Introduction to Java for C++ Programmers

Segment - 1

JAC 444

By: Mahboob Ali

Objectives

Upon completion of this segment you should be able to:

- Characteristics of Java.
- Understand Java Technology.
- Use java platforms.
- Know basic elements of java programming language.

What is Java?

- Java can be broadly defined as
  - A General Purpose → Its not constraint to one particular domain
  - Object Oriented → Helps model real world scenarios in more natural way
  - Platform Independent → Write Once Run Anywhere (WORA)
  - Concurrent ~ Multithreading
  - Very fast

Why Java?

- Java is full-featured, general purpose programming language that can be used to develop robust mission-critical applications.
- It was used to develop the code to communicate with and control the robotic rover on Mars.
- Initially java became attractive because java programs can run from a web browser, called *applets*. *(*Today applets are no longer allowed to run via browser due to security reasons.)
- Java is the Internet Programming Language.
  - The main reason why java got its attention in the business community.

Java is a versatile programming language, used to develop application for:

- Desktops.
- Servers.
- Small handheld devices.
- Android cell phones is developed using java.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is partially modeled on C++, but greatly simplified and improved. Some people refer to Java as "C++--" because it is like C++ but with more functionality and fewer negative aspects. Automatic garbage collection and memory allocation.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java is inherently object-oriented. Although many object-oriented languages began strictly as procedural languages, Java was designed from the start to be object-oriented. Object-oriented programming (OOP) is a popular programming approach that is replacing traditional procedural programming techniques.

One of the central issues in software development is how to reuse code. Object-oriented programming provides great flexibility, modularity, clarity, and reusability through encapsulation, inheritance, and polymorphism.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Distributed computing involves several computers working together on a network. Java is designed to make distributed computing easy. Since networking capability is inherently integrated into Java, writing network programs is like sending and receiving data to and from a file.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

You need an interpreter to run Java programs. The programs are compiled into the Java Virtual Machine code called bytecode. The bytecode is machine-independent and can run on any machine that has a Java interpreter, which is part of the Java Virtual Machine (JVM).

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java compilers can detect many problems that would first show up at execution time in other languages.

Java has eliminated certain types of error-prone programming constructs found in other languages.

Java has a runtime exception-handling feature to provide programming support for robustness.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java implements several security mechanisms to protect your system against harm caused by stray programs.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Write once, run anywhere

With a Java Virtual Machine (JVM), you can write one program that will run on any platform.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Because Java is architecture neutral, Java programs are portable. They can be run on any platform without being recompiled.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java’s performance is sometimes criticized. The execution of the bytecode is never as fast as it would be with a compiled language, such as C++. Because Java is interpreted, the bytecode is not directly executed by the system, but is run through the interpreter.

New JVM is significantly faster. The new JVM uses the technology known as just-in-time compilation.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Multithread programming is smoothly integrated in Java, whereas in other languages you have to call procedures specific to the operating system to enable multithreading.

Characteristics of Java

- Java Is Simple
- Java Is Object-Oriented
- Java Is Distributed
- Java Is Interpreted
- Java Is Robust
- Java Is Secure
- Java Is Architecture-Neutral
- Java Is Portable
- Java's Performance
- Java Is Multithreaded
- Java Is Dynamic

Java was designed to adapt to an evolving environment. New code can be loaded on the fly without recompilation. There is no need for developers to create, and for users to install, major new software versions. New features can be incorporated transparently as needed.

Java Editions

- Java Standard Edition (J2SE):

  to develop client-side applications, run on desktops.

- Java Enterprise Edition (J2EE):

  to develop server-side applications, like Java servlets, Java Server Pages (JSP), and Java Server Faces (JSF)

- Java Micro Edition (J2ME):

  to develop applications for mobile devices.

- We are going to use J2SE to Introduce Java Programming

Understanding Java Technology

Java technology is defined by two elements:

- Java platform
- Java programming language

Java Platform

- Java platform has two components:
  - Java Virtual Machine (JVM).
  - Java Application Programming Interface (API)

Java Virtual Machine

- Technical Definition: an engines that drives the java code.
- Everyday Definition: JVM is how we run our java programs.
- JVM manages system memory
- JVM provides a portable execution environment for java-based applications
- Mostly in other Programming Languages, compiler produce code for a particular system but Java compiler produce Bytecode for a Java Virtual Machine.
- Bytecode is an intermediary language between Java source and the host system.

