A Short History of Java

The story of Java can be traced to early 1991 when a team of engineers led by James Gosling, at Sun Microsystems was formed to explore opportunities in the consumer electronics market. This effort was eventually referred to as the "Green" project. The aim of the project was to build a distributed system that would allow consumer electronic devices to talk to each other in a heterogeneous network as a business product that they can sell as modern software technology to consumer electronics' manufacturers.

The engineers of the Green project started with C++ as their choice of language. However, in order to accommodate the consumer market's demand for CPU flexibility, they wanted to make the development a more platform-neutral process, and this required extending the C++ compiler. Eventually, however, they realized that even with lots of extras, C++ would not suffice, not to mention that C++ is a complex language in its own and current form. This has lead to the development of a new programming language – Oak – named after a tree that James Gosling could see from his office window. A key commercial use of Oak was to be the programming language for interactive cable TV control devices and electronic home management systems. After months of hacking on Oak, the Green team produced an operating system, a toolkit, and interface, a new hardware platform, three custom chips, and they delivered a PDA-like device called "*7" that Gosling calls a "handheld remote control".

In 1993, the Green team (incorporated as FirstPerson Inc.) got a request for a proposal for a set-top box operating system and video-on-demand technology. FirstPerson placed a pid, but despite being told that they have the best technology, they did not win the pid (SGI did) due to what Gosling characterizes as wholly non-technical reasons, such as business politics. FirstPerson kept trying to pursue set-top boxes until 1994, when it concluded that *the market was not real*. FirstPerson dissolved and about half of its staff moved to Sun Interactive to digital video data servers.

By mid-1994, the Internet in general and the World Wide Web (WWW) in particular exploded into widespread use. The Green team decided on adapting the Oak language to the needs of the Internet and provide an Internet programming language for WWW-based applications. The decision was made because Oak included many features that were relevant to the WWW environment, including the idea of being *architecturally neutral* and *platform independent* (meaning that the same program can run on a wide variety of platforms).

In January 1995, Oak was renamed Java (as the name Oak was already in use), and developed into a robust programming language for building WWW-based applications. As a proof-of-concept, they developed a web-browser known as HotJava (written in Java) that is capable of running mini Java applications (known as Applets) that are embedded in WWW pages. This feature of HotJava was then taken up by Netscape and Microsoft. Later in 1995 Sun released the initial Java Development Kit (JDK) and HotJava. From then on Java has become increasingly popular.

So, what is Java?

In the Java whitepaper (available for Sun's website http://java.sun.com), Sun describes Java as follows: *Java is a simple, object-oriented, distributed, interpreted, robust, secure, architectural neutral, portable, high-performance, multithreaded, and dynamic language*.

While this seems like a string of buzzwords, but the fact is that these buzzwords actually describe the language and its features. To get a feeling of why Java is important and interesting, let's look at the features behind some of these buzzwords.

Java is Simple

One of the design goals of Java was to make it much easier to write bug-free code. In order to help programmers with this, the language has to be simple. The simplicity of Java makes it fun to program with, and its programs are easy to write and read.

If you have programmed in C or C++, you would know that half of the bugs in your programs are related to memory allocation. With Java you will not have this problem as the Java runtime environment provides automatic memory allocation and garbage collection.

Java is Object-Oriented

Object-Oriented (OO) programming was the catch phrase of the 1990's. As a marketing strategy, many companies claim that their software is object-oriented, when in fact they are not. An earlier computer scientist (S. King?) claimed that if someone wanted to sell his cat, he should not say it is clean,

nice, beautiful….etc but rather *it is object-oriented*. The next time you go buy a toaster, make sure it is object-oriented! ☺
You will do OO programming this term and more of it next term….

Java is Distributed
As I mentioned earlier, the aim of the Green project was to build a distributed system that would allow all consumer electronic devices to talk to one another. Since this was a design goal, Java provides a lot of high-level support for networking. You will see more of this next year! ☺

Java is Interpreted
Java is an interpreted language. This means that Java programs are not compiled into machine platform-dependant language. But rather they are compiled into byte-codes for the Java Virtual Machine (JVM). To run Java programs, you use the Java interpreter to run the Java byte-codes. Java byte-codes are platform-independent, which means they can run on any platform with a Java interpreter. One catch of interpreted code is that it is a bit slower than machine code when it runs. However, with all the Java optimization techniques and Just In-Time Compilers technology (JIT), Java byte-codes will run as fast as C or C++ compiled code.

Java is Robust
Buggy software can be written in any language, and Java is no exception. However, Java eliminates certain types of programming errors and that makes it easier to write reliable software. Java is a strongly typed language and that allows for extensive compile-time checking. Also, the fact that Java does not have pointers eliminates another class of memory-related bugs.

Where is Java Today?

Java has grown into a full-scale development system that is being used for building large applications that exist outside of the WWW environment. Currently, there are three flavors of the Java platform: Java 2 Standard Edition (J2SE), Java 2 Enterprise Edition (J2EE), and Java 2 Micro Edition (J2ME). The J2SE is being used to develop applications for the well-established desktop market; the J2EE is being used for enterprise applications; and the J2ME is being used for wireless applications (applications that run on handheld wireless devices such as cellular phones and personal digital assistants).

Java Applications

Sample applications Java will be demonstrated in class on Thursday, October 17. In this course, we will be using J2SE….