Java Interview Question

Q-1 Explain public static void main(String args[]).

- **public** : Public is an access modifier, which is used to specify who can access this method. Public means that this Method will be accessible by any Class.
- **static** : It is a keyword in java which identifies it is class based i.e it can be accessed without creating the instance of a Class.
- **void** : It is the return type of the method. Void defines the method which will not return any value.
- **main**: It is the name of the method which is searched by JVM as a starting point for an application with a particular signature only. It is the method where the main execution occurs.
- **String args[]** : It is the parameter passed to the main method.

Q-2. Why Java is platform independent?

Platform independent practically means "write once run anywhere". Java is called so because of its byte codes which can run on any system irrespective of its underlying operating system.

Q-3. Why java is not 100% Object-oriented?

Java is not 100% Object-oriented because it makes use of eight primitive datatypes such as boolean, byte, char, int, float, double, long, short which are **not objects.**

Q-4. What do you mean by Object?

An object consists of methods and class which depict its state and perform operations. A java program contains a lot of objects instructing each other their jobs. This concept is a part of core java.

Q-5 What is association?

Association is a relationship where all object have their own lifecycle and there is no owner. Let's take an example of Teacher and Student. Multiple students can associate with a single teacher and a single student can associate with multiple teachers but there is no ownership between the objects and both have their own

lifecycle. These relationship can be one to one, One to many, many to one and many to many.

Q-6. What do you mean by aggregation?

Aggregation is a specialized form of Association where all object have their own lifecycle but there is ownership and child object can not belongs to another parent object. Let's take an example of Department and teacher. A single teacher can not belongs to multiple departments, but if we delete the department teacher object will not destroy.

Q-7. What is composition in Java?

Composition is again specialized form of Aggregation and we can call this as a "death" relationship. It is a strong type of Aggregation. Child object dose not have their lifecycle and if parent object deletes all child object will also be deleted. Let's take again an example of relationship between House and rooms. House can contain multiple rooms there is no independent life of room and any room can not belongs to two different house if we delete the house room will automatically delete.

Q-8. What's the difference between an array and Vector?

Ans: An array groups data of same primitive type and is static in nature while vectors are dynamic in nature and can hold data of different data types.

Q-9. What is multi-threading?

Ans: Multi threading is a programming concept to run multiple tasks in a concurrent manner within a single program. Threads share same process stack and running in parallel. It helps in performance improvement of any program.

Q-10. Why Runnable Interface is used in Java?

Ans: Runnable interface is used in java for implementing multi threaded applications. Java.Lang.Runnable interface is implemented by a class to support multi threading.

Q-11. What are the two ways of implementing multi-threading in Java?

Ans: Multi threaded applications can be developed in Java by using any of the following two methodologies:

1. By using Java.Lang.Runnable Interface. Classes implement this interface to enable multi threading. There is a Run() method in this interface which is implemented.

2. By writing a class that extend Java.Lang.Thread class.

Q-12. When a lot of changes are required in data, which one should be a preference to be used? String or StringBuffer?

Ans: Since String Buffers are dynamic in nature and we can change the values of StringBuffer objects unlike String which is immutable, it's always a good choice to use StringBuffer when data is being changed too much. If we use String in such a case, for every data change a new String object will be created which will be an extra overhead.

Q-13. What's the purpose of using Break in each case of Switch Statement?

Ans: Break is used after each case (except the last one) in a switch so that code breaks after the valid case and doesn't flow in the proceeding cases too.

If break isn't used after each case, all cases after the valid case also get executed resulting in wrong results.

Q-14. How garbage collection is done in Java?

Ans: In java, when an object is not referenced any more, garbage collection takes place and the object is destroyed automatically. For automatic garbage collection java calls either System.gc() method or Runtime.gc() method.

Q-15. How we can execute any code even before main method?

Ans: If we want to execute any statements before even creation of objects at load time of class, we can use a static block of code in the class. Any statements

inside this static block of code will get executed once at the time of loading the class even before creation of objects in the main method.

Q-16 What's the benefit of using inheritance?

Ans: Key benefit of using inheritance is reusability of code as inheritance enables sub-classes to reuse the code of its super class. Polymorphism (Extensibility) is another great benefit which allow new functionality to be introduced without effecting existing derived classes.

Q-17. What's the default access specifier for variables and methods of a class?

Ans: Default access specifier for variables and method is package protected i.e variables and class is available to any other class but in the same package,not outside the package.

Q-18. Give an example of use of Pointers in Java class.

Ans: There are no pointers in Java. So we can't use concept of pointers in Java.

Q-19. What are the steps to connect to a database in java?

Ans-

- Registering the driver class
- Creating connection
- Creating statement
- Executing queries
- Closing connection

Q-20. What are the JDBC API components?

