



01 Jan 1998

Java, Java, Java

Chaman Sabharwal

Missouri University of Science and Technology, chaman@mst.edu

Follow this and additional works at: https://scholarsmine.mst.edu/comsci_facwork

 Part of the [Computer Sciences Commons](#)

Recommended Citation

C. Sabharwal, "Java, Java, Java," *IEEE Potentials*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1998.

The definitive version is available at <https://doi.org/10.1109/45.714612>

This Article - Journal is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Computer Science Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Everyone is talking about Java. The driving force for this chatter is Java's simplicity and potential power. Java is both a programming language, and a language for the intranet and the World Wide Web (WWW). Java is a Write On one platform and Run on Many platforms (WORM) language. For network-friendly, platform-independent applications, Java (originally Oak, see box) is an object-oriented programming language. Java source code is compiled into a virtual machine code or bytecode. This makes the Java platform independent. It can be placed on a web site, and executed on the client side on a PC—Intel, Mac—Motorola or UNIX—Solaris machine without recompiling.

Sun Microsystems formally announced Java in May 1995. Java is the first language that has built-in capabilities for networking applications, in particular, creating dynamic web pages. Java programs that run on the Internet web pages are called applets. These applets are easily incorporated into Web pages. These applets are executed from homepages on remote web sites. Java programs run interpretively on the client side.

Java reduces development costs and speeds up the learning curve. Traditional client-server development tools—such as Delphi, Power-Builder and Visual Basic—are losing ground to Java as a result. By year-end of 1996, Java had moved ahead in use of both C and C++ and these application development tools.

Java, Internet, Intranets and the WWW

Information found on the Internet is usually public domain, commercial for production and static in context. Java can be incorporated in WWW pages and can be executed without modification as you connect to the web site. Java's design is based on the concept of sharing information across the networks, not hiding information.

Today, it is not practical to produce full-blown network applications because of the low communication bandwidth. For the time being, Java is used to add functionality to network applications. Applets included in WWW pages introduce new levels of multimedia functionality through the use of text, graphics, images and audio.

For example, a tickertape applet dis-

plays information as a scrolling text. The repaint method is embedded in a thread that updates the value of the x-coordinates. This invokes the paint method for displaying the scrolling text:

```
public void paint(Graphics g)
{
    g.drawString("Test Times: Time is Money" + new Date(), x, 36);
}
```

The scrolling text is dynamically updated showing current activity (time or stock prices) of the day (Fig. 1).

Intranets are internal organizational networks set up to connect management. They connect computers to people, people to people, and people to information. Information found on the intranet is protected, proprietary, private and security-sensitive for development activity and dynamically updated. Java makes intranets practical in the academic and business environment. Intranets are embedded in the Internet.

Intranets support two objectives: 1) increased information sharing within certain boundaries, and 2) cost effective management. This is done by storing information on a server on the Internet.

Common protocols are used to link computers to provide client/server communication over the Internet. An intranet uses the same common protocols for communication, but it is limited to a select group of people or a company. A group of people may have similar skills and be working on the related parts of a problem. The collaborating people can access the current copy of documents simultaneously. However, the communication channels may be different. The speed of access is better

within a company because fewer people are accessing the information. On the intranet, information can be shared without any concern for the platform or geographic location.

