

## Java Features/Buzz words

### - Java Features or Java Buzzwords

Following are the features or buzzwords of Java language which made it popular:

- 1.Simple
- 2.Secure
- 3.Portable
- 4.Object-Oriented
- 5.Robust
- 6.Multithreaded
- 7.Architecture neutral
- 8.Interpreted
- 9.High Performance
- 10.Distributed
- 11.Dynamic

#### Simple:

Java easy to learn

It is easy to write programs using Java

Expressiveness is more in Java.

Most of the complex or confusing features in C++ are removed in Java like pointers etc..

#### Secure:

Java provides data security through encapsulation.

Also we can write applets in Java which provides security.

An applet is a small program which can be downloaded from one computer to another computer automatically.

There is no need to worry about applets accessing the system resources which may compromise security.

Applets are run within the JVM which protects from unauthorized or illegal access to system resources.

#### Portable:

Applications written using Java are portable in the sense that they can be executed on any kind of computer containing any CPU or any operating system.

When an application written in Java is compiled, it generates an intermediate code file called as bytecode.

Bytecode helps Java to achieve portability.

This bytecode can be taken to any computer and executed directly.

Object - Oriented:

Java follows object oriented model.

So, it supports all the features of object oriented model like:

Encapsulation

Inheritance

Polymorphism

Abstraction

Robust:

A program or an application is said to be robust(reliable) when it is able to give some response in any kind of context.

Java's features help to make the programs robust. Some of those features are:

1. Type checking
2. Exception handling

Multithreaded:

Java supports multithreading which is not supported by C and C++.

A thread is a light weight process.

Multithreading increases CPU efficiency.

A program can be divided into several threads and each thread can be executed concurrently or in parallel with the other threads.

Real world example for multithreading is computer. While we are listening to music, at the same time we can write in a word document or play a game.

Architecture - Neutral:

Bytecode helps Java to achieve portability.

Bytecode can be executed on computers having any kind of operating system or any kind of CPU.

Since Java applications can run on any kind of CPU, Java is architecture neutral.

Interpreted and High Performance:

In Java 1.0 version there is an interpreter for executing the bytecode. As interpreter is quite slow when compared to a compiler, java programs used to execute slowly.

After Java 1.0 version the interpreter was replaced with JIT(Just-In-Time) compiler.

JIT compiler converts the byte code into machine code piece by piece and caches them for future use.

This enhances the program performance means it executes rapidly.

Distributed:

Java supports distributed computation using Remote Method Invocation (RMI) concept.

The server and client(s) can communicate with another and the computations can be divided

among several computers which makes the programs to execute rapidly.  
In distributed systems, resources are shared.

Dynamic:

The Java Virtual Machine(JVM) maintains a lot of runtime information about the program and the objects in the program.

Libraries are dynamically linked during runtime.

So, even if you make dynamic changes to pieces of code, the program is not effected.