Week 1

1st Programming language typically Pascal / C

* structured programming language
* divide and conquer

This course teaches both OOPS and structured programming

* OOPS will progressively gain acceptance
* still lots of C out there
* internals of methods demand structured programming ideas

Why Java?

* Enhanced multimedia capabilities incorporating images, animation audio and video.
* Enables applications to run on the Internet
* Multithreading
* Reusability - and existing libraries of code
* Portability - applications written on one platform will run on another
* Fully object oriented - cannot write C like procedural programs in Java - you must create and manipulate objects

Basic Terminology:

* Computer Organization: Input Unit - Output Unit - Memory Unit - ALU - CPU –

and Secondary Storage

* Operating Systems: job/task - batch processing - multiprogramming - time sharing w/terminals
* Languages: machine language - assembly language - high-level languages
* OOP: C - Dennis Ritchie at Bell for DECPDP11 in 1972)

 C++ - an extension of C using Object Oriented Programming (OOP).

 - Uses *objects* (reusable software that model items in the real world)

 Java - James Gosling at Sun Microsystems, 1991

* originally called OAK (after Oak tree outside office) but name was already used
* after visiting a local coffee place, name changed to Java
* WEB exploded in 1993 - therefore new life to Java
* formally announced in May 1995

# What’s it all about?

* Hardly anyone is a true Internet expert due to new Internet technologies such as Java
* Most consultants specialize in only one or two areas (due to newness)

This course does not require that you be an Internet guru to understand and use Java. Nevertheless, be sure that you’ve used the Internet enough to understand home pages and maneuver around the *WWW (World Wide Web)*. If you’ve programmed in *HTML (Hypertext Markup Language)*, you’ll be further ahead of the game. If you’ve never written HTML code, don’t fret! I will explain enough for you to build and run Java programs.

A. Simple Navigation

1. *Home Pages* are graphical Internet screens with hot spots that you click to move around the intricate Internet.
2. Web browsing software takes you on a journey through a graphical maze (a spider web of interrelated connections, hence the term *Web*.
3. An Internet *site* might contain several Web pages.
4. The first screen you visit for any site generally is that site’s home page.
5. The home page often contains links to other pages on the site and to other home pages on the Internet.
6. Each page you see on the Web is located somewhere on someone’s computer.

The Web works a lot like the Windows hypertext help system. By clicking hypertext links (sometimes called *hot* spots) or by directing users to another page, you can maneuver throughout the World Wide Web just as you do with online help. Instead of viewing another help screen from your application’s directory, however, you could be viewing an Italian museum’s images or its masterpieces.

1. Each Web page contains a unique address called the *URL (Uniform Resource Locator)*.
2. Instead of requiring that you type each page’s URL, the current page’s hypertext links direct your Web browsing software to the next URL that you wish to display.
3. Surfing the Web is as easy as flipping through pages in a book.
4. A Web page might contain text, graphics, and even icons you can click to view multimedia events.
5. Web pages appear uniform no matter what kind of computer (or, more accurately, which *platform*) the end-user uses to view the page.
6. *HTML (hyper text markup language)* is a machine-independent language that Web page developers use to design Web pages.
7. The terms within angled brackets (< and >) are called *tag references* (or *tag commands*).
8. Many commands contain a beginning and ending tag with a forward slash preceding the ending tag.
9. Tags primarily determine the placement of figures, the location of links to other Web sites, and the formatting of text (i.e., font style and size instructions).
10. Non-bracketed text items are literal constants that are to appear on the resulting Web page.
11. HTML is the vehicle with which you’ll eventually link Java applications to Web pages.
12. One simple HTML tag embeds a complete Java application into any Web page.

When you navigate to a Web page, the server sends to your browser only the HTML text, not the graphical image of the Web page. Your browser then reads the text and responds to the commands by formatting text appropriately and placing links and graphic images where the HTML dictates they should appear. Your browser first receives the full HTML page and then receives whatever graphic images are requires to complete the page. Most browsers offer a *Stop* button that you can click to keep from receiving the graphic images for the times when you don’t want to wait on the images but only want to see the text. In place of those images, your Web browser will display icons that represent the unsent images.

### **Internet: From Text to Multimedia**

The World Wide Web is not the Internet - It is only a tool that dynamically links Web sites to one another and makes for uniform Web retrieval and navigation. The Internet is defined as the collection of interconnected computers all over the globe.

When the Internet first appeared, users had a difficult time locating information on other computers connected to the Internet. One had to be a master of the UNIX operating system to utilize the Internet effectively. Successful Internet usage required a mastery both of your own computer’s hardware and operating system as well as a fairly in-depth knowledge of Internet connections. Internet retrievals required sitting at a text-based terminal and issuing commands to get to where you wanted to go.

In the early 1980’s, computer users realized that additional tools were needed to make the Internet usable for more people. Not only were early text-based navigation tools cryptic to use, but they were also machine-dependent: an individual user could not often go to another computer and access information using a common interface. Internet developers began linking their information together into a simple hypertext system, which let users jump from topic to topic.

Given the transition of computers in the 1980s from a text interface to graphical user interface (GUI), the development of Mosaic in late 1993 finally provided the necessary commonality and graphic access to give Internet users a mouse-driven, graphical Web-browsing vehicle. In addition, the Web solidified the Internet as a tool usable by the masses - both computer-literate users and novices.