With proper access privileges, one can update the information in

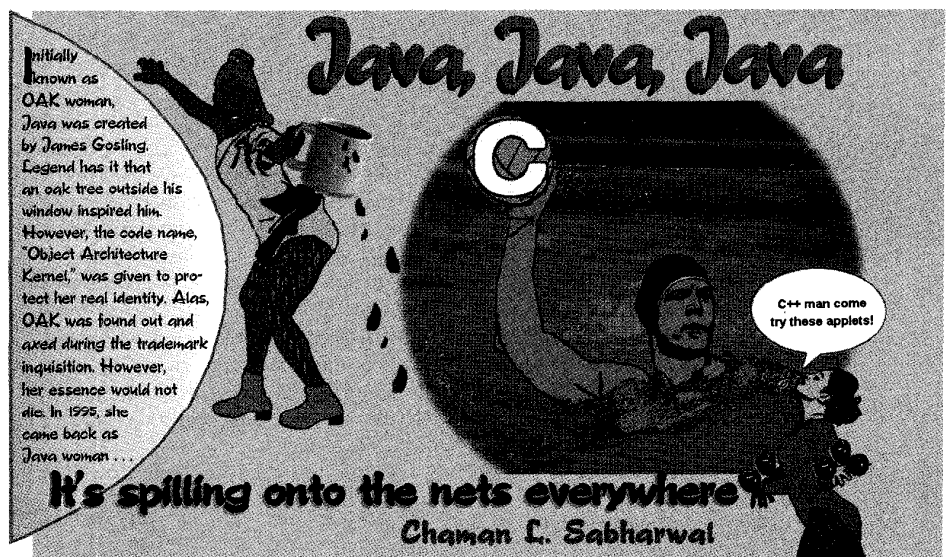
real time. This makes the latest information "current" across the Internet. It's done by making the web site accessible from any platform. Using telecommunication, or other dedicated communications lines anywhere in the world, one can access the intranet or web site and treat it "as the Network" on the Internet.

Java and HTML

Java applets are designed to be run through a web browser. In the HyperText Markup Language (HTML) file associated with the applet bytecode, the minimum requirements needing definitions are code, width and height for the applet display window.

The applets can be parameterized so that the user can personalize them for his or her own use. There are additional useful attributes depending on the nature of the applet. The full list of attributes include: align, alt, archive, codebase, hspace, vspace and name. In addition, param is used to customize the applets which require commandline input. Codebase is required when the.html file and.class file are in different directories. The name attribute is useful when an applet references another applet via the getApplet() method (see Box A pg 34).

Note: There should be no confusion ©



between the Java language and the Java script. The Java script is different in syntax and semantics from the Java language. The Java script is Netscape's interpreted language for programming the Web pages. The two languages have different objectives. The Java script is a common gateway interface (CGI) scripting language. The Java language is an application/applet programming language. However, there are plans to make the Java script more like the Java language.

Java-programming language

The Java language is both a conventional and rapid prototyping language. By definition, Java is a simple, object-oriented, distributed, interpreted, robust, secure, platform independent, portable, high-performance, multi-threaded and dynamic language. (Think of a buzzword, you will find it here.) Java is also called an architecture neutral or platform independent or multi platform language.

If you use a web browser, you are a de facto Java user. Java enables browsers, Netscape or Internet Explorer, to automatically download Java applets across the network. It then runs them on the local machine relieving the remote CPU (central processing unit) of this duty.

Box A

The format of an HTML file is:

```
<applet
  code = class_file_name
  width = number_of_pixels
  height = number_of_pixels
  codebase = [URL_address_of_class_file]
  archive = [something.zip]
  hspace = [horizontal_margin_in_pixels]
  vspace = [vertical_margin_in_pixels]
  align = [display_alignment_in_the_applet_window_top_middle_or_bottom.]
  name = ["usually_class_file_name"]
  alt = ["some_html_address_for_not_JavaEnabled_browser"] >
  <param name = "name_in_source_code" value = "user_input_value">
</applet>
```

Java programs also can be true conventional standalone applications. In fact, applets are bytecode instead of text source code. There are things one program can do the others cannot including I/O and Socket connections on the network. This is entirely for security reasons. Most Java programs, however, are applets due to the popular acceptance and success of the World Wide Web. (Remember, Java programs that run on the Internet web pages are called applets.)

Java shares language structures with other languages. Every Java program is

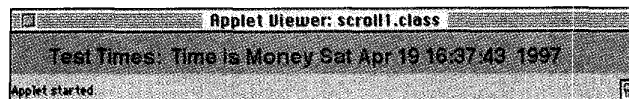


Fig. 1 Scrolling text with dynamically updating the current date



Fig. 2 Applet window and status bar display output of hello applet

composed of at least one class. All Java applets are classes and inherit (all properties and methods) from the Applet class. A Java applet can be stored as a text file using any editor.

