**1. Why is Java a platform independent language?**

[**Java language**](https://www.interviewbit.com/blog/features-of-java/) was developed in such a way that it does not depend on any hardware or software due to the fact that the [**compiler**](https://www.interviewbit.com/online-java-compiler/) compiles the code and then converts it to platform-independent byte code which can be run on multiple systems.

* The only condition to run that byte code is for the machine to have a runtime environment (JRE) installed in it

**2. Why is Java not a pure object oriented language?**

Java supports primitive data types - byte, boolean, char, short, int, float, long, and double and hence it is not a pure [**object oriented language**](https://www.interviewbit.com/oops-interview-questions/)

Non primitive date type- class,object,string,array,interfaces.

**3. How is Java different from C++?**

* C++ is only a  compiled language, whereas Java is compiled as well as an interpreted language.
* Java programs are machine-independent whereas a c++ program can run only in the machine in which it is compiled.
* C++ allows users to use pointers in the program. Whereas java doesn’t allow it. Java internally uses pointers.
* C++ supports the concept of Multiple inheritances whereas Java doesn't support this. And it is due to avoiding the complexity of name ambiguity that causes the diamond problem.

### 4. Pointers are used in C/ C++. Why does Java not make use of pointers?

Pointers are quite complicated and unsafe to use by beginner programmers. Java focuses on code simplicity, and the usage of pointers can make it challenging. Pointer utilization can also cause potential errors. Moreover, security is also compromised if pointers are used because the users can directly access memory with the help of pointers.

Thus, a certain level of abstraction is furnished by not including pointers in Java. Moreover, the usage of pointers can make the procedure of garbage collection quite slow and erroneous. Java makes use of references as these cannot be manipulated, unlike pointers

**5. What are the default values assigned to variables and instances in java?**

* There are no default values assigned to the variables in java. We need to initialize the value before using it. Otherwise, it will throw a compilation error of (**Variable might not be initialized**).
* But for instance, if we create the object, then the default value will be initialized by the default constructor depending on the data type.
* If it is a reference, then it will be assigned to null.
* If it is numeric, then it will assign to 0.
* If it is a boolean, then it will be assigned to false. Etc.

**6. What do you mean by data encapsulation?**

* Data Encapsulation is an Object-Oriented Programming concept of hiding the data attributes and their behaviours in a single unit.
* It helps developers to follow modularity while developing software by ensuring that each object is independent of other objects by having its own methods, attributes, and functionalities.
* It is used for the security of the private properties of an object and hence serves the purpose of data hiding.

### ****7. Explain JDK, JRE and JVM?****

|  |  |  |
| --- | --- | --- |
| **JDK vs JRE vs JVM** | | |
| **JDK** | **JRE** | **JVM** |
| It stands for Java Development Kit. | It stands for Java Runtime Environment. | It stands for Java Virtual Machine. |
| It is the tool necessary to compile, document and package Java programs. | JRE refers to a runtime environment in which Java bytecode can be executed. | It is an abstract machine. It is a specification that provides a run-time environment in which Java bytecode can be executed. |
| It contains JRE + development tools. | It’s an implementation of the JVM which physically exists. | JVM follows three notations: Specification, **Implementation,**and **Runtime Instance**. |

**8. Explain public static void main(String args[]) in Java.**

main() in Java is the entry point for any Java program. It is always written as **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. main() is made static in Java so that it can be accessed without creating the instance of a Class. In case, main is not made static then the compiler will throw an error as **main**() is called by the JVM before any objects are made and only static methods can be directly invoked via the 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.

### ****12. What is the difference between equals() and == in Java?****

Equals() method is defined in Object class in Java and used for checking equality of two objects defined by business logic.

“==” or equality operator in Java is a binary operator provided by Java programming language and used to compare primitives and objects. *public boolean equals(Object o)* is the method provided by the Object class. The default implementation uses == operator to compare two objects. For example: method can be overridden like String class. equals() method is used to compare the values of two objects.

### ****9. Why Java is platform independent?****

Java is called platform independent because of its byte codes which can run on any system irrespective of its underlying operating system.

### ****10. Why Java is not 100% Object-oriented?****

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

**11. What are constructors in Java?**

In Java, constructor refers to a block of code which is used to initialize an object. It must have the same name as that of the class. Also, it has no return type and it is automatically called when an object is created.

There are two types of constructors:

1. **Default Constructor:** In Java, a default constructor is the one which does not take any inputs. In other words, default constructors are the no argument constructors which will be created by default in case you no other constructor is defined by the user. Its main purpose is to initialize the instance variables with the default values. Also, it is majorly used for object creation.
2. **Parameterized Constructor:** The parameterized constructor in Java, is the constructor which is capable of initializing the instance variables with the provided values. In other words, the constructors which take the arguments are called parameterized constructors.
3. **Q8. What is the difference between Array list and vector in Java?**

|  |  |
| --- | --- |
| **ArrayList** | **Vector** |
| Array List is not synchronized. | Vector is synchronized. |
| Array List is fast as it’s non-synchronized. | Vector is slow as it is thread safe. |
| If an element is inserted into the Array List, it increases its Array size by 50%. | Vector defaults to doubling size of its array. |
| Array List does not define the increment size. | Vector defines the increment size. |
| Array List can only use Iterator for traversing an Array List. | Vector can use both Enumeration and Iterator for traversing. |

**13.What is the importance of reflection in Java?**

Reflection is a runtime API for inspecting and changing the behavior of methods, classes, and interfaces. Java Reflection is a powerful tool that can be really beneficial. Java Reflection allows you to analyze classes, interfaces, fields, and methods during runtime without knowing what they are called at compile time. Reflection can also be used to create new objects, call methods, and get/set field values. External, user-defined classes can be used by creating instances of extensibility objects with their fully-qualified names. Debuggers can also use reflection to examine private members of classes.

**14. Can you call a constructor of a class inside another constructor?**

Yes, we can call a constructor of a class inside another constructor. This is also called as constructor chaining. Constructor chaining can be done in 2 ways-

1. **Within the same class:** For constructors in the same class, the this() keyword can be used.
2. **From the base class:** The super() keyword is used to call the constructor from the base class.  
   The constructor chaining follows the process of inheritance. The constructor of the sub class first calls the constructor of the super class. Due to this, the creation of sub class’s object starts with the initialization of the data members of the super class. The constructor chaining works similarly with any number of classes. Every constructor keeps calling the chain till the top of the chain.

**15. How is the creation of a String using new() different from that of a literal?**  
When we create a string using new(), a new object is created. Whereas, if we create a string using the string literal syntax, it may return an already existing object with the same name.

**16.What is a package in Java? List down various advantages of packages.**

Packages in Java, are the collection of related classes and interfaces which are bundled together. By using packages, developers can easily modularize the code and optimize its reuse. Also, the code within the packages can be imported by other classes and reused. Below I have listed down a few of its advantages:

* Packages help in avoiding name clashes
* They provide easier access control on the code
* Packages can also contain hidden classes which are not visible to the outer classes and only used within the package
* Creates a proper hierarchical structure which makes it easier to locate the related classes