Java Introduction

- Java is a programming language originally developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform.
- The language derives much of its syntax from C and C++. Java applications are typically compiled to bytecode (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture.
- Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere."
- Java is currently one of the most popular programming languages in use, particularly for client server web applications.
- Java Features
- Simple: - Java was developed by taking the best points from other programming languages, primarily C and C++. Java therefore utilizes algorithms and methodologies that are already proven. Error prone tasks such as pointers and memory management have either been eliminated or are handled by the Java environment automatically rather than by the programmer. Since Java is primarily a derivative of C++ which most programmers are conversant with, it implies that Java has a familiar feel, rendering it easy to use.
- 1. Object oriented: - Java is a fully Object-oriented programming language. & Object oriented concepts are following.
    - Class
    - Object
    - Inheritance
    - Encapsulation
    - Polymorphism.
- 2. Portable:- The feature Write-once-run-anywhere makes the java language portable provided that the system must have an interpreter for the JVM. Java also has the standard data size irrespective of the operating system or the processor. This feature makes Java a portable language.
- 3. Distributed:- The widely used protocols like HTTP and FTP are developed in java. Internet programmers can call functions on these

protocols and can access the files from any remote machine on the internet rather than writing codes on their local system.

- 4. High performance:- Java uses native code usage, and lightweight processes called threads. In the beginning interpretation of bytecode resulted in slow performance but the advanced version of JVM uses the adaptive and just in time compilation technique that improves the performance.
- 5. Multithreaded:- As we all know several features of Java like Secure, Robust, Portable, dynamic etc; you will be more delighted to know another feature of Java which is Multithreaded.
- Java is also a Multi-threaded programming language. Multithreading means a single program having different threads executing independently at the same time. Multiple threads execute instructions according to the program code in a process or a program. Multithreading works the similar way as multiple processes run on one computer. Multithreading programming is a very interesting concept in Java. In multithreaded programs not even a single thread disturbs the execution of other threads. Threads are obtained from the pool of available ready to run threads and they run on the system CPUs.
- 6. Robust:- Java has a strong memory allocation and automatic garbage collection mechanism. It provides a powerful exception handling and type checking mechanism as compared to other programming languages. Compiler checks the program whether there is any error and interpreter checks any run time error and makes the system secure from crash. All of the above features make the java language robust.
- 7. Dynamic:- While executing the java program the user can get the required files dynamically from a local drive or from a computer thousands of miles away from the user just by connecting with the Internet.
- 8. Secure:- Java does not use memory pointers explicitly. All the programs in java are run under an area known as the sand box. Security manager determines the accessibility options of a class like reading and writing a file to the local disk. Java uses the public key encryption system to allow the java applications to transmit over the internet in the secure encrypted form. The bytecode Verifier checks the classes after loading.