

Software's New Brew: Java

By Ernie Longmire

Beverage fans were brought up short a few years ago when the word "java" suddenly took on a completely new meaning. Once merely a slang term for coffee, people using the word in conversation nowadays are likely to find their audience assuming that they're talking about a computer programming language.



But the loss to the coffee-drinking beatniks of the world is made up by everyone else's gain. The Java programming language, developed by Sun Microsystems, makes possible a new variety of applications that would have been difficult or impossible to create in the pre-Java computing world. And the adoption of Java applications is surging. According to the International Data Corporation, in the next three years, the size of the Java market is expected to exceed \$2 billion, with some 1.5 million programmers working in the language.

Many programmers feel like they are in familiar territory from the first sight of Java source code. Though it was developed separately, Java bears some strong similarities to C and C++, the powerful languages at the heart of the UNIX operating system (and the heart of BASIS' product line). The structure and syntax of Java code is remarkably similar to both languages, and like C++, Java is object-oriented.

Java's differences from those languages, however, are at the heart of its advantages over them. For example, unlike C and C++, Java programs are not compiled to the native machine code of the machine on which they will run but rather to a platform-independent "bytecode" format. The bytecode is then executed by a copy of the Java Virtual Machine (JVM), a special interpreter that has been ported to the local hardware and operating system. In this respect, the JVM is similar to Business Basic interpreters such as BBxPROGRESSION/4® and PRO/5®; you only need to write a program once, after which you can run it on any machine to which the language's interpreter has been ported.

Thanks to the JVM concept, Java is a deeply portable language: any device that has the JVM running on it can run Java applications. That device need not be a computer in the conventional sense. For instance, Java was originally developed as a language for consumer electronics devices such as fancy cable TV boxes. The restrictions inherent in this environment resulted in a language design that was modular and compact enough to be embedded just about anywhere, from a Web browser to a customized Java microchip.

Java has a number of other advantages over C and C++ that make Java programs easier to develop and maintain. For example, Java is a strongly typed language, meaning that programmers have to be particularly careful in their use of the data types (such as characters and integers) that are built into the language. This makes Java less flexible but also less prone to programmer mistakes. Similarly, Java does not offer any way for the programmer to directly access specific locations in a computer's memory: a facility that is extremely powerful but that most C programmers will acknowledge allows the introduction of some of the hardest bugs they've ever had to track down.

Java's portability and robustness make it one of the most interesting new languages of the last decade. Software developers owe it to themselves to take a look at what Java might have to offer them.

A Short Java Glossary

Applets are, as the name implies, small Java applications. They're usually designed to perform simple, specialized tasks and can run well in restricted environments such as Web browsers.

Beans are small, standalone pieces of Java code that are designed to support the JavaBeans component object model. This model defines a standardized, portable way for components to interact with applications and with each other.

Java **bytecode** is the compiled version of a Java program. Java bytecode is platform independent and is interpreted by the Java Virtual Machine; it is similar to the tokenized version of a BBx® program.

The **Java Virtual Machine (JVM)** is the "virtual computer" that makes Java a truly portable language.

JavaScript is a scripting language for Web browsers. JavaScript isn't related to Java outside of its name, which was chosen for marketing reasons.

A **Just-In-Time (JIT) compiler** compiles Java bytecode into the local machine language and immediately executes it. This is a different process than the usual Java interpretation process and can make code run faster but at the cost of some additional processing time up front.

Object-oriented programming is a software design approach that emphasizes the relationship between data and operations on that data. Because data is very closely tied to the things that can be done to it, object-oriented programs can be easier to maintain and extend.