

A Taste of Java

Barry Cornelius

IT Service, University of Durham, Durham, UK

last updated: 25th June 1997

mailto:Barry.Cornelius@durham.ac.uk

http://www.dur.ac.uk/~dcl0bjc/Java

©1997 Barry Cornelius

1 Introduction

1.1 What is Java?

- Java is an object-oriented programming language developed by Sun Microsystems.
- It has: garbage collection, multithreading and exception handling.
- It does not have: architecture-dependent constructs, structs, unions, pointer arithmetic, operator-overloading and multiple inheritance.
- It is accompanied by many *packages* (collections of classes) including one for building GUIs.

1.2 How is it executed?

- Unlike most programming languages, Java source code is not compiled into native code. Instead, a Java compiler translates Java source code into an architecturally-neutral intermediate form known as *bytecode*.
- Instructions in this bytecode are interpreted by a Java interpreter.
- Sun's *Java Development Kit* (JDK) includes a compiler and interpreter. They provide (free of charge) versions for Solaris 2.x (both sparc and x86), for Windows 95, for Windows NT and for MacOS System 7 (and above). Although both JDK 1.0.2 and JDK 1.1.x are currently being used, where possible new systems should be written using JDK 1.1.x. Both JDK 1.0.2 and JDK 1.1.x can be downloaded from: <http://java.sun.com/nav/download/>.
- Programs written for the JDK 1.0.x definition can be compiled by a JDK 1.1.x compiler. Bytecodes produced by a JDK 1.0.x compiler can be interpreted by a JDK 1.1.x interpreter.

1.3 What are Java applications?

- A Java application is a conventional program. It must have a method called `main`.
- Suppose the file `HWTion.java` contains the Java application:

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

- It can be compiled by using the Unix/MS-DOS command:
`javac HWTion.java`
This produces a file of bytecodes in the file `HWTion.class`.
- The file `HWTion.class` can be interpreted (i.e., executed) by using the command:
`java HWTion`

1.4 What are Java applets?

- A Java applet is Java source code whose bytecodes will be executed as part of viewing a WWW page.
- The applet's author compiles the Java source code into bytecodes.
- These bytecodes will be downloaded from their author's site by a WWW browser when the WWW page is visited. So, the browser needs to have a Java interpreter to interpret the bytecodes.
- This is true for WWW browsers that are *Java-aware*, e.g., Sun's *HotJava*, Microsoft's *Internet Explorer*, or Netscape's *Navigator*. You will have to use the latest version of a browser if you have JDK 1.1.x bytecode files that use the new facilities of JDK 1.1.x. Examples are Version 4.x of Netscape's *Navigator* (which is now available) and Version 4 of Microsoft's *Internet Explorer* (which will be released in beta on 31st July).
- In the past, there has not been much support for Java-aware WWW browsers on Windows 3.1. There are few versions of Netscape's *Navigator* 3.x supporting Java on Windows 3.1. At the moment, the choice is between Netscape's *Navigator* 4.x (which supports JDK 1.1.x) and a version of Microsoft's *Internet Explorer* (3.01) (that supports JDK 1.0.2).
See: <http://home.netscape.com> and <http://www.microsoft.com/ie/product/win31/java16.htm>.
- Sun's JDK also includes an *appletviewer* that can be used if you do not have a Java-aware WWW browser.

- Suppose `HWLet.java` contains the Java applet:

```
import java.awt.Graphics;
import java.applet.Applet;
public class HWLet extends Applet {
    public void paint(Graphics gGraphics) {
        gGraphics.drawString("Hello World!", 50, 25);
    }
}
```

- It can be compiled by using the Unix/MS-DOS command:
`javac HWLet.java`
This command produces the file `HWLet.class`.

- When a browser reads the WWW page given below, it finds that it has to retrieve the file `HWLet.class`. When the bytecodes in this file arrive, it can interpret them.

```
<HTML>
<HEAD>
<TITLE> HWLet example </TITLE>
</HEAD>
<BODY>
Before the output from the applet.
<APPLET CODE="HWLet.class" WIDTH=150 HEIGHT=25>
</APPLET>
After the output from the applet.
</BODY>
</HTML>
```

- You can access a WWW page containing the above HTML instructions by using the URL:
<http://www.dur.ac.uk/~dcl0bjc/Java/a.taste.of.java/code/HWLet.html>.

