



# Lab: Introduction to Java Programming

*IBM @server iSeries 400 and AS/400e*

**Clifton Nock**

© Copyright IBM Corporation, 2000. All Rights Reserved.  
This publication may refer to products that are not currently  
available in your country. IBM makes no commitment to  
make available any products referred to herein.

**IBM @server. For the next generation of e-business.**

# *Lab: Introduction to Java Programming*

---

This publication may refer to products that are not available in your country.

IBM makes no commitment to make available any products referred to herein.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX, AS/400, DB2, IBM, OS/400

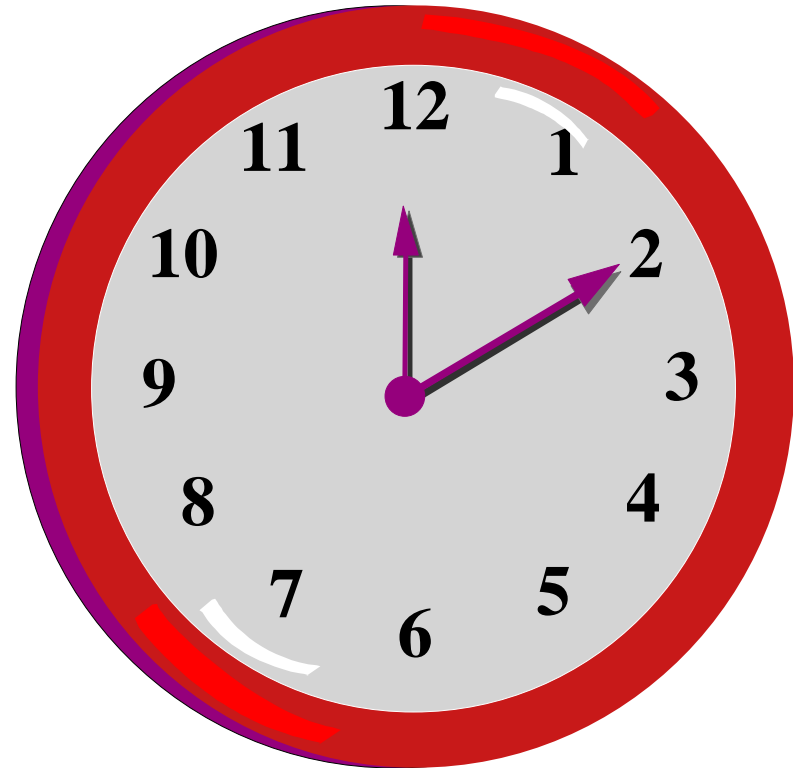
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product or service names may be the trademarks or service marks of others.

