

1. What is Java?

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.

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

A package is a namespace that organizes a set of related classes and interfaces. ... Because software written in the Java programming language can be composed of hundreds or thousands of individual classes, it makes sense to keep things organized by placing related classes and interfaces into packages.

Advantages of Packages:

- 1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.
- 2) Java package provides access protection.
- 3) Java package removes naming collision.

3. Explain JDK, JRE and JVM?

JDK:

The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the Java Runtime Environment (JRE), an interpreter/loader (Java), a compiler (javac), a documentation generator (Javadoc) and other tools needed in Java development.

JDK is a superset of JRE, it contains everything that JRE has along with development tools

Such as compiler, debugger, etc.

JRE:

Java Runtime Environment (to say JRE) is an installation package which provides environment to only run(not develop) the java program(or application) onto your machine. JRE is only used by them who only wants to run the Java Programs i.e. end users of your system.

JVM:

Java Virtual machine(JVM) is a very important part of both JDK and JRE because it is contained or inbuilt in both. Whatever Java program you run using JRE or JDK goes into JVM and JVM is responsible for executing the java program line by line hence it is also known as interpreter.

The primary function of JVM is to execute bytecode produced by the compiler. Each OS has different JVM, however the output they produce after execution of bytecode is same across

all OS. That's why JAVA is platform independent.

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

Whether class contains main() method or not and whether main() method is declared according to requirement or not these things won't be checked by compiler. At runtime, JVM is responsible to check these things.

At runtime if JVM is unable to find required main() method then we will get runtime exception saying **NosuchMethod:main**

public : To call by JVM from anywhere.

static : Without existing object also JVM has to call this method & main method no way related to any object.

void : main() method won't return anything to JVM.

main : This is method name which is configured inside JVM.

String[] args : command line argument

5. What are the differences between C++ and Java?

Java	C++
1. Java does not support pointers, unions, operator overloading and structure.	1. C++ supports pointers, unions, operator overloading and structure.
2. Java supports garbage collection.	2. C++ does not support garbage collection.
3. Java is platform independent.	3. C++ is platform dependent.
4. Java supports inheritance except for multiple inheritance	4. C++ supports inheritance including multiple inheritances
5. Java is interpreted.	5. C++ is compiled.
6. Java does not support destructor	6. C++ supports destructors.

6. Why Java is platform independent?

Unlike many other programming languages including C and C++, when JAVA is compiled, it is not compiled into platform independent byte code. This byte code is distributed over the web and interpreted by virtual machine (JVM) on whichever platform it is being run.

7. What are wrapper classes in Java?

The wrapper class in Java provides the mechanism *to convert primitive into object and object into primitive*.

Java is an object-oriented programming language, so we need to deal with objects many times like in Collections, Serialization, Synchronization, etc. Let us see the different scenarios, where we need to use the wrapper classes.

8. Why pointers are not used in Java?

Memory access via pointer arithmetic - this is fundamentally unsafe. Java has a robust security model and disallows pointer arithmetic for this reason. Java does have reference to objects; it just doesn't call them pointers. Any normal object reference works like a pointer would.

9. List some features of Java?

Object Oriented

In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent

Unlike many other programming languages including C and C++, when Java is compiled, it is not compiled into platform specific machine, rather into platform-independent byte code. This byte code is distributed over the web and interpreted by the Virtual Machine (JVM) on whichever platform it is being run on.

Simple

Java is designed to be easy to learn. If you understand the basic concept of OOP Java, it would be easy to master.

Secure

With Java's secure feature it enables to develop virus-free, tamper-free systems. Authentication techniques are based on public-key encryption.

Architecture-neutral

Java compiler generates an architecture-neutral object file format, which makes the compiled code executable on many processors, with the presence of Java runtime system.

Portable

Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. The compiler in Java is written in ANSI C with a clean portability boundary, which is a POSIX subset.

Robust

Java makes an effort to eliminate error-prone situations by emphasizing mainly on compile time error checking and runtime checking.

Multithreaded

With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously. This design feature allows the developers to construct interactive applications that can run smoothly.

Interpreted

Java byte code is translated on the fly to native machine instructions and is not stored anywhere. The development process is more rapid and analytical since the linking is an incremental and light-weight process.