1.5 Some real examples

- Sun's WWW browser *HotJava* is written in Java: <http://java.sun.com/products/hotjava/>.
- Sun has produced an IDE (Integrated Development Environment) for Java written in Java called *Java Workshop*: <http://www.sun.com/workshop/java/>.
- Borland's *AppExpert* and *Debugger* for Java are written in Java: <http://www.borland.com/jbuilder/introguide/java1.1.html>.
- Two groups have produced WWW servers written in Java. Sun has produced *Java Web Server* (formerly known as *Jeeves*) at <http://jserv.javasoft.com> and the World Wide Web Consortium has produced *Jigsaw* at <http://www.w3.org/pub/WWW/Jigsaw/>.
- A beta release of *Corel Office for Java*, a suite of office productivity applications (including *Quattro Pro* and *WordPerfect*) written entirely in Java, is at: <http://officeforjava.corel.com>.
- Applix's *Anyware Office* is a complete suite of applications (written in Java) including a word processor, a spreadsheet, business graphics and an e-mail client. It can be run on "Webtops, NCs, Windows 95, NT and Unix desktops and servers" and costs \$295 per concurrent user. See: <http://www.applix.com>.
- For other uses of Java, see: <http://www.sun.com/javastation/customersuccesses/>, <http://java.sun.com/nav/used/index.html>, <http://www.sun.co.uk/java/jcentre.html> and: <http://www.sun.co.uk/java/jenterprise.html>.

1.6 A digression: what is JavaScript?

- Here is a WWW page containing some source code that is written in *JavaScript*:

```
<HTML> <HEAD> <TITLE> Square demo </TITLE> </HEAD>
<BODY>
<P> Start. </P>
<SCRIPT LANGUAGE="JavaScript">
<!-- hide this script from some browsers
function mysquare(myarg) {
    document.write("<P> Hello once more </P>");
    document.write("<P> <code>myarg</code> is: ", myarg, "</P>");
    return myarg*myarg;
} ;
document.write("<P> Value returned is: ", mysquare(7), "</P>");
// end of hide -->
</SCRIPT>
<P> Finish. </P>
</BODY> </HTML>
```

- JavaScript is another programming language.
- You put JavaScript code in HTML documents with a `<SCRIPT>` tag.
- The JavaScript code is not compiled: instead, it is interpreted by a JavaScript-aware WWW browser. Netscape's Navigator (including the Windows 3.x version) understands JavaScript.
- You can access a WWW page containing the above HTML instructions by using the URL: <http://www.dur.ac.uk/~dcl0bjc/Java/a.taste.of.java/code/square.html>.
- JavaScript has most of Java's expression syntax and basic control flow constructs, but it does not have Java's strong type checking and static typing. You cannot write your own classes.
- So, JavaScript is not as powerful as Java. IMHO, it is confusing for *Java* to be included as part of the name *JavaScript*.

2 The Java language

2.1 What's been taken out from C/C++?

- `#define`, `typedef`, `enum`, `union`, `struct`, `sizeof`, `goto`, pointer arithmetic
- global variables - every variable has to belong to a class (or is a local variable)
- standalone functions - every function has to be a method of a class
- optional arguments, operator overloading, templates
- friend functions - however, by default everything in a package is a friend of everything else in the package
- multiple inheritance - however, some uses can be replaced by interfaces

2.2 What's left?

- Java's primitive data types have specified ranges:
 - `boolean` (1 bit), `byte` (1 byte), `short` (2 bytes), `int` (4 bytes),
 - `long` (8 bytes), `char` (2 bytes, Unicode),
 - `float` (4 bytes, IEEE 754), `double` (8 bytes, IEEE 754)
- Java also has arrays, which can be multi-dimensional
- Java contains C's `if`, `else`, `for`, `while`, `do while` and `switch` statements. However, all conditions must return a `boolean`. So, some of the more obscure conditions of C are prohibited.

2.3 What's new? (a) Garbage collection

- Pascal/C/C++ programs inadvertently `dispose/free/delete` objects which are still in use:

```
var p, q: ^integer;    int *p, *q;          int *p, *q;
new(p);              p = malloc(sizeof(int));  p = new int;
p^ := 27;            *p = 27;              *p = 27;
q := p;             q = p;                q = p;
dispose(p);         free(p);              delete p;
writeln(q^);       printf("%d\n", *q);      cout << *q << endl;
```

- And programs often cause memory leaks by *not* using `dispose/free/delete` on unwanted objects.
- In Java, you don't delete objects: instead, Java has *garbage collection*. The garbage collector detects objects no longer in use, and reuses their space.
- You also don't have to provide destructors for classes.