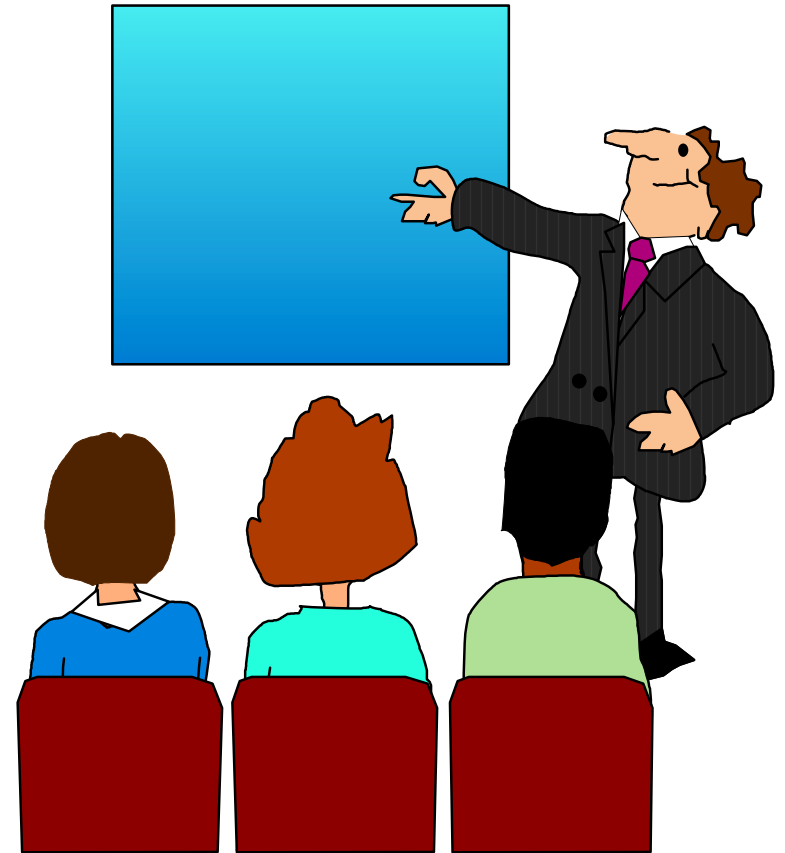
# *Agenda*

- Introduction
- Buzzwords
- Environments
- Program structure
- Syntax
- Development tools
- References
- Lab exercises



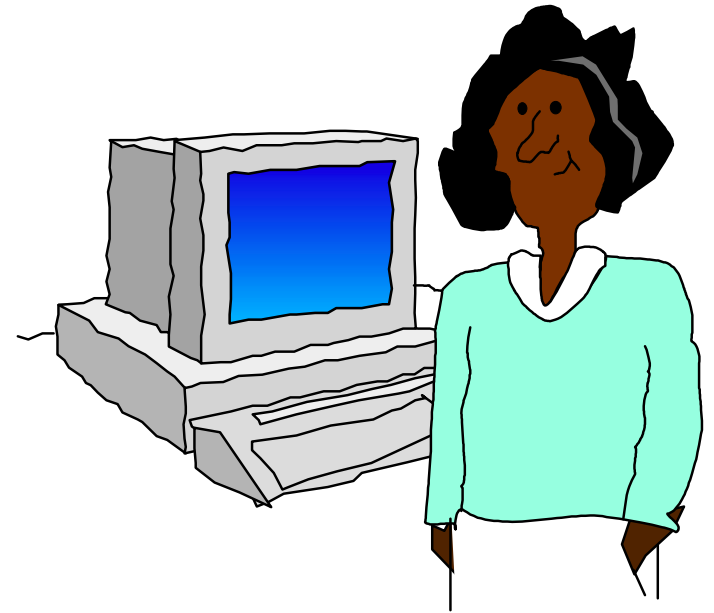
# Buzzwords

- *"Simple"*
  - Syntax is based on C++
  - Avoids complicated aspects of C++:
    - no pointer arithmetic, no memory management
  - Many reusable objects available★
- *Object oriented*
  - Classes, objects, methods, inheritance
- *Distributed*
  - TCP/IP, HTTP communication built in