The java.sql package contains interfaces and classes for JDBC API.

Interfaces:

- Connection
- Statement
- PreparedStatement
- ResultSet
- ResultSetMetaData
- DatabaseMetaData
- CallableStatement etc.

Classes:

- DriverManager
- Blob
- Clob
- Types
- SQLException etc.

Q-21. What is the role of JDBC DriverManager class?

Ans- The DriverManager *class* manages the registered drivers. It can be used to register and unregister drivers. It provides factory method that returns the instance of Connection.

Q-22. What is Hibernate Framework?

Ans- Object-relational mapping or ORM is the programming technique to map application domain model objects to the relational database tables. Hibernate is java based ORM tool that provides framework for mapping application domain objects to the relational database tables and vice versa.

Hibernate provides reference implementation of Java Persistence API, that makes it a great choice as ORM tool with benefits of loose coupling. We can use Hibernate persistence API for CRUD operations. Hibernate framework provide option to map plain old java objects to traditional database tables with the use of JPA annotations as well as XML based configuration.

Similarly hibernate configurations are flexible and can be done from XML configuration file as well as programmatically.

Q-23. What are the important benefits of using Hibernate Framework?

Ans- Some of the important benefits of using hibernate framework are:

1. Hibernate eliminates all the boiler-plate code that comes with JDBC and takes care of managing resources, so we can focus on business logic.
2. Hibernate framework provides support for XML as well as JPA annotations, that makes our code implementation independent.
3. Hibernate provides a powerful query language (HQL) that is similar to SQL. However, HQL is fully object-oriented and understands concepts like inheritance, polymorphism and association.
4. Hibernate is an open source project from Red Hat Community and used worldwide. This makes it a better choice than others because learning curve is

small and there are tons of online documentations and help is easily available in forums.
5. Hibernate is easy to integrate with other Java EE frameworks, it's so popular that Spring Framework provides built-in support for integrating hibernate with Spring applications.
6. Hibernate supports lazy initialization using proxy objects and perform actual database queries only when it's required.
7. Hibernate cache helps us in getting better performance.
8. For database vendor specific feature, hibernate is suitable because we can also execute native sql queries.

Overall hibernate is the best choice in current market for ORM tool, it contains all the features that you will ever need in an ORM tool.

Q-24. Is it possible to define a method in Java class but provide it's implementation in the code of another language like C?

Ans: Yes, we can do this by use of native methods. In case of native method based development, we define public static methods in our Java class without its implementation and then implementation is done in another language like C separately.

Q-25. How are destructors defined in Java?

Ans: In Java, there are no destructors defined in the class as there is no need to do so. Java has its own garbage collection mechanism which does the job automatically by destroying the objects when no longer referenced.

Q-26. Can a variable be local and static at the same time?

Ans: No a variable can't be static as well as local at the same time. Defining a local variable as static gives compilation error.

Q-27. Can we have static methods in an Interface?

Ans: Static methods can't be overridden in any class while any methods in an interface are by default abstract and are supposed to be implemented in the classes being implementing the interface. So it makes no sense to have static methods in an interface in Java.

Q-28. In a class implementing an interface, can we change the value of any variable defined in the interface?

Ans: No, we can't change the value of any variable of an interface in the implementing class as all variables defined in the interface are by default public,

static and Final and final variables are like constants which can't be changed later.

Q-29 Is it correct to say that due to garbage collection feature in Java, a java program never goes out of memory?
Ans: Even though automatic garbage collection is provided by Java, it doesn't ensure that a Java program will not go out of memory as there is a possibility that creation of Java objects is being done at a faster pace compared to garbage collection resulting in filling of all the available memory resources.
So, garbage collection helps in reducing the chances of a program going out of memory but it doesn't ensure that.

Q-30. Can we have any other return type than void for main method?
Ans: No, Java class main method can have only void return type for the program to get successfully executed.
Nonetheless , if you absolutely must return a value to at the completion of main method , you can use System.exit(int status).

Q-31. I want to re-reach and use an object once it has been garbage collected. How it's possible?
Ans: Once an object has been destroyed by garbage collector, it no longer exists on the heap and it can't be accessed again. There is no way to reference it again.

Q-32. In Java thread programming, which method is a must implementation for all threads?
Ans: Run() is a method of Runnable interface that must be implemented by all threads.

Q-33. I want to control database connections in my program and want that only one thread should be able to make database connection at a time. How can I implement this logic?
Ans: This can be implemented by use of the concept of synchronization.
Database related code can be placed in a method which hs **synchronized** keyword so that only one thread can access it at a time.

Q-34 What are real-world practices of volatile modifier?