High Performance

With the use of Just-In-Time compilers, Java enables high performance.

Distributed

Java is designed for the distributed environment of the internet.

Dynamic

Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment. Java programs can carry an extensive amount of run-time information that can be used to verify and resolve accesses to objects at run-time.

10. Why is Java Architectural Neutral?

Java compiler generates an architecture-neutral object file format which makes the compiled code to be executable on many processors, with the presence of Java runtime system.

11. How Java enabled High Performance?

With the use of Just-in time compilers, java enables high performance. Because JIT executes the byte code only on requirement basis.

12. Why Java is considered dynamic?

Java is considered to be more dynamic than C or C++ since it is designed to adapt to an evolving environment.

Java programs can carry extensive amount of run-time information that can be used to verify and resolve accesses to objects on run-time.

13. What is Java Virtual Machine and how it is considered in context of Java's platform independent feature?

Java Virtual Machine (JVM) is a specification that provides runtime environment in which java bytecode(. class files) can be executed. The JVM is the platform. ... JVM makes this possible because it is aware of the specific instruction lengths and other particularities of the platform (Operating System).

14. List two Java IDE's?

The following are the best Java IDEs that are mostly used in the world:

- Eclipse
- NetBeans
- IntelliJ IDEA
- BlueJ
- JCreator
- JDeveloper
- MyEclipse
- Greenfoot
- DrJava
- Xcode
- Codenvy

15.Why Java is called as “Platform” ?

Platform is a software and hardware programs that runs. JAVA is platform independent because it having its own JVM so that it can run on any platform . java is platform independent , which means once written you can run it anywhere. The platform is a hardware or software used to run an application.

16.Is Java Pure-Object oriented Language ?

Java is not an pure object oriented language because it supports Primitive datatype such as int, byte, long... etc, to be used, which are not objects. Contrast with a pure OOP language like Smalltalk, where there are no primitive types, and boolean, int and methods are all objects. ... All user defined types are objects.

Java language is not a Pure Object Oriented Language as it contain these properties: Primitive Data Type ex. ... In Smalltalk, primitive values such as integers, booleans and characters are also objects. In Java, we have predefined types as non-objects (primitive types).

17.Which version of java have u learned? Name some of the new features added to it.

Oracle released a new version of Java as Java 8 in March 18, 2014. It was a revolutionary release of the Java for software development platform. It includes various upgrades to the Java programming, JVM, Tools and libraries.

Java 8 provides following features for Java Programming:

- Lambda expressions,
- Method references,
- Functional interfaces,
- Stream API,
- Default methods,
- Base64 Encode Decode,
- Static methods in interface,
- Optional class,
- Collectors class,
- ForEach() method,
- Parallel array sorting,
- Nashorn JavaScript Engine,
- Parallel Array Sorting,
- Type and Repating Annotations,
- IO Enhancements,
- Concurrency Enhancements,
- JDBC Enhancements etc.

18.What gives Java its 'write once and run anywhere' nature?

The "Write Once, Run Everywhere" slogan refers to the fact that an application written in Java can be run on any hardware which has the Java Virtual Machine (JVM), and that the JVM is now licensed to hundreds of operating systems vendors including Microsoft for Windows.

19.Difference between path and classpath.

Let us clear the difference in points:

PATH

- a) An environment variable which is used by the operating system to find the executables.
- b) PATH is nothing but setting up an environment for operating system. Operating System will look in this PATH for executables.
- c) Refers to the system

CLASSPATH

- a) An environment variable which is used by the Java compiler to find the path, of classes i.e in J2EE we give the path of jar files.
- b) Classpath is nothing but setting up the environment for Java. Java will use to find compiled classes.
- c) Refers to the Developing Environment.

20.What is the signature of main function in java ?

The method signature for the main() method contains three modifiers: public indicates that the main() method can be called by any object. static indicates that the main() method is a class method. void indicates that the main() method has no return value.

Void keyword acknowledges the compiler that main() method does not return any value. main(): It is a default signature which is predefined in the JVM. It is called by JVM to execute a program line by line and end the execution after completion of this method. We can also overload the main() method.

21.What is the difference between JDK and JRE?

