables.**
* ***Type of variables*: Abstract class can have final, non-final, static and non-static variables. Interface has only static and final variables.**
* ***Implementation*: Abstract class can provide the implementation of interface. Interface can’t provide the implementation of abstract class.**
* ***Inheritance vs Abstraction*: A Java interface can be implemented using keyword “implements” and abstract class can be extended using keyword “extends”.**
* ***Multiple implementation*: An interface can extend another Java interface only, an abstract class can extend another Java class and implement multiple Java interfaces.**

***Accessibility of Data Members*: Members of a Java interface are public by default. A Java abstract class can have class members like private, protected, etc.**

# Design patterns

Design pattern is a general reusable solution or template to a commonly occurring problem in software design

SINGLETON:

Singleton pattern is a creational pattern which allows only one instance of a class to be created which will be available to the whole application. The major advantage of Singleton design pattern is its saves memory because the single instance is reused again and again; there is no need to create a new object at each request. For example, in our application, we can use a single database connection shared by multiple objects, instead of creating a [database](https://www.educba.com/course/advanced-database-designer-4/) connection for every request.

Drawbacks:

a)Singleton causes code to be tightly coupled. The singleton object is exposed globally and is available to a whole application. Thus, classes using this object become tightly coupled; any change in the global object will impact all other classes using it.  
b)They hide dependencies instead of exposing them.  
c)Singleton Pattern does not support inheritance.

FACTORY:  this pattern provides one of the best ways to create an object In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a common interface.

Benefit: Factory pattern encapsulates the implementation details and underlying implementation can be changed without any impact on caller api.

ADAPTER : The Adapter Design Pattern is part of structural patterns Adapter pattern works as a bridge between two incompatible interfaces. This pattern involves a single class which is responsible to join functionalities of independent or incompatible interfaces.

PROXY : Proxy pattern is used when we need to create a wrapper to cover the main object’s complexity from the client.

Benefit:

One of the advantages of Proxy pattern is security.

This pattern avoids duplication of objects which might be huge size and memory intensive. This in turn increases the performance of the application.

Drawback:

This pattern introduces another layer of abstraction which sometimes may be an issue if the RealSubject code is accessed by some of the clients directly and some of them might access the Proxy classes. This might cause disparate behaviour.

Introduction:

First of all, Thank you very much for taking your time and giving me this opportunity. My name is Quan. I'm from VietNam. I am a software engineer with nearly 7 years experience in software development. My favorite programming language is Java. I am responsible for coding, analyzing and implementing base on the requirements. Although, I’m happy with my current job, but I feel now I’m ready for new challenges & assignments. This position really excites me.

1. Discuss one of your previous projects and explain how you completed it successfully

The closest project I am going to tell you that is Web Release Database project. It is a web application to manage all belts information from 3 factories ( HcP, Tbp, and SLPP) on central location and export reports. I working as a full stack developer in this project, from backend to frontend. In the backend, we use Java, Spring boot, Spring JPA. In Frontend, we use angular 5 and Bootstrap 4. Additional, we also follow agile methodologies for this project. Especially Scrum Methodology.

<https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>

another project is also a web application to manage the account. the main feature is to: manage the public account, whitelist, and dialog.

To complete the project successfully we always follow the rules. Firstly, when we process the task if we have any doubt or any concern, we will not hesitate to clarify them and when we have a difficult issue, we will support together.

when coding we adhere to some criteria:

* + Reuse
  + Consistency
  + Transparent
  + Maintenance
  + performance

1. Did you run into any obstacles with your project and how did you handle the issue?

If I have any obstacles in my task. Firstly, I will find the root cause, and search the solution in the internet. If there are no result, I will ask some help from my colleague. The last solution, I will discuss with team get the advice from them and give another direction.

It was about two weeks ago I support to fix a performance bug. I remember the root cause is three nested loops in a method. It is quite difficult to change the logic handling. It take me nearly a half-day to find the solution for this. instead of using I use the Hashmap and separate the loops.