Ans-One of the real-world use of the volatile variable is to generate interpretation double and long atomic. Equally double and long are 64-bit extensive and they are reciting in two parts, primary 32-bit first time and following 32-bit another time, which is non-atomic but then again volatile double in addition long read is atomic in Java. Additional use of the volatile variable is to deliver a recall barrier, just like it is cast-off in Disrupter framework.

Fundamentally, Java Memory model pull-outs a write barrier subsequently you write to a volatile variable besides a read barrier beforehand you read it. That means, if you inscribe to volatile field then it's definite that any thread retrieving that variable will see the worth you wrote and everything you did beforehand doing that correct into the thread is certain to have occurred and any rationalized data values will also be noticeable to all threads, since the memory barrier flushed all additional writes to the cache.

Q-35. Explain the meaning of thread local variable in Java?

Answer- Thread-local variables are variables limited to a thread, it's like thread's individual copy which is not public between numerous threads. Java offers a Thread Local class to care thread-local variables. It's one of the numerous ways to attain thread-safety. Though be cautious while using thread locally adjustable in managed environment e.g. with network servers where operative thread out lives somewhat application variable. Somewhat thread local variable which is not detached once its work is done can possibly reason a memory leak in Java application.

Q-36. Explain the meaning of platform?

Answer- A platform is the hardware or software setting in which a program executes. Maximum platforms in JAVA can be defined as a grouping of the operating system and hardware, Windows 2000/XP, Linux, Windows 2000/XP, MacOs and Solaris.

Q-37 What's the variance amid an Abstract Class and Interface in Java?

Answer- The main difference amid an abstract class and interface is that a boundary cans only own assertion of public static approaches with no existing application while an abstract class can have associated with any admittance specifies (i.e. public, private etc.) with or without real implementation. Additional key variance in the usage of abstract classes and lines is that a class which gears an interface must contrivance all the approaches of the interface while a class which receives from an abstract class doesn't need execution of all the methods of its superclass. A class can instrument numerous interfaces but it can spread only one abstract class.

Q-38 What is the variance amid an Inner Class and a Sub-Class?

Answer An Inner class is a class which is copied up to the additional class. An Inner class has admittance privileges for the class which is nesting it and it can contact all variables and system well-defined in the outer class, whereas a sub-class is a class which receives from another class named Superclass. Sub-class can contact all public and protected approaches and fields of its superclass.

Q-39 Can we affirm a class as Abstract deprived of having an abstract method?

Answer- Yes, we can generate an abstract class via abstract keyword beforehand class name even if it doesn't have any abstract method. Though, if a class has even one abstract technique, it must be acknowledged as abstract or else it will give an error.

40- What is difference between JDK,JRE and JVM?

Answer-

JVM

JVM is an acronym for Java Virtual Machine, it is an abstract machine which provides the runtime environment in which java bytecode can be executed. It is a specification. JVMs are available for many hardware and software platforms (so JVM is platform dependent).

JRE

JRE stands for Java Runtime Environment. It is the implementation of JVM.

JDK

JDK is an acronym for Java Development Kit. It physically exists. It contains JRE + development tools.

Q-41 How many types of memory areas are allocated by JVM?

Many types:

1. Class(Method) Area

2. Heap
3. Stack
4. Program Counter Register
5. Native Method Stack

Q-42 What is JIT compiler?

Answer- Just-In-Time(JIT) compiler: It is used to improve the performance. JIT compiles parts of the byte code that have similar functionality at the same time, and hence reduces the amount of time needed for compilation.Here the term “compiler” refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

Q-43 What is platform?

A platform is basically the hardware or software environment in which a program runs. There are two types of platforms software-based and hardware-based. Java provides software-based platform.

Q-44 What is the main difference between Java platform and other platforms?

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:

1. Runtime Environment
2. API(Application Programming Interface)

Q-45 What gives Java its 'write once and run anywhere' nature?

The bytecode. Java is compiled to be a byte code which is the intermediate language between source code and machine code. This byte code is not platform specific and hence can be fed to any platform.

Q-46 What is classloader?

The classloader is a subsystem of JVM that is used to load classes and interfaces.There are many types of classloaders e.g. Bootstrap classloader, Extension classloader, System classloader, Plugin classloader etc.

Q-47 What is method overriding:

Answer- If a subclass provides a specific implementation of a method that is already provided by its parent class, it is known as Method Overriding. It is used for runtime polymorphism and to provide the specific implementation of the method.

Q-48 Can we override static method?

No, you can't override the static method because they are the part of class not object.

Q-49 Why we cannot override static method?

It is because the static method is the part of class and it is bound with class whereas instance method is bound with object and static gets memory in class area and instance gets memory in heap.

Q-50 Can we override the overloaded method?

Yes.