  - Applet and servlet programming models
  - "Internet-ready"
    - (it is great for non-Internet apps, too!)



# Buzzwords

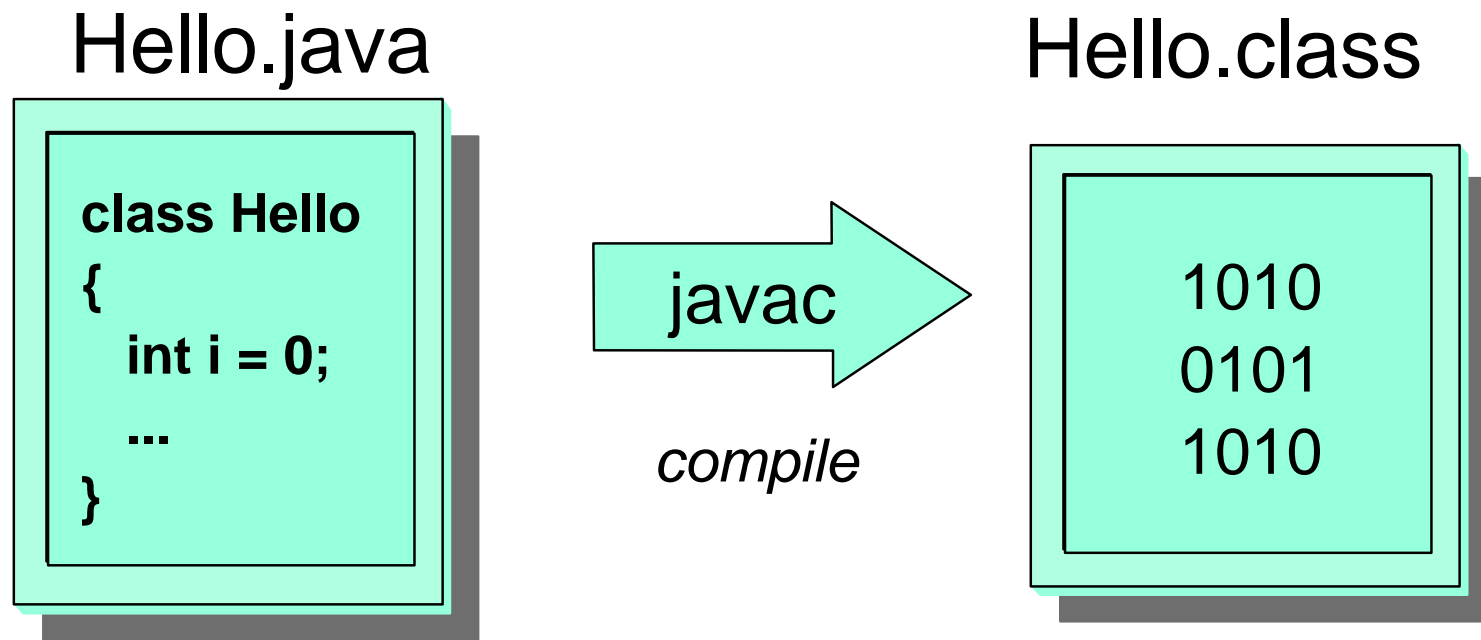
- *Robust*
  - Array bounds checking
  - Automatic garbage collection
- *Secure*
  - Applet sandbox
  - Class verifier - prevents "forged" Java code
- *Other built in features*
  - Threads
  - Graphical user interface components
  - Collections
  - Input/output
  - Exception handling
  - Internationalization (national language support)



