JAVA

Features of java:

* **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 machine, rather into platform independent byte code. 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 to develop 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 Java runtime system.
* **Portable:** Being architecture-neutral and having no implementation dependent aspects of the specification makes Java portable. Compiler in Java is 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.
* **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 light-weight process.
* **High Performance:** With the use of Just-In-Time compilers, Java enables high performance.

Object:

Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behavior such as wagging their tail, barking, eating. An object is an instance of a class.

Class:

A class can be defined as a template/blueprint that describes the behavior/state that the object of its type supports.

Methods:

A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.

Instance Variables:

Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.

**Unit Testing**

A unit test is a way of testing a unit - the smallest piece of code that can be logically isolated in a system.

**assertEquals** - Asserts that two objects are equal.

**assertFalse** - Asserts that the condition is false.

**assertNull** - assertNull checks if the object is null

**assertArrayEquals** - assertArrayEquals asserts that two object arrays are equal.

**Parameterised test** - To execute the same test over and over again using different values.

**Assertion Error** - error is thrown if the test fails