The name of the Java source file has the extension Java. For example, the hello applet file is stored as hello.java. It is compiled by using the Java compiler with a javac command. It creates a new file with the name hello.class in the same folder as the hellojava file. Then the browser uses the hello.html file in Box B.

Now the applet is viewed by using a Java enabled browser or by an appletviewer command. Thus, the minimal program development environment consists of an editor for creating a source code, a javac compiler for compiling and an appletviewer interpreter for executing it.

```
<html> <applet code = "hello.class"
width = 400 height = 300
></applet> </html>
```

Box B

A complete example of an applet is the hello program which displays a simple message on the screen as shown in Fig. 2.

Java allows the user to display an applet response at any of the three places on the screen depending on the nature of the program's output. This example changes the default font to a 24-point Times Roman typeface that is bolded. This program is stored as hello.java. It is compiled by using a Java compiler with the command javac hello.java.

It creates a new file hello.class.

```
<html> <applet code = "hello.class"
width = 400 height = 300
></applet> </html>
```

Box C

The html file, usually denoted by the same name, is hello.html and contains what is shown in Box C.

Now the applet can be viewed by using the appletviewer command, appletviewer hello.html. The result of this command are shown in Figs. 3 and 4.

Classes

Java programming is based on class. Every Java program has at least one class. In Object Oriented Programming, the focus is on the encapsulation of data types and the operations on the data. A class embodies this concept. Java classes promote code reuse.

Java is simplified by the fact that most of the useable code is available in Java packages called libraries. Classes are grouped by categories into the packages. One strength of Java is the Application Programming Interface (API) that promotes reusability and saves programmers from reinventing efficiently tested code. The hierarchy for a particular package refers to all the classes in the package according to their inheritance. Object Oriented Programming (OOP) prevents duplication of effort in reinventing classes every time the user needs them. OOP simplifies maintenance by pushing the details as far as possible from the user to the super classes. Java classes enforce object orientation through information hiding and extensibility.

In Java, some methods are automatically invoked for the programmer such as init, start, paint, stop and destroy. In an applet, the init method is executed first. When init (which does the initializing) is completely finished, it is followed by the start method, other methods are invoked on an as-needed basis.

A final class contains static final data members which can be used by any class. A final class is never extended. In a class, static members are called class members. They expand the scope of the variable values to all objects of the class and are referenced by using the class name or object name. The static variables are shared by each instance/object of the class. An example of a class which comes with the language is the Math class which has static members PI and E, the well known mathematical numbers. The Math class does not need to be imported. These symbols are used directly as Math.PI and Math.E. The method Math.Random() returns a random number in (0, 1).

Information hiding

Information hiding is a technique which provides the minimal information needed by the user. When the user wants a method to accomplish some task, the person does not need to know how it is implemented. That is, the user needs to know only the interface and the parameter list. This way the user cannot inadvertently change anything in the method code. The methods that provide this information are called interface functions. What's more, the developer can replace the implemented code with optimized code without affecting the user.

Information hiding is also accomplished by prefixing an access type to the data members, methods and classes. The access types are *public*—for universal access by everyone; *private*—for access of members by the defining class only (i.e., the owner); *protected*—for only the subclasses and classes within the package. A class cannot be declared private because then it defeats its own purpose. It cannot be instantiated or extended. Java class extensibility is achieved through inheritance, abstraction and overloading. A class can be public, abstract, interface or final. However, it is a good practice to keep all data members private or protected.

Abstraction

Abstract classes promote encapsulation of features common to several classes. An abstract method is one whose interface appears in the super class and its implementation appears in a subclass. An abstract method must always be subclassed where it may be implemented. A final method is never overridden in any subclass.

If a class has one or more abstract methods, it is an abstract class. An abstract class must be declared by prefixing a keyword abstract. An abstract class prevents instantiation and promotes subclassing. An interface is a conceptual model. It is the next level of abstraction to abstract class. A Java interface is an abstract class in which each method is abstract, each data member is final and static. Thus, an interface is equivalent to a class, but is not a "class."

Garbage collection

Java provides a constructor to build

objects.

