

## 1. What is Java?

Java is a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

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

A **java package** is a group of similar types of classes, interfaces and sub-packages.

Package in java can be categorized in two form, built-in package and user-defined package.

There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

### Advantage of Java Package

- 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?

### JVM

JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment in which Java bytecode can be executed. It can also run those programs which are written in other languages and compiled to Java bytecode.

JVMs are available for many hardware and software platforms. JVM, JRE, and JDK are platform dependent because the configuration of each OS is different from each other. However, Java is platform independent. There are three notions of the JVM: *specification*, *implementation*, and *instance*.

The JVM performs the following main tasks:

Loads code, Verifies code, Executes code, Provides runtime environment

### JRE

JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.

### JDK

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools.

JDK is an implementation of any one of the below given Java Platforms released by Oracle Corporation:

Standard Edition Java Platform

Enterprise Edition Java Platform

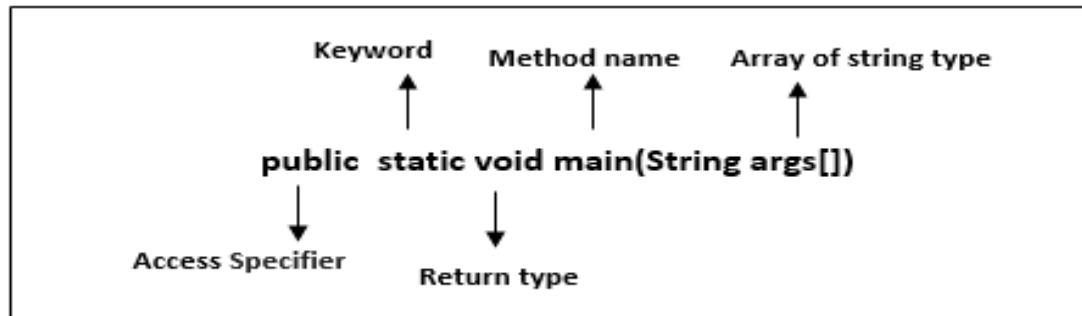
Micro Edition Java Platform

The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application.

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

### Assignment 3

The `main()` is the starting point for JVM to start execution of a Java program. Without the `main()` method, JVM will not execute the program. The syntax of the `main()` method is:



**public:** It is an access specifier. We should use a public keyword before the `main()` method so that JVM can identify the execution point of the program. If we use private, protected, and default before the `main()` method, it will not be visible to JVM.

**static:** You can make a method static by using the keyword static. We should call the `main()` method without creating an object. Static methods are the method which

invokes without creating the objects, so we do not need any object to call the `main()` method.

**void:** In Java, every method has the return type. 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.

**String args[]:** The `main()` method also accepts some data from the user. It accepts a group of strings, which is called a string array. It is used to hold the command line arguments in the form of string values.

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

The languages are based on each other but still, they are different in design and philosophy. The following table describes the major differences between C, C++, and Java. It will help you to select which language you have to learn.

S.N.	Basis	C++	Java
1	Origin	The C++ language is based on the C language.	The Java programming language is based on both C and C++.
2	Programming Pattern	It is an object-oriented programming language.	It is a pure object-oriented programming language.
3	Approach	It uses the bottom-up approach.	It also uses the bottom-up approach.
4	Dynamic or Static	It is also a static programming language.	It is a dynamic programming language.
5	Code Execution	The code is executed directly.	The code is executed by the JVM.
6	Platform Dependency	It is platform dependent.	It is platform-independent because of byte code.
7	Translator	It also uses a compiler only to translate the code into machine language.	Java uses both compiler and interpreter and it is also known as an interpreted language.
8	File Generation	It generates .exe file.	It generates .class file.
9	Number of Keyword	There are <b>60</b> keywords in the C++ language.	There are <b>52</b> keywords in the Java language.
10	Source File Extension	The source file has a .cpp extension.	The source file has a .java extension.
11	Pointer Concept	It also supports pointer.	Java does not support the pointer concept because of security.

### Assignment 3

12	<b>Union and Structure Datatype</b>	It also supports union and structure data types.	It does not support union and structure data types.
13	<b>Pre-processor Directives</b>	It uses pre-processor directives such as #include, #define, #header, etc.	It does not use directives but uses packages.
14	<b>Constructor/ Destructor</b>	It supports both constructor and destructor.	It supports constructors only.
15	<b>Exception Handling</b>	It supports exception handling.	It also supports exception handling.
16	<b>Memory Management</b>	It uses new and delete operator to manage the memory.	It uses a garbage collector to manage the memory.
17	<b>Overloading</b>	Method and operator overloading can be achieved.	Only method overloading can be achieved.
18	<b>goto Statement</b>	It also supports the goto statement.	It does not support the goto statements.
19	<b>Used for</b>	It is widely used for system programming.	It is used to develop web applications, mobile applications, and windows applications.
20	<b>Array Size</b>	An array should be declared with size.	An array can be declared without declaring the size. For example, int num[].

## 6. Why Java is platform independent?

Java is platform independent because it can run on any platform i.e on windows, mac etc. How, Because it contains its own virtual machine which have JRE which runs java program as byte code. You may have different JRE for different platforms Once you installed it in your system you can run any java application on it.

## 7. What are wrapper classes in Java?

A wrapper class wraps around a data type and gives it an object appearance. Wherever, the data type is required as an object, this object can be used. Wrapper classes include methods to unwrap the object and give back the data type. Wrapper classes are used to convert any data type into an object. The wrapper types are Byte, Short, Integer, Long, Character, Boolean, Double, Float.