1. Do you follow the Agile software development process?
2. What do you know about multi-threading?
   1. Multi-threading is a way to improve the performance of an application.
   2. Multi-threading is a process that uses several processors to run snippets of code.
   3. It improves performance and stops code from "hanging" when it runs.
3. What are your thoughts on unit testing? (Junit)

Unit testing and test driven development (TDD) are often regarded as best practices in software development and code maintenance. Unit testing is an extra set of code that tests various methods and procedures for logic errors and coding flaws. This helps eliminate many of the bugs that could be promoted to production.

1. What is MVC?

MVC stands for Model-View-Controller. It's a way to separate code into its own compartments, typicially like so:

* 1. Model is the data layer that manages business logic and maintains data
  2. View is the front-end presentation layer, or UI layer
  3. Controller is the logic layer that handles user requests and behavior, and updates the Model

1. How familiar are you with OOP analysis and design?

Object-oriented programming (OOP) is the core of major languages such as VB, Java, C++, and C#, so top software developers should have strong OOP skills. OOP is a process of using classes to describe components of a program. For instance, if your program explains a house, classes would be the roof, doors, windows, and rooms

1. How do you organize your class modules and assets?

There is no one "right" way to answer this question, but your team probably has a set standard and it helps to know if the developer organizes code in a way that is easily maintained and can be further documented.

1. What do you know about ORM?

Object-relational mapping (ORM) is a way to use software code so it can map to database tables. This technique turns tables into their own classes, so then developers can use those classes for LINQ queries.

1. What is your process for finding a bug in an application? How much time do you typically spend on debugging?

Here is some my thoughts of how to "finding" **performance problems**:

For a "new" api/application or other

* Analzying the detail api and then preparing the Jmeter/Grinder testing scripts for it.
* Using different load to identify the threshold for the api
* Adding profiling codes find the slownes
* Restart from point one again..

For a "old" api/application or other

* Analyzing the user pattern from the access detail log
* Simulate the real user load to find the slowness
* Adding profiling codes find the slownes
* Restart from point one again..

# Rest API

<https://www.javacodegeeks.com/2018/02/top-20-spring-rest-interview-questions-answers-java-programmers.html>

<https://www.javacodegeeks.com/2014/05/spring-interview-questions-and-answers.html>

<https://www.javacodegeeks.com/2018/08/hibernate-interview-questions-answers-ultimate-list.html>

# REST is web standards based architecture and uses HTTP Protocol for data communication

Common method of Rest

* **GET:**It requests a resource at the request URL. It should not contain a request body as it will be discarded. Maybe it can be cached locally or on the server.
* **POST:**It submits information to the service for processing; it should typically return the modified or new resource
* **PUT:**At the request URL it update the resource
* **DELETE:**At the request URL it removes the resource
* **OPTIONS:**It indicates which techniques are supported
* **HEAD:**About the request URL it returns meta information

# JAXB stands for java arch for XML binding.

What is the difference PUT and POST

"PUT" puts a file or resource at a particular URI and exactly at that URI. If there is already a file or resource at that URI, PUT changes that file or resource. If there is no resource or file there, PUT makes one

POST sends data to a particular URI and expects the resource at that URI to deal with the request. The web server at this point can decide what to do with the data in the context of specified resource

PUT is idempotent meaning, invoking it any number of times will not have an impact on resources.

However, POST is not idempotent, meaning if you invoke POST multiple times it keeps creating more resources

**What is the difference between @Controller and @RestController?**([answer](http://javarevisited.blogspot.sg/2017/08/difference-between-restcontroller-and-controller-annotations-spring-mvc-rest.html))

There are many differences between  @Controller and @RestController as discussed in my earlier article (see the answer) but the most important one is that with  @RestController you get the  @ResponseBody annotation automatically, which means you don’t need to separately annotate your handler methods with  @ResponseBody annotation. This makes the development of RESTful web service easier using Spring. You can see here to learn

# 

I would accept a competitive salary that best fits the industry standards

Although, I’m happy with my current job, but I feel now I’m ready for new challenges & assignments. This position really excites me.