JAVA AND LISP COMPARISION

LISP

The name LISP derives from "LISt Processing". Lisp was originally created as a practical mathematical notation for computer programs, influenced by the notation of Alonzo Church's lambda calculus. It quickly became the favoured programming language for artificial intelligence (AI) research. As one of the earliest programming languages, Lisp pioneered many ideas in computer science, including tree data structures, automatic storage management, dynamic typing, conditionals, higher-order functions, recursion, and the self-hosting compiler.

Most Lisp compilers are not Just In Time compilers. You as a programmer can invoke the compiler, for example in Common Lisp with the function COMPILE-FILE. compile-file calls the compiler to translate a Lisp source file into a binary data form that both loads and runs faster. A compiled function typically runs more than ten times faster than when interpreted (assuming that it is not spending most of its work calling already compiled functions). A source file with a .lisp or .lsp extension compiles to produce a file with a .*x*fasl extension (the actual extension depends on the host machine CPU). Subsequent use of load loads the compiled version (which is in LispWorks's FASL or Fast Load format) in preference to the source.In compiling a file the compiler has to both compile each function and top level form in the file, and to produce the appropriate FASL directives so that loading has the desired effect. In particular objects need to have space allocated for them, and top level forms are called as they are loaded.

IDE Used : Allegro cl

JAVA

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented,[12] and specifically designed to have

as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

A just-in-time (JIT) compiler is a program that turns Java bytecode (a program that contains instructions that must be interpreted) into instructions that can be sent directly to the processor. After a Java program is written, the source language statements are compiled by the Java compiler into bytecode rather than into code that contains instructions that match a particular hardware platform's processor (for example, an Intel Pentium microprocessor or an IBM System/390 processor). The bytecode is platform-independent code that can be sent to any platform and run on that platform.The just-in-time compiler comes with the virtual machine and is used optionally. It compiles the bytecode into platform-specific executable code that is immediately executed. Sun Microsystems suggests that it's usually faster to select the JIT compiler option, especially if the method executable is repeatedly reused.

IDE Used : Eclipse