# Buzzwords

- *Portable*

- Source code (.java file) edited and compiled on any platform
- Compiled bytecodes (.class file) run on any platform
- All aspects of pure Java are portable (GUI, threads, networking, I/O, etc.)



# Environments: Applications

- Program is loaded from local or network storage
- Runs on local processor
- Can do just about anything!



SQL Query  
Viewer

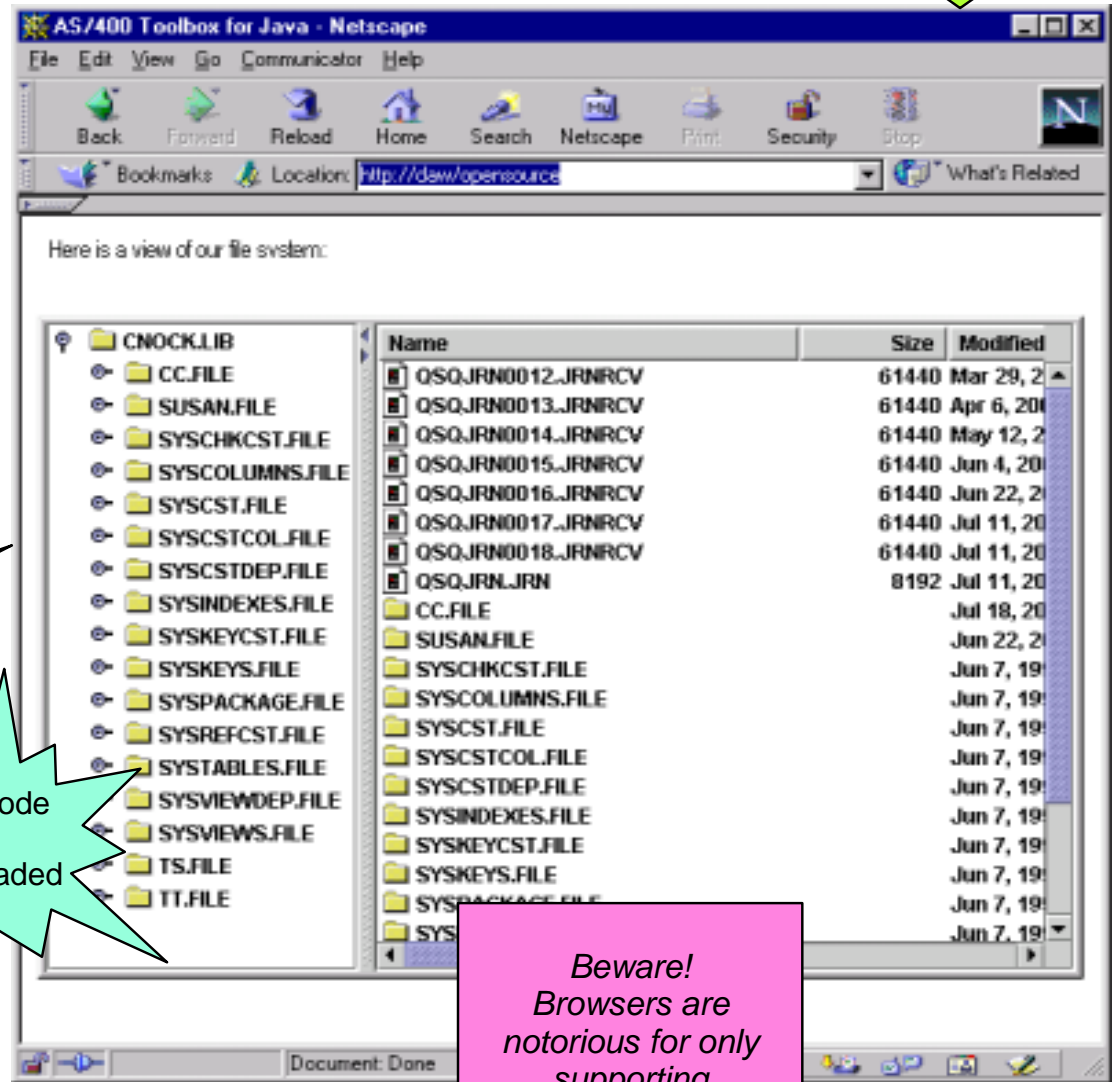
Testing the visual classes

Name	Size	Modified
QSQJRN0012.JRNRCV	61440	Mar 29, 2000 8:01:43 AM
QSQJRN0013.JRNRCV	61440	Apr 6, 2000 6:19:19 PM
QSQJRN0014.JRNRCV	61440	May 12, 2000 9:01:06 AM
QSQJRN0015.JRNRCV	61440	Jun 4, 2000 10:58:21 PM
QSQJRN0016.JRNRCV	61440	Jun 22, 2000 12:13:17 PM
QSQJRN0017.JRNRCV	61440	Jul 11, 2000 7:59:22 PM
QSQJRN0018.JRNRCV	61440	Jul 11, 2000 7:59:22 PM
QSQJRN.JRN	8192	Jul 11, 2000 7:59:22 PM
CC.FILE		Jul 18, 2000 4:02:18 PM
SUSAN.FILE		Jun 22, 2000 9:15:52 AM
SYSCHKCST.FILE		Jun 7, 1999 7:55:00 AM
SYSCOLUMNS.FILE		Jun 7, 1999 7:54:59 AM
SYSCST.FILE		Jun 7, 1999 7:54:59 AM
SYSCSTCOL.FILE		Jun 7, 1999 7:55:00 AM
SYSCSTDEP.FILE		Jun 7, 1999 7:55:00 AM
SYSINDEXES.FILE		Jun 7, 1999 7:54:59 AM
SYSKEYCST.FILE		Jun 7, 1999 7:55:00 AM
SYSKEYS.FILE		Jun 7, 1999 7:54:59 AM
SYSPACKAGE.FILE		Jun 7, 1999 7:54:59 AM
SYSREFCST.FILE		Jun 7, 1999 7:55:00 AM
SYSTABLES.FILE		Jun 7, 1999 7:54:58 AM
SYSVIEWDEP.FILE		Jun 7, 1999 7:54:59 AM
SYSVIEWS.FILE		Jun 7, 1999 7:54:59 AM
TS.FILE		May 31, 2000 4:13:34 PM
TT.FILE		

# Environments: Applets

- Program is downloaded as part of a web page, along with text and images
- Runs on local processor, but always within the context of the web browser
- Input and output via GUI
- Extra security measures:
  - limited file access
  - limited network access

