1. Is Java Platform Independent if then how?

* Java is platform-independent because it uses a virtual machine.

* The Java programming language and all APIs are compiled into bytecodes.

* Bytecodes are effectively platform-independent.

* The virtual machine takes care of the differences between the bytecodes for the different platforms.

2. What are the top Java Features?

* Object Oriented. In Java, everything is an Object. ...

* Platform Independent. ...

* Simple. ...

* Secure. ...

* Architecture-neutral. ...

* Portable. ...

* Robust. ...

* Multithreaded

3. What is JVM?

* JVM (Java Virtual Machine) is an abstract machine.

* It is a specification that provides runtime environment in which java bytecode can be executed.

4. What is JIT?

* Just-in-time, or JIT, is an inventory management method in which goods are received from suppliers only as they are needed.

5. What are Memory storages available with JVM?

* The JVM divides its memory into two main categories: heap memory and non-heap memory.

* Heap memory is the part with which people are typically the most familiar.

* It's where objects that are created by the application are stored.

* They remain there until they are no longer referenced and are garbage collected.

6. What is a class loader?

* The Java Class Loader is a part of the Java Runtime Environment that dynamically loads Java classes into the Java Virtual Machine

7. Difference between JVM, JRE, and JDK.

* JDK is the development platform, while JRE is for execution. JVM is the foundation, or the heart of Java programming language,

* and ensures the program's Java source code will be platform-agnostic.

* JVM is included in both JDK and JRE – Java programs won't run without it.

8. What is Java String Pool?

* String pool is a storage space in the Java heap memory where string literals are stored.

* It is also known as String Constant Pool or String Intern Pool. It is privately maintained by the Java String class.

* By default, the String pool is empty.

9. What are Packages in Java?

* A Java package is a set of classes, interfaces, and sub-packages that are similar.

* In Java, it divides packages into two types: built-in packages and user-defined packages.

* Built-in Packages (packages from the Java API) and User-defined Packages are the two types of packages (create your own packages).

10. Why Packages are used?

* We use packages to avoid name conflicts, and to write a better maintainable code.

11. What are the advantages of Packages in Java?

* The packages organize the group of classes into a single API unit.

* It will control the naming conflicts.

* The access protection will be easier. ...

* Easy to locate the related classes.

* Reuse the existing classes in packages.

12. Differentiate between instance and local variables.

* An instance variable is a variable that is declared in a class but outside a method while the local variable is a variable declared within a method or a constructor. Thus, this is the main difference between instance variable and local variable.

13. Explain the difference between instance variable and a class variable?

* Instance variables are fields declared within a class but outside any method. They are used to store unique data for each instance of the class. Unlike class variables (static variables), which are shared among all instances, instance variables have distinct values specific to each individual object.

14. What is a static variable?

* In computer programming, a static variable is a variable that has been allocated "statically", meaning that its lifetime (or "extent") is the entire run of the program.

15. What is an array in Java?

* An array is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed.

16. What are the types of an array?

* Arrays are classified into two types based on their dimensions : single-dimensional and multi-dimensional.

* Logically, a single-dimensional array represents a linear collection of data, and a two-dimensional array represents a mathematical matrix.

17. What are the main concepts of OOPs in Java?

* The main principles of object-oriented programming are abstraction, encapsulation, inheritance, and polymorphism.

* These concepts aim to implement real-world entities in programs.

18. What are Classes in Java?

* A class — in the context of Java — is a template used to create objects and to define object data types and methods.

* Classes are categories, and objects are items within each category. All class objects should have the basic class properties.

19. What is the difference between static (class) method and instance method?

* Static methods can be called without the object of the class. Instance methods require an object of the class.

20. What is this keyword in Java?

* The this keyword refers to the current object in a method or constructor.

21. What are Brief Access Specifiers and Types of Access Specifiers?

* In Java, access specifiers or access modifiers are used to control the visibility and accessibility of classes, methods, and variables.

* There are four access modifiers in Java, namely: public, protected, private, and default .

22. What is an object?

* In the Java programming language, an object is an instance of a Java class, meaning it is a copy of a specific class.

23. What is the constructor?

* A constructor in Java Programming is a block of code that initializes (constructs) the state and value during object creation.

24. How many types of constructors are used in Java?

* In Java, constructors can be divided into 4 types: No-Argument Constructor. Parameterized Constructor. Default Constructor, and copy constructor.

25. What is an Interface?

* In Java, an interface specifies the behavior of a class by providing an abstract type. As one of Java's core concepts, abstraction, polymorphism, and multiple inheritance are supported through this technology. Interfaces are used in Java to achieve abstraction.

26. What are the differences between abstract class and interface?

* An abstract class comes with non-final variables. A Java interface comes with variables declared final by default.

27. Define Inheritance.

* Inheritance in Java is a concept that acquires the properties from one class to other classes; for example, the relationship between father and son.

28. What are the different types of inheritance in Java?

In this Article, you will learn about the types of Inheritance in Java.

* Single-level inheritance.
* Multi-level Inheritance.
* Hierarchical Inheritance.
* Multiple Inheritance.
* Hybrid Inheritance.

29. What is multiple inheritance? Is it supported by Java?

* Java doesn't support multiple inheritances in classes because it can lead to diamond problem and rather than providing some complex way to solve it, there are better ways through which we can achieve the same result as multiple inheritances.