JDK is a software development kit whereas JRE is a software bundle that allows Java program to run, whereas JVM is an environment for executing bytecode. The full form of JDK is Java Development Kit, while the full form of JRE is Java Runtime Environment, while the full form of JVM is Java Virtual Machine.

JDK includes the JRE plus command-line development tools such as compilers and debuggers that are necessary or useful for developing applets and applications. JRE is basically the Java Virtual Machine where your Java programs run on. JDK is an abstract machine.

22.What is JVM ? What it does?

A Java virtual machine (JVM) is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java bytecode. ... The JVM reference implementation is developed by the OpenJDK project as open source code and includes a JIT compiler called HotSpot. The JVM has two primary functions: to allow Java programs to run on any device or operating system (known as the "Write once, run anywhere" principle), and to manage and optimize program memory.

23.Why JVM is called as “virtual machine”?

The JVM is "virtual" because it is generally implemented in software on top of a "real" hardware platform and operating system. All Java programs are compiled for the JVM. Therefore, the JVM must be implemented on a particular platform before compiled Java programs will run on that platform.

24.What are the main components of JVM? Explain them. Or Explain JVM Architecture.

It performs three major functions viz. Loading, Linking, and Initialization. JVM Method Area stores class structures like metadata, the constant

runtime pool, and the code for methods. All the Objects, their related instance variables, and arrays are stored in the heap.

Java Virtual Machine (JVM) is a engine that provides runtime environment to drive the Java Code or applications. It converts Java bytecode into machines language. JVM is a part of Java Run Environment (JRE). In other programming languages, the compiler produces machine code for a particular system.

25.What is the difference between Java compiler (javac) and JIT ?

When compiling a java program, the static compiler that is run using the command javac converts the source code to byte code which are in the form of . class files. ... JIT compiles the code when it is needed but not before runtime.

26.Is Empty .java file name a valid source file name?

Since, you cannot leave class name empty as well as you can't also change its name to any other since it is public. ... If you write a file in Java which is already present in the location, it will be overwritten automatically. Unless you are writing to that file with an append flag set to True.

27. Is JRE different for different Platforms ?

Whenever we try to run the code, JVM requires some library set and files for code execution and these files are presented in JRE. JRE = JVM + set of libraries. ... JRE is also platform dependent. That means we have different JRE versions for different platforms.

28.Difference between C++ and java in terms of object creation.

C++ supports manual object management with the help of new and delete keywords whereas Java has built-in automatic garbage collection. C++ supports structures whereas Java doesn't supports structures. C++ supports unions while Java doesn't support unions.

29.Who invokes main() function ?

In 'C', the "main" function is called by the operating system when the user runs the program and it is treated the same way as every function, it has a return type. Although you can call the main() function within itself and it is called recursion.

30.What is .class file known as ?

A Java class file is a file (with the .class filename extension) containing Java bytecode that can be executed on the Java Virtual Machine (JVM). A Java class file is usually produced by a Java compiler from Java programming language source files.

31.Can we define more than one public class in a java source code ? what is the rule of public class and file name?

No, while defining multiple classes in a single Java file you need to make sure that only one class among them is public. If you have more than one public classes a single file a compile-time error will be generated.

32.What is JIT compiler?

The Just-In-Time (JIT) compiler is a component of the runtime environment that improves the performance of Java™ applications by compiling bytecodes to native machine code at run time. ... The JIT compiler helps improve the performance of Java programs by compiling bytecodes into native machine code at run time.

33.How many types of memory areas are allocated by JVM?

The memory in the JVM divided into 5 different parts:

- Class(Method) Area.
- Heap.
- Stack.
- Program Counter Register.
- Native Method Stack.

34.What is the rule for local member in java.

Local variables cannot use any of the access level since their scope is only inside the method. Final is the Only Non Access Modifier that can be applied to a local variable. Local variables are not assigned a default value, hence they need to be initialized.

35.What are the various access specifiers in Java?

Access Specifiers in Java | Access Modifiers

- Public Access Specifier.
- Protected Access Specifier.
- Default Access Specifier.

- Private Access Specifiers.

36.What is the rule for local member in java.