B. What is Java?

1. Java developed by Sun Microsystems in the mid-1990’s.
2. Java is a programming language with language features similar to C++.
3. Instead of using Web browsers to view data, the browsers seamlessly download programs written in Java which execute on the user’s computer.
4. There are two kinds of programs you can write with the Java language:
* Java applets - small programs that travel with HTML code and execute on the Web user’s computer.
* Java applications - complete stand-alone programs that do not require a Web browser or HTML to execute.
1. Most Java programs to date reside in applets.

C. Java Provides Executable Content



D. Multi-Platform Executable Content

1. When you write a Java-based Web page, you want the code to work on the end-user’s remote computer no matter what kind of computer the remote user uses.
2. Bytecode - Most programming compilers (such as C++ and Visual Basic) compile code into a machine-dependent executable program. All Java compilers compile into a special machine-independent module called bytecode.
3. No computer can really read bytecode, but each computer’s Java-enabled browser can. Therefore, browsers such as Netscape Navigator and Internet Explorer can run the very same bytecode.
4. Each Web browser actually interprets the machine-independent bytecode and then translates that bytecode into machine-specific instructions that particular computer can understand.
5. Java is considered *architecture neutral* in that a Java program is not written for a specific machine but rather for the universal virtual machine.

### **A Coding Background**

A program is a set of instructions, written in a programming language such as Java, that tell the computer what to do. Both Java applets and Java applications are programs.

You’ll use Borland’s JBuilder software package (included with your textbook and installed in our lab) to write the code, or programming instructions, that will form you Java programs. Once you write the code, you’ll have to compile the code. No computer can really understand a programming language such as Java. Programming languages exist for humans. When you compile the program, a compiler translates your human-readable code into machine-readable code.

Due to the virtual machine nature of Java, Java code is not completely compiled, but the Java compiler compiles the code into an in-between stage called bytecode. Your Java-enabled Web browsing software then translates this compiled bytecode into instructions that your computer can then execute.

##

When you want the fastest response time possible for a user’s interaction with a Web page, you will often want to use a Java applet. Although the user still has to wait for the Web page to download, and although the user still has to wait for the bytecode to get to the computer, an applet is relatively small, and it will execute as quickly (as quick the user’s CPU can allow). The user is no longer bound to interactive; back and forth download times once the computer gets the executable content. All other forms of Web page interaction are slaves to download response time and current Internet activity.

### Java Usage Summary:

1. You log on to the Internet, using your web-browsing software.
2. You see your Web browser’s default home page.
3. You enter a Java-enabled web site’s URL.
4. The serving computer sends you your Web browser the HTML document.
5. The document’s <APPLET> tag informs your Web browser of the Web page’s Java-based executable content.
6. Your Web browser downloads graphic images from the server, if any graphics appear on the Web page
7. Depending on how the Java applet is to be triggered (either automatically or by a user’s event), the server will also send the bytecode to your computer (the target) when the time is right.
8. Your computer’s Web browser interprets the bytecode and executes the java executable content.
9. When you leave the Web page, the executable content goes away. (Some Web browsers will keep the Java code in memory for a while in case you return to the page.)

### Creating Web pages with Java Applets:

1. Start Borland JBuilder.
2. Create your Java applet.
3. Create the HTML Web page that will contain the Java applet. Use appropriate tags to indicate the applet and its parameters.
4. Compile the Java applet using JBuilder. JBuilder keeps track of Java applets using the project concept. You’ll store each applet in the project workspace, and all associated files for that project will also reside in the project workspace, including the HTML Web page.
5. If the compiler finds errors, fix them and recompile the project’s files.
6. Test your applet. You can test Java project applications without logging on to the Internet. If the Java applet does not do what you intended, fix the problem and recompile the applet.
7. Store the applet and HTML on you Web server, where it will await an Internet user’s request.

E. Multithreaded Applications

1. A *multithreaded application* is a program that might spin off into more than one *thread* (i.e., you can be reading and scrolling through text while graphic images are loading elsewhere on the page - or, you can download a file or image by clicking an icon and then, while the downloading takes place you can start the download of yet another image or file.)

#### *CAUTION*

*Don’t be fooled. At its lowest level, a computer can only execute one instruction at a time. Nevertheless, computers run so quickly that they can appear to perform more than one task at the same time. Although there are a handful of computers that contain multiple processors, these multi-processor computers are not yet in the mainstream; most people still use a single processor machine.*

G. Borland JBuilder Specifics:

1. JBuilder is a programming platform used to develop Java-enabled applets and applications.
2. JBuilder includes not only a Java compiler, but also an editor, debugger, and online documentation provider.
3. The JBuilder compiler can compile over one million lines of Java code each minute.
4. The JBuilder editor contains a word processing set of powerful text features including the usual search, cut, and paste, as well as multi-pane viewing, moving and copying between windows, bookmarks, full-screen viewer, and advanced windowing features.
5. JBuilder contains a complete online help system that includes all available documentation.
6. The debugger lets you specify breakpoints and examine data contents at any point in your program.
7. JBuilder integrates *COM (Common Object Model)* so that you can integrate external applications such as a working Excel spreadsheet, into your own Java applets and applications.

#### Summary

* The Internet became more usable for mass consumption as soon as the World Wide Web came onto the scene with browsers that displayed graphic pages and allowed for simple navigation between Web sites.
* Java-enabled Web pages provide true interaction and animation.
* A Java program is actually an applet, or a miniature application, that executes on the end-user’s computer as soon as the user looks at the Web page and triggers the Java program.
* Java is a robust but fairly secure system that respects many security boundaries and provides safe content for both developers and users.
* The Java language provides for both protocol handlers and content handlers, so you need not worry about your user’s successful receipt of the data you send with you Java applet.
* JBuilder is one of the most powerful Java development systems available.