CANTiLEVER
bridging colleges to corporates

JAVA
Preparation Toolkit

CANTILEVER LABS IS OFFICIAL TRAINING PARTNER OF

IIT BOMBAY | IIT MADRAS | IIT HYDERABAD | IIT INDORE |
BITS PILANI | BITS HYDERABAD | NIT ROURKELA | SYMBIOSIS |
JNTU HYDERABAD | SREENIDHI | MAHENDRA UNIVERSITY |
GITAM & many more

PLACEMENT
PREPARATION
TOOL KIT

INDEX
Table of Contents :

PLACEMENT
PREPARATION
TOOL KIT

What is Java?

Java is a popular programming language, created in 1995.
It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

Why Use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming languages in the world
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast, and powerful
- It has huge community support (tens of millions of developers)
- Java is an object-oriented language that gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

History of Java

Java was invented by James Gosling, Patrick Naughton, Chris
Warth, Ed Frank, and Mike Sheridan at Sun Microsystems, Inc. in 1991. Java is related to C++, which is inherited from the language C. The characters of Java are inherited from C and C++ languages. It took approx. Eighteen months to develop the first working version. It was first named as "Oak" but was renamed as "Java" in 1995. It is publicly announced in the spring of 1995.

The basic idea behind creating this language is to create a platform-independent language that is used to develop software for consumer electronic devices such as microwave ovens, remote controls, etc. Initially, it was not designed for Internet applications.

Other languages have the problem that they are designed to compile the code for a specific platform. Let us take the example of C++, it is possible to compile C++ code for any processor but
to do so it requires a full C++ compiler targeted for that particular processor and platform. That makes it expensive and
time-consuming. To overcome this, Gosling and others started
working on a portable and platform-independent language, which led to the creation of Java.

Java had an extreme effect on the Internet by the innovation of a new type of networked program called the Applet. An applet is a Java program that is designed to be transmitted over the Internet and executed by the web browser that is Java-compatible. Applets are the small program that is used to display data provided by the server, handle user input, provide a simple function such as calculator etc.

Java solves the Security and the portability issue of the other language that is being used. The key that allows doing so is the
Bytecode. Bytecode is a highly optimized set of instructions that is designed to be executed by the Java Virtual Machine (JVM). Java programs executed by the JVM also helps to make Java a secure programming language because the JVM contains the application and prevents it from affecting the external systems.

Why to Learn Java Programming?

Java is a MUST for students and working professionals to become a great Software Engineers specially when they are working in Software Development Domain. I will list down some of the key advantages of learning Java Programming:

- 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 machines, rather into platform-independent bytecode. 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 the development of 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 a Java runtime system.

- Portable - Being architecture-neutral and having no implementation-dependent aspects of the specification makes Java portable. Compilers in Java are 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.

Hello world program in Java

```
public class MyFirstJavaProgram {
/* This is my first java program.
* This will print 'Hello World' as the output *1
public static void main(String []args)
{
System.out.println("Hello World"); // prints Hello World
```

Applications of Java

Java is guaranteed to be Write Once, Run Anywhere.

- 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 lightweight 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 on run-time.

Applications of Java

Java is guaranteed to be Write Once, Run Anywhere.

- 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 lightweight 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 on run-time.

How Java is Platform Independent

The meaning of platform-independent is that the java compiled code(byte code) can run on all operating systems.

A program is written in a language that is a human-readable language. It may contain words, phrases, etc which the machine does not understand. For the source code to be understood by the machine, it needs to be in a language understood by machines, typically a machine-level language. So, here comes the role of a compiler. The compiler converts the high-level language (human language) into a format understood by the machines. Therefore, a compiler is a program that translates the source code for another program from a programming language into executable code.

This executable code may be a sequence of machine instructions that can be executed by the CPU directly, or it may be an intermediate representation that is interpreted by a virtual machine. This intermediate representation in Java is the Java Byte Code.

Step by step Execution of Java Program:

- Whenever a program is written in JAVA, the javac compiles it.
- The result of the JAVA compiler is the .class file or the bytecode and not the machine native code (unlike C compiler).
- The bytecode generated is a non-executable code and needs an interpreter to execute on a machine. This interpreter is the JVM and thus the Bytecode is executed by the JVM.
- And finally, the program runs to give the desired output.
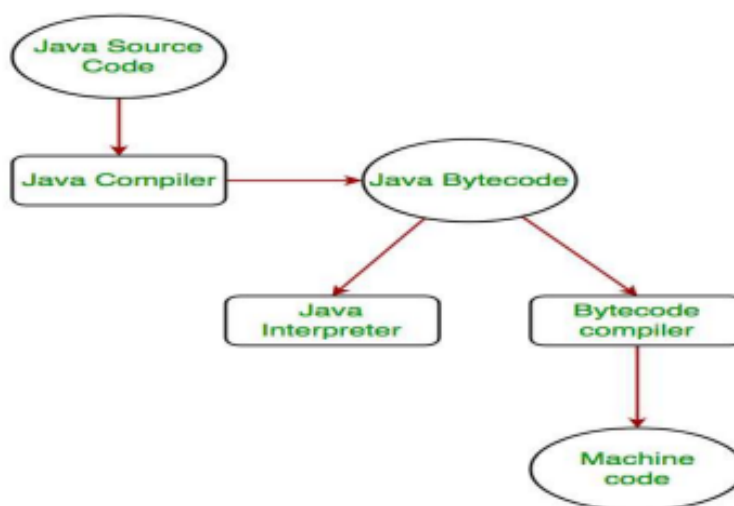
In the case of C or C++ (languages that are not platform independent), the compiler generates an .exe file which is OS dependent. When we try to run this .exe file on another OS it does not run, since it is OS dependent and hence is not compatible with the other OS.

Java is platform-independent but JVM is platform dependent

In Java, the main point here is that the JVM depends on the operating system — so if you are running Mac OS X you will have a different JVM than if you are running Windows or some other operating system. This fact can be verified by trying to download the JVM for your particular machine — when trying to download it, you will be given a list of JVMs corresponding to different operating systems, and you will obviously pick whichever JVM is targeted for the operating system that you are running. So we can conclude that JVM is platform-dependent and it is the reason why Java is able to become "Platform Independent".

Important Points:

- In the case of Java, it is the magic of Bytecode that makes it platform independent.

- This adds to an important feature in the JAVA language termed as portability. Every system has its own JVM which gets installed automatically when the jdk software is installed. For every operating system a separate JVM is available which is capable of reading the .class file or byte code.

- An important point to be noted is that while JAVA is platform-independent language, the JVM is platform-dependent. Different JVM is designed for different OS and byte code is able to run on different OS.

END

cantileverlabs.in

PLACEMENT
PREPARATION
TOOL KIT