Java Fundamentals

The Origins of Java

Java was conceived by James Gosling, Patrick Naughton, Chris Warth, Ed Frank, and Mike Sheridan at Sun Microsystems in 1991. The initial name of the language was "Oak" but was changed to "Java" in 1995. The development of Java was projected at providing a platform-independent language that could produce code that would execute on any processor. After the advent of Internet and World Wide Web Java started to become popular due to its platform independent feature**.**

How Java relates to C, C++

Java is closely related to both C and C++. Any programmer who is familiar with C or C++ can easily learn Java since it uses the same syntax that C++ uses. Although Java was influenced by C++, it is the neither enhanced version of C++ nor designed to replace C++. Java and C++ were designed for two different set of problems.

How Java relates to C#

Both Java and C# share the same general C++ syntax, support distributed programming and also utilize same object model. Any programmer who is familiar with C#, then learning Java will be easy and vice versa. Though there are some differences between Java and C#, the overall feel of these languages is very similar.

Java's Contribution to the Internet

Java Applets

- It is an application designed to be transmitted over the Internet and executed by java compatible web browser. An Applet is actually a tiny Java program, dynamically downloaded across the Internet.

- Applets are typically used to display data provided by the server, handle user input, or provide simple functions that execute locally rather than on the server.

- The applet is a dynamic, self-executing program, which extends the functionality of a web browser.

- Applets are platform-independent and portable.

Security

- Applets can be safely downloaded to the user computer.
- Java achieved this safety by confining an applet which is executed independently without accessing other parts of the computer**.**

Portability

- Java programming is portable.
- Java applets can be downloaded and executed by a wide verity of different CPUs, operating systems, and browsers connected to the internet.

The Bytecode

- Bytecode is a highly optimized set of instructions designed to be executed by the Java run-time system which is called the Java Virtual Machine (JVM).
- Bytecodes are platform independent instructions, so Java's bytecodes are portable that is, the same bytecode can execute on any platform containing a JVM that understands the same Java bytecode.
- Java program is executed by the JVM which prevents it from generating side effects outside of the system and helps to make it secure.
- A just-in-time (JIT) is a code generator that converts Java bytecode into native machine code. When JIT is a part of the JVM, a selected portion of bytecode compiled into executable code in real time on a piece by piece, demand basis. JIT compiler compiles code as it is needed, during execution.

What is a servlet?

A small program that executes on a server is called a servlet which extends the functionality of a web server and provides a mechanism that Java would also be useful on the server-side. With the advent of a servlet, Java spanned both sides of the client/server connection.

Java Features

- **Simple -** Java language is simple in the sense that it is easy to learn since it contains user-friendly syntax**.**
- **Secure -** Java provides a secure means of creating Internet application.

- **Object-Oriented -** Java has all Object-oriented programming features such as abstraction, encapsulation, inheritance, and polymorphism**.**
- **Robust -** Java is strongly typed, which allows extensive compile-time error checking.
- **Multithreaded -** We can write Java programs that deal with many tasks at once by defining multiple threads.
- **Architecture neutral -** Java is a platform-independent language that means one does not have to rely on a specific machine.
- **Interpreted -** Java supports cross-platform code through the use of Java bytecode**.**
- **High performance -** Java Bytecode is more efficient and highly optimized for speed of execution.
- **Distributed** - We can create distributed applications in Java. We may access files by calling the methods from any machine on the internet.
- **Dynamic -** Java programs carry the substancial amount of run-time information.

Object-Oriented programming

Java is Object-oriented language. Object-oriented programs are organized around data, with the key principle being "data controlling access to code". It is a type of programming in which programmers define the data and sort of operations that can be applied to the data. All programming languages that support encapsulation, polymorphism and inheritance are known as object-oriented languages.

Encapsulation

- Encapsulation in Java is a process of wrapping code and data together into a single unit and keep both code and data safe from outside interference and misuse. When code and data are linked together an object is created which supports encapsulation.
- Encapsulation is the technique in which code or data may be private to the object or public. If code or data is private, it cannot be accessed by anyone outside the class. When code or data is public, another part of a program can access it.
- Java's basic unit of encapsulation is the class. Java uses class

specifications to construct objects which are instances of a class. Thus, a class is essentially a set of plans that specify how to build an object.

Polymorphism

- Polymorphism means many forms. Polymorphism in Java is a concept by which we can perform a single action by different ways. Thus, polymorphism is the ability of different types of related objects to respond to the same message in their own ways.

- It allows different objects to share the same external interface although the implementations may be different. Polymorphism helps reduce complexity by allowing the same interface to be used to specify a general class of action.

Inheritance

- Inheritance is the process by which one object can acquire the properties of another object. Inheritance allows the creation of hierarchical classification in which first a broad category/parent is formed and then it's sub-categories are formed.

- These sub-classes inherit all the attributes and the properties from their parent class. Using inheritance, an object need only define those qualities that make it unique within its class.

Java Development Kit (JDK)

To compile and run Java programs you must have Java Development kit. The JDK supplies two primary programs. The first is javac, which is the Java compiler and the second is java, which is the standard Java interpreter. JDK runs in the command prompt environment and uses command-line tools.