A garbage collector (GC) frees memory that is no longer needed by the objects. An object is marked for the garbage collector by setting it to null. The GC runs periodically to check for objects no longer needed. The programmer has no control over Java GC. The Java garbage collector is asynchronous, thus, it may not collect garbage as soon as the object is dead. This frees the programmer from allocating and de-allocating the memory. Memory management is left to the Java runtime environment. However, GC gives the object a chance to clean up after itself by calling first the finalize method. Even if the program has a direct call to the finalize method, GC will still call it before freeing its resources.

Multidimensional arrays

A multidimensional array is an array of arrays stored in row major order. In general, a two dimensional array is a rectangular array of order $m \times n$. Java arrays are an exception to this rule. Java allows unbalanced multidimensional arrays, a two dimensional array may have rows of different lengths.

Exception handling

Defensive programming saves time when something goes wrong. Without a debug environment, the programmer inserts print statements and then removes them after debugging for the production version. The irony is that when one removes all the print comments, one inadvertently introduces additional errors. Now when one detects another bug after one has removed all comments, one has to redo all the work just undone. Runtime errors may be anticipated and handled easily by using an exception handling mechanism.

Exception handling leads to safe and error-free programming. Java builds exception handling in most methods and enforces its use on the part of the devel-

oper. When Java anticipates that something could go wrong, it encapsulate such statements in a try block. If something fails in the try block, an exception object is generated. The catch statement verifies the type of exception. The exception is handled in the block of statements immediately following the catch statement. However, the programmer may decide not do anything in response to the exception if he or she wants.

Graphics and graphical user interfaces

Java has built-in capabilities of Graphics, Graphical User Interfaces (GUIs), Concurrency via multiple threads and Networking via Sockets and Uniform Resource Locators (URLs). Graphics and animation are the salient features of Java. Graphics is one of the key classes of Java. Graphical User

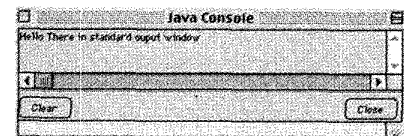


Fig. 4 Applet display on standard output device for the hello applet

Interface in Java runs without change in Windows, UNIX, and Mac—three different GUI environments. This allows the Java programmer to create multimedia applets by integrating text, graphics, images, animation and sound.

As a result, Java applets are useful for user interaction via interaction devices and user interface via GUI widgets. Java supports interaction with the keyboard and the mouse devices. Java supports GUI with labels, text fields, text areas, buttons, checkboxes, radio buttons, pop-up menus and List for interaction with the users. The following example figure has a label, an updatable textfield and a button to activate data updating and hide widgets. With a very simple program, the 3-D view is generated in Fig. 5 (pg 36). The code segment

```
import java.applet.Applet; // All Java applets must inherit from the Applet class
import java.awt.*; // For Graphics environment
public class hello extends Applet
{
    public void paint(Graphics g) // to display in the applet window
    {
        g.setFont(new Font("TimesRoman", Font.BOLD, 24)); //define new font
        g.drawString("Hello There in Applet window", 10, 15);
        showStatus("Hello There in Status Bar");
        System.out.println("Hello There in standard output window");
    }
}
```

Fig. 3 Hello applet to display a simple message on the screen

```

while (yPos < 3*winHeight/4)
{
    font = new Font("Helvetica", Font.BOLD,fontSizeValue);
    fontMetrics = g.getFontMetrics(font);
    textWidth = fontMetrics.stringWidth(text); //gives stringlength
    fontSizeValue +=20;
    yPos =fontSizeValue; // relative vertical positioning
    colorR+=15; colorG+=15-colorR; colorB-=15;
    g.setFont(font); g.setColor(new Color(colorR, colorG,colorB));
    //print in applet window
    g.drawString(text, (winWidth-textWidth)/2,yPos+3/4*winHeight/4);
}

```

for generating it is just above Fig. 5.

All display/drawing in Java is done through the Graphics object. This object provides many methods for drawing on the screen, e.g. we have already seen one method drawString(). The most common use of the Graphics object is through the paint function. The paint method is overridden in an applet.

When the browser invokes the paint function, it sends the applets' graphics context to it dependent on the local system. To make the graphics environment portable across multiple platforms, it uses the environment of the machine on which the Java applet is executed. There are many other graphics objects such as Frame, Panel and Canvas for organizing the widgets.