Java code runs as part of the browser



Java code is downloaded

Beware!  
Browsers are notorious for only supporting backlevel versions of Java

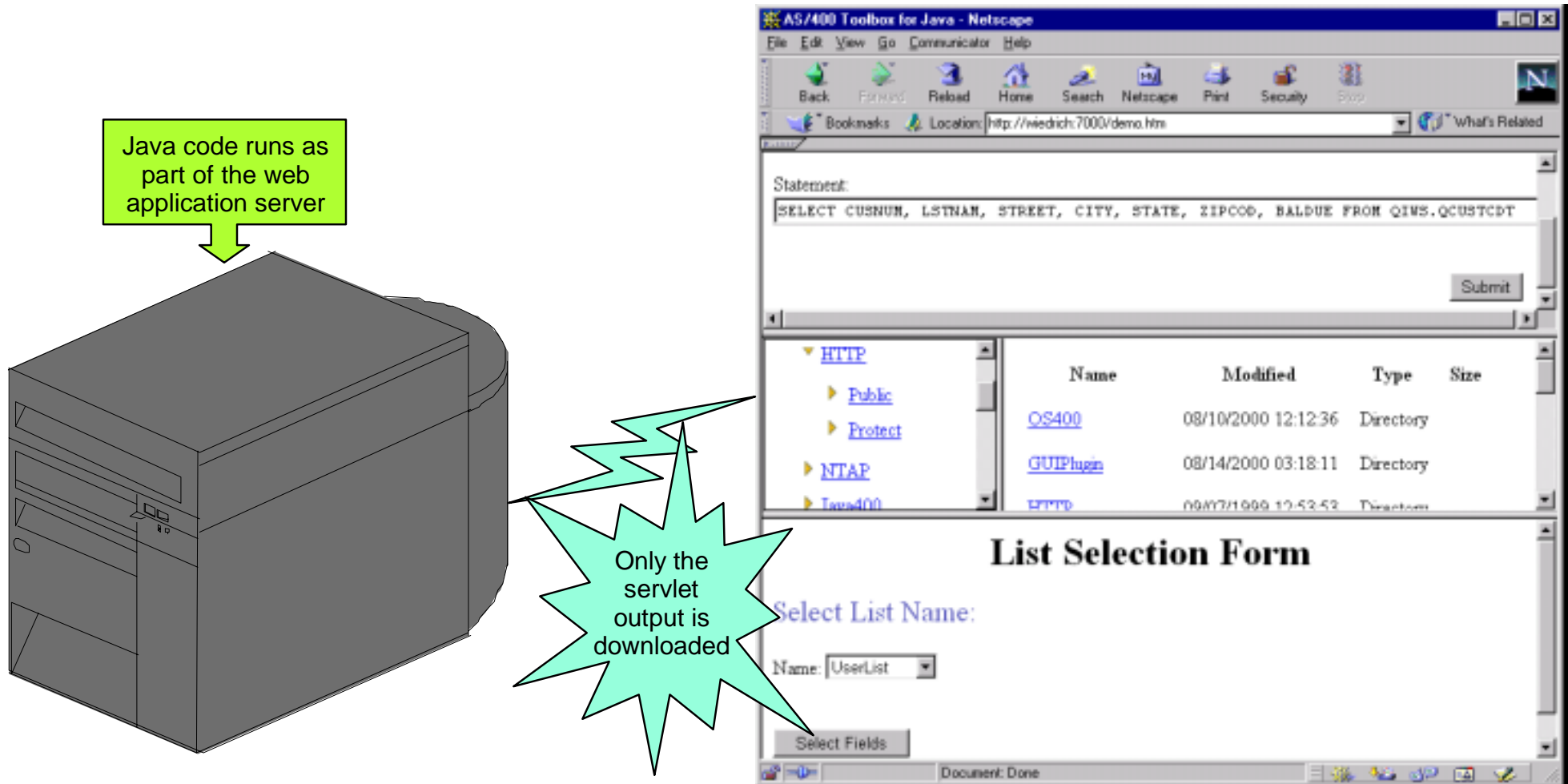
Java != JavaScript

AS/400e



# Environments: Servlets

- Runs on web server in response to an HTTP request (e.g. `http://www.mysite.com/info`)
- Output is HTML (not GUI) - the HTML is sent back to the client and presented in the browser



# Program structure

## ReportGenerator.java

**Imports** instruct the compiler where to find other classes that are used.

A **class** defines the variables and methods for a type of **object**.

A **constructor** is a special method used for initializing an object. Its name matches the class name.

All data and methods are strongly-typed. Java compilers are very strict!

```
import java.util.Date;
```

```
public class ReportGenerator  
{
```

```
    private Date reportDate;  
    private String title;
```

```
    public ReportGenerator(String theTitle)  
    {  
        title = theTitle;  
        reportDate = new Date(); // The current date and time.  
    }
```

```
    public void print()  
    {  
        System.out.println("Report: " + title);  
        System.out.println("Generated on: " + reportDate);  
        // ... more Java code here ...  
    }
```

```
    public static void main(String[] args)  
    {  
        ReportGenerator rg = new ReportGenerator("Marketing expenses");  
        rg.print();  
    }
```

```
}
```

There is usually one **class** per source file. Its name matches the source file name.