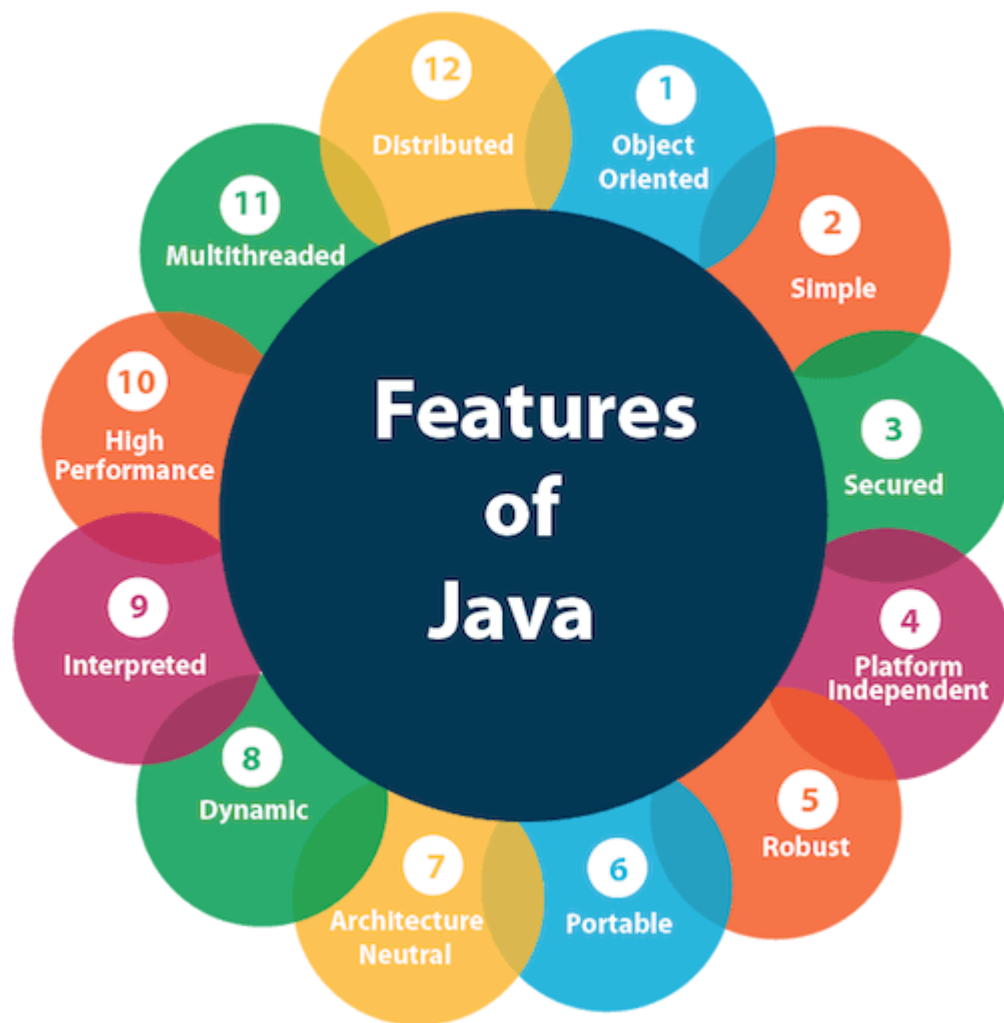
## 8. Why pointers are not used in Java?

The Java language does not provide pointers. Instead, all objects are handled by references, not to be confused with pointers or C++ references. The difference is that Java references do not refer directly to the memory location, but rather contain the pointer to the actual memory location, which the programmer cannot get direct access to. This extra level of indirection is required for GC to work. When GC kicks in and frees objects, that are no longer in use, it then does heap compaction to defragment the heap. To do this, some objects must be moved around in memory. This is possible because there is only one place where this pointer is located, in the reference, and thus the memory can be relocated, old location freed, and pointer changed. In C++ this is not possible because pointers are copied all over the code.

## 9. List some features of Java?

The primary objective of [Java programming](#) language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as java *buzzwords*.

A list of most important features of Java language is given below.



## 10. Why is Java Architectural Neutral?

**Java Architecture** is a collection of components, i.e., **JVM**, **JRE**, and **JDK**. It integrates the process of interpretation and compilation. It defines all the processes involved in creating a Java program. **Java Architecture** explains each and every step of how a program is compiled and executed.

**Java Architecture** can be explained by using the following steps:

There is a process of compilation and interpretation in Java.

Java compiler converts the Java code into byte code.

After that, the JVM converts the byte code into machine code.

The machine code is then executed by the machine.

The following figure represents the **Java Architecture** in which each step is elaborate graphically.

### 11.How Java enabled High Performance?

Java uses Just-In-Time compiler to enable high performance. Just-In-Time compiler is a program that turns Java bytecode, which is a program that contains instructions that must be interpreted into instructions that can be sent directly to the processor.

### 12.Why Java is considered dynamic?

Java is considered as Dynamic because of Byte code[a class file].

A source code written in one platform, the same code can be executed in any platform[ which JDK is installed.].

And it also loads the class files at run time,anything that happens at run time is considered as dynamic.

Java also has a WORA property(Write Once Run Anywhere)

### 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.

### Assignment 3

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 an engine that provides runtime environment to drive the Java Code or applications. It converts Java bytecode into machine 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 support 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

### Assignment 3

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 ?**



### Assignment 3

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. 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.

## **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).