Web pages are enhanced by incorporating images in the applet window. Java supports two formats for image display: Joint Photographers Expert Group (JPEG) and Graphics Interchange Format (GIF). It does not support other formats including Tagged Image File Format (TIFF), Portable Network Graphics (PNG) and Picture Format (PICT). Java has a built-in uncompress utility. That is, the image is automatically decompressed without your knowledge. Downloading and displaying an image is as easy as displaying a text string.

Besides text, graphics and images, Java provides the ability to include audio to create truly multimedia applets. Playing sounds within Java is quite simple. Java supports only the Next/Sun audio (AU) sound format, popularized on the UNIX Machines. Other formats such as WAV (a Microsoft DOS and Windows sound format) must be converted to the AU format. (There are public domain utilities for format conversion such as Goldwave.)

MultiThreading and concurrency

A thread is a specific task or a set of instructions executed by the applet at any given time. There are times when more than one task must be executed at the

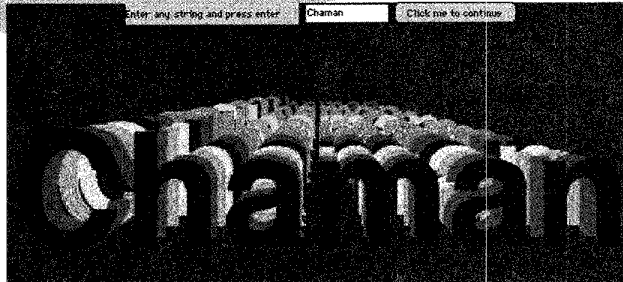


Fig. 5 The output generated by varying the sizes and colors of characters.

same time. MultiThreading is the process of executing two or more threads at the same time. Unlike other languages, Java provides a built-in library for multitasking rather than a separate layer of software to implement parallelism such as PVM (Parallel Virtual Machine) or MPI (Message Passing Interface). Even though the client may be a uniprocessor, Java simulates concurrent processing via threads giving the illusion of multitasking. Creating and using threads is very easy in Java.

Java allows us to create threads in two ways. The common way to create a thread class is by extending the Thread class. The Thread class implements run() method. The thread subclass inherits and overrides the run() method for concurrent processing. The other equally common way is to implement the Runnable interface in the applet class and override the run() method. In either case, a thread overrides the run() method for specific processing.

In concurrent programming, there are deadlock problems. These problems are

very easily handled by using the keyword "synchronized."

There are several other help methods: sleep, yield, wait, notify, suspend, resume and priority which make programming concurrency much easier.

Networking

Since Java is designed to be used on the Internet, it comes loaded with networking libraries such as Transmission Control Protocol/Internet Protocol (TCP/IP) networking and protocols used on the Internet, such as HyperText Trans-

fer Protocol (HTTP) and File Transfer Protocol (FTP). It is very easy to provide network applications through applets. Thus, the concept of an applet also fits well with a distributed software model. Since Java is endowed with built-in networking libraries, it adds to the functionality of traditional HTML and CGI scripts. The good thing is that you do not have to depend on the server administrator to provide services to run CGI type applications. This is an extremely useful feature.

It is very easy to connect to other web sites. To communicate on the Internet, one only needs to know the location's URL. AppletContext object is the web browser which runs the applet. The Applet class has a method, getAppletContext(), which returns the AppletContext object. In Java, one can connect to other sites with a single line of code.

Other issues

There are many other things in Java which are useful for the programmer and enterprise computing. These

Glossary

- bytecode— Java compiler compiles the source code into platform-independent code call bytecode.
- init— init() method is used to initialize data value. Also Applet executes init() method before any other method.
- instantiation— instantiation creates an object instance of a class.
- I/O— Input/output
- javac— javac is a command that invokes the Java compiler which translates the source code into bytecode.
- paint— paint() method is used to do graphics in the display window.
- RISC— Reduced Instruction Set Computer
- runnable— Runnable is a Java interface which has the run method. Any class that implements Runnable, must implement the run() method.
- text code— Java source code can be written by using any word processing editor, but it can be saved as a text file or text code.
- thread— Java class to create concurrent processes.

include pitfalls in the Java language at both the compile level and the run level. For instance, there are several inconsistencies in the language. These may be confusing to the programmer.

a) To create an instance of date 1901/01/01 new Date(1901,00,01) is used because the months begin at zero.

b) To get the size of an object:

1. activeCount() is used for a thread.
2. countItems() is used for a choice list.
3. length() is used for a file.
4. size() is used for a vector.