A **field** defines the part of the content (data) of the object. A **private field** is usable only within this class. This is called **encapsulation**.

**Comments** are denoted by `//` or `/* */`.

A **method** defines a behavior of the object. It is analogous to a function, procedure, subroutine, etc.

The **main()** method is special - it is the **entry point** for a Java application

# Syntax

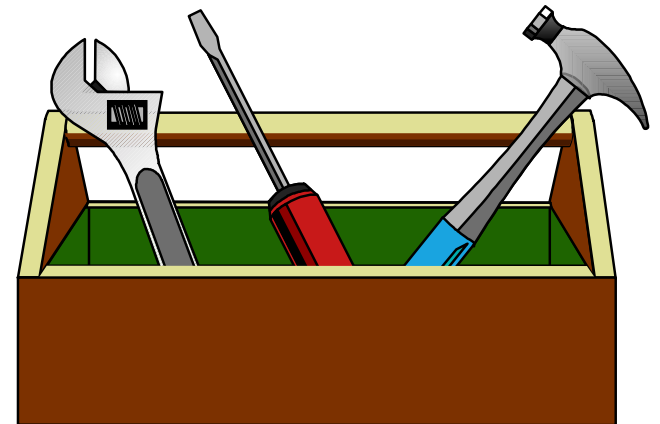
- Everything in Java is case-sensitive
- Java syntax is based on C++
- Whitespace is not significant
  - Use indentation and whitespace to make code readable
- Everything in Java is case-sensitive
- Variable names can be any length
  - Using descriptive names can lead to self-documenting code
  - Can be any Unicode characters (except for a few)
- Java handles all Strings as Unicode
- *Everything in Java is case-sensitive!*

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

```
public class HelloWorld { public static void main(String[] args) { System.out.println("Hello World"); } }
```

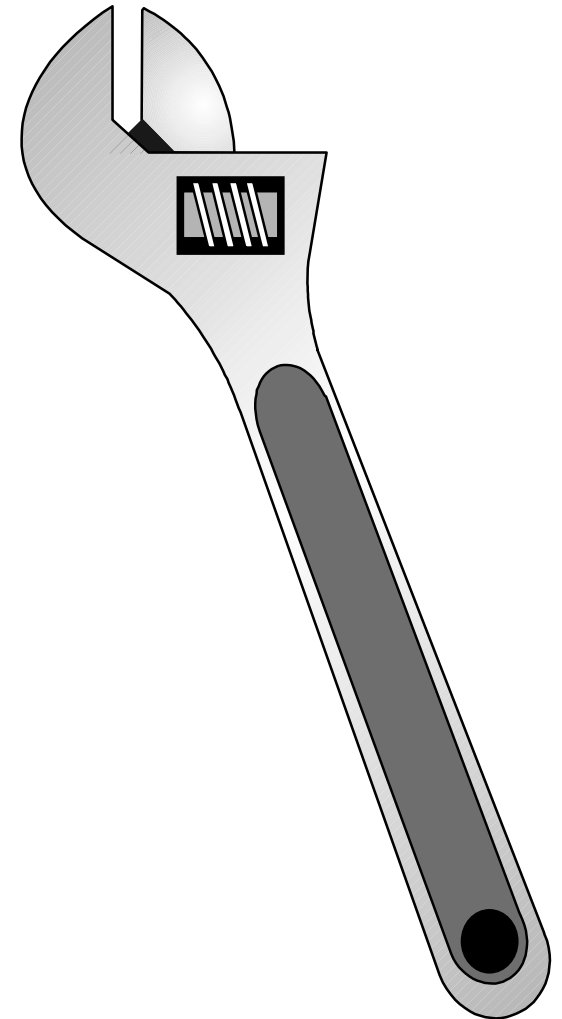
# Development tools

- **Java Developers Kit (JDK or SDK)** - includes compiler, JVM, debugger, javadoc, appletviewer, etc.
  - Available from Sun, IBM, others...
  - **The minimum you need for Java development**
- **Compiler** - compiles source code (.java file) into Java bytecodes (.class file)
  - Sun: javac
  - IBM: jikes - much faster!
- **Jar** - Java archive file
  - Used for packaging multiple .class files
- **Java Virtual Machine (JVM)** - runs Java bytecodes
  - Interprets bytecodes and runs as machine instructions
  - Just-In-Time (JIT) compiler - caches machine instructions
  - AS/400 stores optimized machine instructions with .class file
  - **The minimum you need to run Java programs**