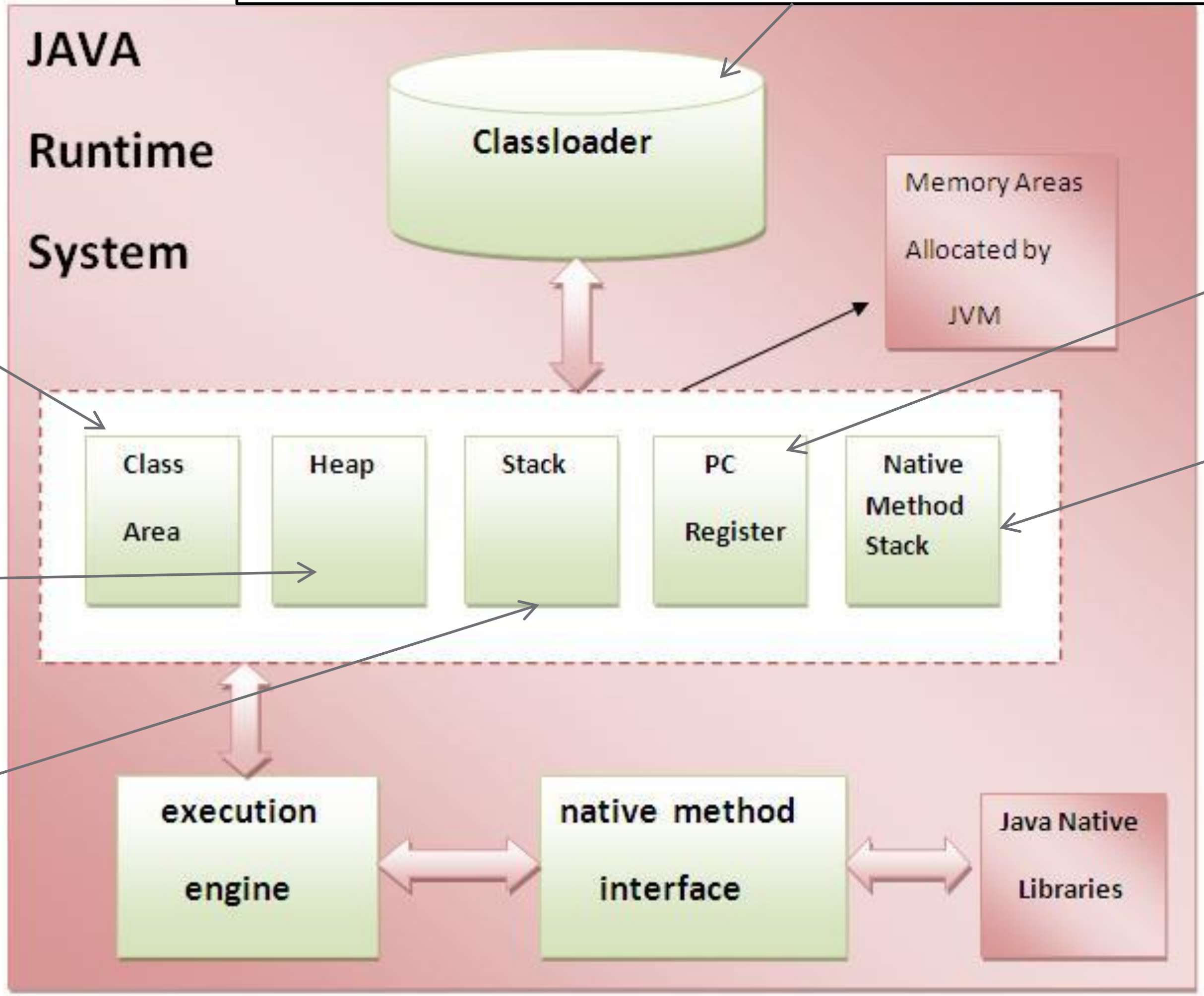
Internal Architecture of Java

Subsystem of JVM that is used to load class files

Class(Method) Area stores per-class structures. Such as runtime constant pool.

It is the runtime data area in which objects are allocated.

Java Stack stores frames. Local variables, partial results, thread information, etc.



Contains the address of current instruction

It contains all the native methods used in the application.

Java Application Program Interface

- Java API also known as *library*, contains predefined:
  - Classes
  - Interfaces

  These libraries are known in Java as Packages, and still expanding.

Execution of Java Program

- Normally there are five phases
  - edit
  - compile
  - load
  - verify
  - execute.

Steps to Use Java SE

- **Coding:** produced by the programmer.
- **Compiling:** build the program into bytecode
  - Result is a ".class" file.
  - Compiler command: javac
- **Running:** class file (bytecode) is loaded by the JVM.
  - JVM command: java
- **Bytecode Verification:** The JVM verifies the class file digital signature. The JVM is simply an interpreter.
- **Internal Integrity Check:** It checks if loaded Java program is well formed. Data types are verified along with other syntax structure.
- **Execution:** Program execution begins from the main entry point.

Popular Java IDEs (Integrated Development Environment)

- Net Beans Open Source by Oracle
- Eclipse Open Source by IBM
- IntelliJ IDEA