Also, important issues involve the second wave of Java development kits, Client/Server applications, Database applications including Remote Method Invocation (RMI) and Common Object Request Broker Architecture (CORBA) which have not been mentioned due to space limitations.

This article provides only snippets of some programs. All the complete programs illustrate the language concepts through programmed fully functional exercises. These are available from the author at <<http://www.umn.edu/~chaman>>.

Acknowledgments

The author developed this paper

while teaching the Winter 1997 Csc301 class of very enthusiastic students who were really motivated to learn the subject. It is enjoyable to have such students. The author thanks the reviewers for substantial improvement in the presentation of this article.

Read more about it

Within a year of emergence, there were 163 published Java books with 161 Java books in the works for a total of 324 English language Java books. Another 228 foreign language books make a total of 552 books worldwide <<http://lightyear.ncsa.uiuc.edu/~srp/java/javabooks.html>>.

* Andreeson Marc, Netscape Communications Corporation, The Red Herring, October 1996.

* Chen Mark C., Steven W. Griffith, and F. Iasi, *1001 Java Programming Tips*, Jams Press, 1997.

* Deitel H. M. and P. J. Deitel, *Jav a How to Program*, Prentice Hall, 1997.

* Eskow, Dennis, *NetReadyAdvisor*, Winter 1997.

* Gosling James, Frank Yellin, *The Java Application Programming Interface, Core Packages*, Vol. 1, Addison Wesley, 1996.

* Gosling James, Frank Yellin, *The Java Application Programming Interface, Window Toolkit and Applets*, Vol. 2, Addison Wesley, 1996.

* Harold, Rusty Elliot, *Java Developer's Resource*, Prentice Hall, 1997.

* Laffra, Chris, *Advanced Java*, Prentice Hall, 1997.

About the author

Chaman L. Sabharwal is Professor of Computer Science at the University of Missouri-Rolla, Graduate Engineering Center at St. Louis, Missouri. He has been Professor of Mathematics at St. Louis University and Lead Engineer and Senior Specialist at McDonnell Douglas Corporation. He developed an object-oriented software Engineering Methodology for the IRM project for McDonnell Douglas Corporation. He designed and implemented an automated symbolic kinematics solver for robot arm, algorithms for Parametric Triangular Blending Patches, reparameterization of Parametric Curve String Structures, computationally intensive algorithms for curve and surface modeling. He has published in various conference proceedings and journals including *SIAM Journal on Applied Mathematics*, *Computer Aided Geometric Design*, and *Pattern Recognition Journal*.

BENEFITS

Discounts on conferences, workshops and tutorials.

Discounts on leading technical publications. Just \$10/year for the MTT Transactions.

Free MTT-S newsletter.

Graduate fellowships.

Student paper contest.

Attend local, national and international MTT sponsored meetings.

Meet colleagues and learn!



We are everything from basic researchers through applied engineers working on RF, Microwave and Millimeter-Wave Technologies. We are over 9500 members strong and have over 90 local chapters. We publish leading technical journals and sponsor preeminent technical meetings. Join us!

OUR MEMBERS WORK ON

RFICs

Wireless components and systems

Digital microwave systems

Solid-state devices

Micromachined devices

Millimeter-wave components and systems

Fiber and integrated optics

Field theory

Superconductivity

Measurement techniques

JOIN

Yes! I am a Student member for 1999 and I want to join MTT-S (free for the first year).

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

Member #: _____

Send to:

IEEE Operations Center
PO Box 1331
445 Hoes Lane
Piscataway, NJ
08855-1331

Acct# 30-0170-01900-40852-0** POT1799 109589

TODAY