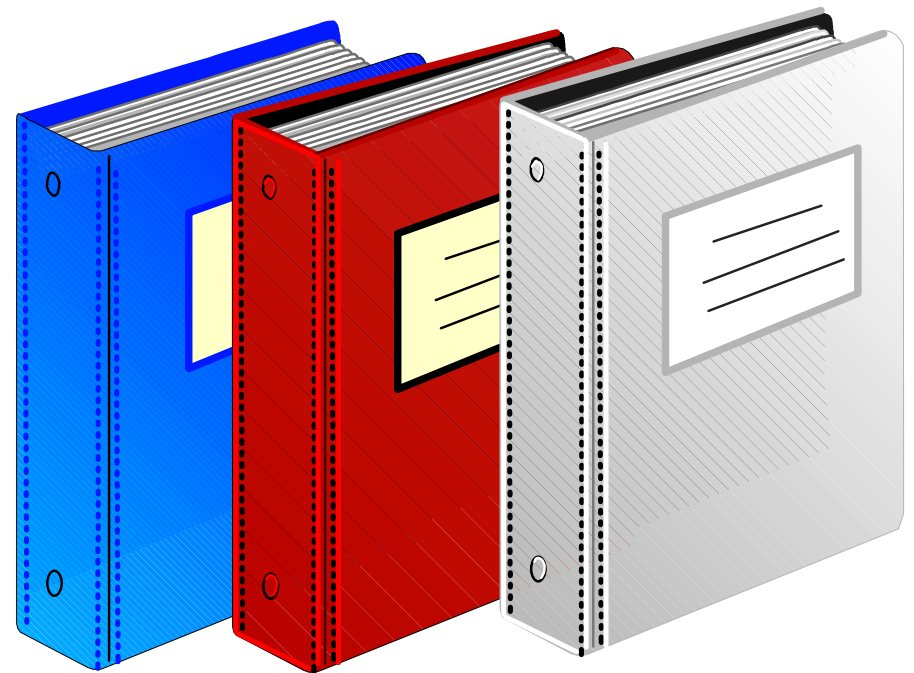
# *Development tools*

- Integrated development environment (IDE)
  - Many available from many vendors (e.g. IBM VisualAge for Java)
  - Includes graphical debugger, visual development
- Toolbox for Java/JTOpen
  - Building blocks specifically for AS/400 Java programs
  - Runs on client or server
  - Open source
  - Example code is available



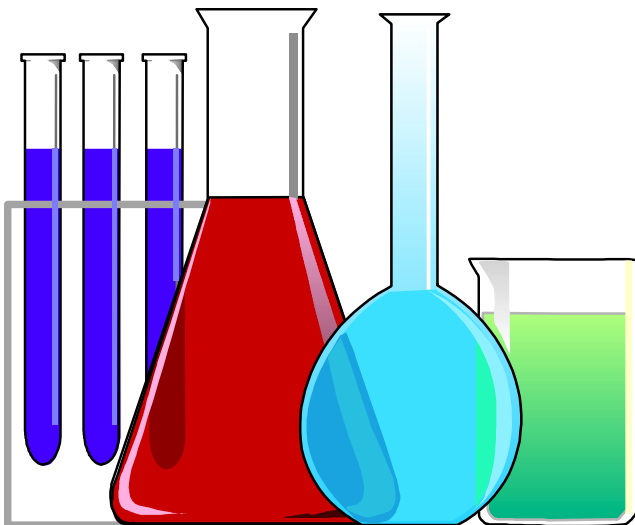
# *References*

- [java.sun.com](http://java.sun.com)
- [ibm.com/java](http://ibm.com/java)
- [ibm.com/as400/java](http://ibm.com/as400/java)
- [ibm.com/as400/toolbox](http://ibm.com/as400/toolbox)



# *Lab exercises*

- Create and run a simple Java application on Windows
- Run a simple Java application on AS/400
- Optimize an AS/400 Java application
- Create a simple web page
- Create and run a simple Java applet
- Use Qshell on an AS/400
- Create and run a simple Java servlet



*Please ask questions!*