JAVA Buzzwords / Features of JAVA

1. **Simple:** Simple when compared with C and C++, because most of the complex and confusing features like pointers, multiple inheritance, structure and many other concepts are removed in java. Also, to learn java no prior knowledge is required.
2. **Object oriented programming:** Java is purely an object-oriented language **due to the absence of global scope, everything in java is an object, all the program codes and data resides within classes and objects**. It comes with an extensive set of classes, arranged in packages, object model in java in sample and easy to extend. There are seven qualities to be satisfied for a programming language to be pure Object Oriented. They are:
3. Encapsulation/Data Hiding
4. Inheritance
5. Polymorphism
6. Abstraction
7. All predefined types are objects
8. All user defined types are objects
9. All operations performed on objects must be only through methods exposed at the objects.

**Why Java is not a Pure Object-Oriented Language?**

1. **Primitive Data Type ex. int, long, bool, float, char, etc. as Objects.**
2. **The static keyword:**When we declare a class as static then it can be used without the use of an object in Java. If we are using static function or static variable then we can’t call that function or variable by using dot(.) or class object defying object-oriented feature.
3. **Platform Independent:** In case of C or C++ (language 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”.

1. **Portable:** We can run the same java byte code on any platform (Windows or Linux) without making any changes in java byte code (java program).
2. **Architecture-Neutral:** The compiler generates an architecture-neutral object file format. The compiled code is executable on many processors, given the presence of the Java runtime system. The Java compiler does this by generating bytecode instructions which have nothing to do with a **particular computer architecture**. Rather, they are designed to be both easy to interpret on any machine and easy to translate into native machine code on the fly.
3. **Secure:** java is secure due to the following reasons:

* Java programs run inside a virtual machine which is known as a sandbox.
* Java does not support explicit pointer.
* Byte-code verifier checks the code fragments for illegal code that can violate access right to object.
* Run-time security check takes place when we load new code.

1. **Multithreaded:** Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.
2. **Java is ROBUST:** Java is Robust because it is highly supported language. It is portable across many Operating systems. Java also has feature of Automatic memory management and garbage collection. Strong type checking mechanism of Java also helps in making Java Robust. Bugs, especially system crashing bugs, are very rare in Java.  First Java is highly supported language. Compiler checks the program that whether there are any errors or not and interpreter ascertains about the run time error and makes the system secure from crash. Secondly, java is portable across multiple platforms which are strongly being supported by Java Virtual Machine. Also, java is provided with another important feature of automatic garbage collection & strong memory allocation. Other factors which make java a highly robust language include that it facilitates the programmers with efficacious type checking & exception handling mechanism as compared to other languages.
3. **Java is Distributed:** Java is also a distributed language. Programs can be designed to run on computer networks. Java has a special class library for communicating using TCP/IP protocols. Creating network connections is very much easy in Java as compared to C/C++.
4. **Compiled and Interpreted:** Java is both compiled and interpreted programming language. First java compiler compiles java code and generates machine independent Byte code.

At runtime JVM interprets this byte code into machine specific code and executes that machine code.

1. **High Performance:** Initially on any computer JVM converts the byte code line by line into the machine language and whichever line that gets converted into machine language computer executes it, this way of running line by line execution is called interpretation this is not a best approach. There is serious problem with performance in this approach so they introduced a JIT (Just in time compilation approach).

**What is JIT?**

* 1. Some portion of byte code are very critical – computer keeps on executing these byte code instructions repeatedly again and again for most of its execution time. So, the JVM should convert bytecode of all such critical portion into the machine code for just once and used the already converted code to next time, that saves much time and increase the performance.