Local variables cannot use any of the access level since their scope is only inside the method. Final is the Only Non Access Modifier that can be applied to a local variable. Local variables are not assigned a default value, hence they need to be initialized.

37.What is native code?

Native code is computer programming (code) that is compiled to run with a particular processor (such as an Intel x86-class processor) and its set of instructions.

... Java bytecode and Microsoft's Intermediate Language can be compiled into native code before execution by a just-in-time compiler for faster performance.

38.Why there is no sizeof operator in java ?

Because the size of primitive types is explicitly mandated by the Java language. There is no variance between JVM implementations. Moreover, since allocation is done by the new operator depending on its argument there is no need to specify the amount of memory needed.

39.What kinds of programs u can develop using Java ?

- Mobile Applications
- Desktop GUI Applications
- Web-based Applications
- Enterprise Applications
- Scientific Applications
- Gaming Applications
- Big Data technologies
- Business Applications

- Distributed Applications
- Cloud-based Applications

40. U have reference type as a member of class. What is the default value it gets?

'null'

The default value of a reference type variable is null when they are not initialized. Null means not referring to any object.

41. What is the job done by classloader ?

The Java ClassLoader is a part of the Java Runtime Environment that dynamically loads Java classes into the Java Virtual Machine. The Java run time system does not need to know about files and file systems because of classloaders. Java classes aren't loaded into memory all at once, but when required by an application.

42. Explain the hierarchy of classloaders in java.

ClassLoader is hierarchical in loading a class into memory. Whenever a request is raised to load a class, it delegates it to the parent classloader. This is how uniqueness is maintained in the runtime environment. If the parent class loader doesn't find the class then the class loader itself tries to load the class.

43. What is the role played by Bytecode Verifier ?

The bytecode verifier acts as a sort of gatekeeper: it ensures that code passed to the Java interpreter is in a fit state to be executed and can run without fear of breaking the Java interpreter. Imported code is not allowed to execute by any means until after it has passed the verifier's tests.

44. What are the memory areas allocated by JVM ?

The memory in the JVM divided into 5 different parts:

- Class(Method) Area.
- Heap.
- Stack.
- Program Counter Register.
- Native Method Stack.

Heap – Runtime storage allocation for objects (reference types). Stack – Storage for local variables and partial results. A stack contains frames and allocates one for each thread. ... Native method stacks – It contains all the native methods used by the application.

45.What kinds of programs u can develop using Java?

- Mobile Applications
- Desktop GUI Applications
- Web-based Applications
- Enterprise Applications
- Scientific Applications
- Gaming Applications
- Big Data technologies
- Business Applications
- Distributed Applications
- Cloud-based Applications

46.When parseInt() method can be used?

The Integer.parseInt() java method is used primarily in parsing a String method argument into an Integer object. The Integer object is a wrapper class for the int primitive data type of java API.

Convert a string to an integer with the parseInt method of the Java Integer class. The parseInt method is to convert the String to an int and throws a NumberFormatException if the string cannot be converted to an int type.

47.What is finalized() method ?

Finalize() is the method of Object class. This method is called just before an object is garbage collected. finalize() method overrides to dispose system resources, perform clean-up activities and minimize memory leaks.

The finalize() method of Object class is a method that the Garbage Collector always calls just before the deletion/destroying the object which is eligible for Garbage Collection, so as to perform clean-up activity.

48.Difference between C++ pointer and Java reference.

Reference: A reference is a variable that refers to something else and can be used as an alias for that something else. ... Pointers are a particular implementation of the

concept of a reference, and the term tends to be used only for languages that give you direct access to the memory address. References are used to refer an existing variable in another name whereas pointers are used to store address of variable. References cannot have a null value assigned but pointer can. A reference variable can be referenced by pass by value whereas a pointer can be referenced by pass by reference.

49. You have reference type as a member of class. What is the default value it gets?
'null'

The default value of a reference type variable is null when they are not initialized. Null means not referring to any object.

50. What are the expressions allowed in switch block of java ?

A switch works with the byte , short , char , and int primitive data types. It also works with enumerated types (discussed in Enum Types), the String class, and a few special classes that wrap certain primitive types: Character , Byte , Short , and Integer (discussed in Numbers and Strings).