Java Interview Questions

1. What do you mean by platform independence of Java?

Platform independence means that you can run the same Java Program in any Operating System. For example, you can write java program in Windows and run it in Mac OS.

2. What is JVM and is it platform independent?

JVM is responsible for converting byte code into machine readable code. JVM is not platform independent, thats why you have different JVM for different operating systems. We can customize JVM with Java Options, such as allocating minimum and maximum memory to JVM. It's called virtual because it provides an interface that doesn't depend on the underlying OS.

3. What is the difference among JDK, JRE and JVM?

Java Development Kit (JDK) is for development purpose and JVM is a part of it to execute the java programs. JDK provides all the tools, executables and binaries required to compile, debug and execute a Java Program. The execution part is handled by JVM to provide machine independence.

Java Runtime Environment (JRE) is the implementation of JVM. JRE consists of JVM and java binaries and other classes to execute any program successfully. JRE doesn't contain any development tools like java compiler, debugger etc. If you want to execute any java program, you should have JRE installed.

4. What is the importance of main method in Java?

main() method is the entry point of any standalone java application. Java main method is public and static so that Java runtime can access it without initializing the class. The input parameter is an array of String through which we can pass runtime arguments to the java program.

5. Name some OOPS Concepts in Java

Java is based on Object Oriented Programming Concepts, following are some of the OOPS concepts implemented in java programming.

I. Abstraction
II. Encapsulation
III. Polymorphism
IV. Inheritance

6. What is overloading and overriding in java?

When we have more than one method with the same name in a single class but the arguments are different, then it is called as method overloading.

Overriding concept comes in picture with inheritance when we have two methods with same signature, one in parent class and another in child class. We can use @Override annotation in the child class overridden method to make sure if parent class method is changed, so as child class.

7. Can we overload main method?

Yes, we can have multiple methods with name "main" in a single class.

8. Which class is the superclass of all classes?

java.lang.Object is the root class for all the java classes and we don't need to extend it.

9. Can we have multiple public classes in a java source file?

We can't have more than one public class in a single java source file. A single source file can have multiple classes that are not public.

10. What is Java Package and which package is imported by default?

Java package is the mechanism to organize the java classes by grouping them. The grouping logic can be based on functionality or modules based. A java class fully classified name contains package and class name. For example, java.lang.Object is the fully classified name of Object class that is part of java.lang package.

java.lang package is imported by default and we don't need to import any class from this package explicitly.

11. What are access modifiers?

Java provides access control through public, private and protected access modifier keywords. When none of these are used, it's called default access modifier. A java class can only have public or default access modifier.

12. What is *final* keyword?

final keyword is used with Class to make sure no other class can extend it, for example String class is final and we can't extend it. We can use the final keyword with methods to make sure child classes can't override it.

final keyword can be used with variables to make sure that it can be assigned only once. However the state of the variable can be changed, for example, we can assign a *final* variable to an object only once but the object variables can change later on. Java interface variables are by default final and static.

13. What is *static* keyword?

static keyword can be used with class level variables to make it global i.e all the objects will share the same variable.

static keyword can be used with methods also. A static method can access only static variables of class and invoke only static methods of the class.

14. What is an interface?

Interfaces are core part of java programming language and used a lot not only in JDK but also java design patterns, most of the frameworks and tools. Interfaces provide a way to achieve abstraction in java and used to define the contract for the sub-classes to implement.

15. What is an abstract class?

Abstract classes are used in java to create a class with some default method implementation for sub-classes. An abstract class can have abstract method without body and it can have methods with implementation also.

abstract keyword is used to create a abstract class. Abstract classes can't be instantiated and mostly used to provide base for sub-classes to extend and implement the abstract methods and override or use the implemented methods in abstract class.

16. What is the difference between *abstract* class and *interface*?

abstract keyword is used to create abstract class whereas *interface* is the keyword for interfaces. Abstract classes can have method implementations whereas interfaces can't. A class can extend only one abstract class but it can implement multiple interfaces.

17. What are Wrapper classes?

Java wrapper classes are the Object representation of eight primitive types in java. All the wrapper classes in java are immutable and final.

18. What does super keyword do?

super keyword can be used to access super class method when you have overridden the method in the child class. We can use super keyword to invoke super-class constructor in child class constructor but in this case, it should be the first statement in the constructor method.

19. Can an interface implement or extend another interface?

Interfaces don't implement another interface, they extend it. Since interfaces can't have method implementations, there is no issue of diamond problem. That's why we have multiple inheritance in interfaces i.e an interface can extend multiple interfaces.

20. What is Java Annotations?

Java Annotations provide information about the code and they have no direct effect on the code they annotate. Annotations are introduced in Java 5. Annotation is metadata about the program embedded in the program itself. It can be parsed by the annotation parsing tool or by the compiler. We can also specify annotation availability to either compile time only or till runtime also. Java Built-in annotations are @Override, @Deprecated and @SuppressWarnings.

21. What is inner class in java?

We can define a class inside a class and they are called nested classes. Any non-static nested class is known as inner class. Inner classes are associated with the object of the class and they can access all the variables and methods of the outer class. Since inner classes are associated with the instance, we can't have any static variables in them.

22. What is anonymous inner class?

A local inner class without name is known as anonymous inner class. An anonymous class is defined and instantiated in a single statement. Anonymous inner class always extend a class or implement an interface. Since an anonymous class has no name, it is not possible to define a constructor for an anonymous class. Anonymous inner classes are accessible only at the point where it is defined.

23. What is *break* and *continue* statement?

We can use *break* statement to terminate for, while, or do-while loop. We can use *break* statement in switch statement to exit the switch case. We can use *break* with label to terminate the nested loops.

The *continue* statement skips the current iteration of a for, while or do-while loop. We can use continue statement with the label to skip the current iteration of the outermost loop.

24. What is *this* keyword?

this keyword provides the reference to the current object and it's mostly used to make sure that object variables are used, not the local variables having the same name.

25. Can we have try without catch block?

Yes, we can have try-finally statement and hence avoiding catch block.

26. What is Garbage Collection?

Garbage Collection is the process of looking at heap memory, identifying which objects are in use and which are not, and deleting the unused objects. In Java, process of deallocating memory is handled automatically by the garbage collector.

27. What is Serialization and Deserialization?

We can convert a Java object to an Stream that is called Serialization. Once an object is converted to Stream, it can be saved to file or send over the network or used in socket connections.

The object should implement Serializable interface and we can use java.io.ObjectOutputStream to write object to file or to any OutputStream object.

The process of converting stream data created through serialization to Object is called deserialization.

28. Can we use String with switch case?

One of the Java 7 feature was improvement of switch case of allow Strings. So if you are using Java 7 or higher version, you can use String in switch-case statements.

29. What is difference between Heap and Stack Memory?

Major difference between Heap and Stack memory are as follows:

1. Heap memory is used by all the parts of the application whereas stack memory is used only by one thread of execution.
2. Whenever an object is created, it's always stored in the Heap space and stack memory contains the reference to it. Stack memory only contains local primitive variables and reference variables to objects in heap space.
3. Memory management in the stack is done in LIFO manner whereas it's more complex in Heap memory because it's used globally.

30. Java Compiler is stored in JDK, JRE or JVM?

The task of java compiler is to convert java program into bytecode, we have javac executable for that. So it must be stored in JDK, we don't need it in JRE and JVM is just the specs.