Java Buzzwords

Wednesday, August 19, 2020 10:54 PM

Features of Java (Java Buzzwords)

Simple: Java language is simple to use and easy to learn. It manages to handle all of its concepts in quite flexible manner. If the programmer is already aware of object oriented concepts, it becomes even easier. Moreover, it extracts all most all the features of C/C++. Java was designed to be easy for the professional programmer to learn and use effectively.

Secure: All the Java that provides the user is nothing but secured programming techniques. Java implements a separate Security Manager so that the user can be benefited in implementing the objects with ease of use. Java is intended to be used in networked/distributed environments. Toward that end, a lot of emphasis has been placed on security. Java enables the construction of virus-free, tamper-free systems

Robust: To provide better reliability, Java has to implement applications on variety of platforms. Hence it requires being robust language. To do so, It has to concentrate on few areas like identifying the errors i.e., error handling & memory management. In fact, Java doesn't allow you to make any mistakes. As Java is a strictly

typed language, it checks your code at compile time. However, it also checks your code at run time.

Strongly typed: Often, saying Java is a strongly typed language is absolute because it is very much particular about the type of the data. The user needs to be careful while dealing with data types.

Portability: As Java generates a byte codes (class file) as intermediate files, it can be ported to any platform without any problem. This itself is one of the major advantages of Java.

Architecture neutral: A central issue for the Java designers was that of code longevity and portability. One of the main problems of a programmer is that no guarantee exists that if you write a program today, it will run tomorrow – even on the same machine. As Java runs on JVM, this problem may not arise. Hence you can assume that Java is architecture – neutral language. The main goal was "write once; run anywhere, anytime, forever." To a great extent, this goal was accomplished.

Object oriented: Object oriented programming was well proven technique with all of its concepts like polymorphism, inheritance. In future, it may get even more potentiality in software development. Hence implementing such concept will be an added advantage of Java. Moreover, the learner may not be in the illusion that he is learning a new language. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non --objects.

Dynamic: Dynamic nature of Java gives more comfortness to the designer because dynamic declaration & redeclaration of data members becomes easy at runtime. This makes it possible to dynamically link the code in a safe manner.

Distributed: Java is designed for distributed environments like Internet, because it handles TCP/IP protocols. With this nature Java objects are distributed over the network and get executed remotely on demand.

Multithreaded: Java has another advantage of allowing the user to develop interactive, networked programs. To achieve this, Java supports multithreading this allows you to run many tasks simultaneously. Java provides built-in support for multithreading so that the user can design such application in a most sophisticated way.

Interpreted & High Performance: Java enables the creation of cross-platform programs by compiling into an intermediate representation called Java bytecode. This code can be interpreted on any system that provides a Java Virtual Machine. the Java bytecode was carefully designed so that it would be easy to translate directly into native machine code for very high performance by using a just-in-time compiler. Java run-time systems that provide this feature lose none of the benefits of the platform-independent code.