2.4 What's new? (b) Multithreading

- In Java, you can easily start more than one *thread* of execution.
- If a variable is accessed by two (or more) methods which are called from different threads, you can ensure that the variable is accessed properly by using `synchronized`:

```
public class Store {
    private int i; ...
    public synchronized int get() { ... return i; }
    public synchronized void put(int v) { i = v; ... }
}
```

2.5 An example of a class

- In Java, it is possible to use a *class declaration* to define your own type. Suppose that it is necessary to manipulate some dates in a program. Here is a rather primitive class declaration:

```
public class Date {
    public int day, month, year;
    public void display() {
        System.out.println(day + "-" + month + "-" + year);
    }
}
public class UseDate {
    public static void main(String[ ] args) {
        Date noel = new Date();
        noel.day = 25;
        noel.month = 12;
        noel.year = 1996;
        noel.display();
    }
}
```

2.6 A better version of the class Date

- A private modifier can be used to hide the representation:

```
public class BetterDate {
    private int day, month, year;
    BetterDate (int d, int m, int y) {
        day = d; month = m; year = y;
    }
    public void display() {
        System.out.println(day + "-" + month + "-" + year);
    }
}
public class UseBetterDate {
    public static void main(String[ ] args) {
        BetterDate noel = new BetterDate(25, 12, 1996);
        noel.display();
    }
}
```

- If necessary, other methods can be provided to access the hidden fields or to update them.

2.7 An example of inheritance

- *Inheritance* can be used when a class is a more specialized form of another class:

```
import java.util.Vector;
public class Stack extends Vector {
    public Stack() { super(); }
    public void push(Object item) { addElement(item); }
    public Object pop() {
        int s = size();
        Object item = elementAt(s - 1);
        removeElementAt(s - 1);
        return item;
    }
}
```

- The URL: <http://www.dur.ac.uk/~dcl0bjc/Java/a.taste.of.java/code/UseStack.java> contains a Java application that uses the Stack class.

2.8 Exception handling: (a) throwing

- If a method detects that some untoward event has occurred, it can use a *throw statement* to signify that an *exception* has occurred:

```
import java.util.Vector;
public class BetterStack extends Vector {
    public BetterStack() { super(); }
    public void push(Object item) { addElement(item); }
    public Object pop() throws BetterStackException {
        int s = size();
        if ( s == 0 ) throw new BetterStackException();
        Object item = elementAt(s - 1);
        removeElementAt(s - 1);
        return item;
    }
}
public class BetterStackException extends Exception {
    public BetterStackException() { super(); }
}
```

- If a method throws an exception, this must be documented in the heading of the method by a *throws cause*.

2.9 Exception handling: (b) catching

- In Java, a *try statement* with one or more *exception handlers* (each introduced by the keyword *catch*) is used to indicate that a piece of code wishes to handle exceptions.
- If a throw statement is executed, control is transferred to the exception handler of the most-recently-entered try statement containing an appropriate exception handler:

```
BetterStack s = new BetterStack();
try {
    System.out.println(s.pop().toString());
}
catch( BetterStackException e ) {
    System.out.println("empty stack");
}
```

- The URL: <http://www.dur.ac.uk/~dcl0bjc/Java/a.taste.of.java/code/UseBetterStack.java> contains a complete Java application that uses the `BetterStack` class.

2.10 Package declarations

- In Java, a *package* is a collection of classes. You can indicate that a class belongs to a particular package by a *package declaration* at the start of the file:

```
package myutils;
```

- It is possible for Java source code to refer to classes in different packages even if they have the same name. For example, suppose some code requires to use both:
 - class `Stack` from the package `java.util`
 - class `Stack` from the package `myutils`

It can do this using the names `java.util.Stack` and `myutils.Stack`.

- Java specifies a convention for generating globally unique names, e.g., `UK.ac.dur.dcl0bjc.myutils.Stack`

3 Java's packages

3.1 Packages of the Core API of JDK 1.0.2

- In order to support programming in Java, many packages have already been constructed.
- Here is a list of the packages that belong to the Core API of JDK 1.0.2:

<code>java.applet</code>	used to support the writing of applets
<code>java.awt</code>	<i>abstract window toolkit</i> which allows a GUI to be written without regard to the platform(s) on which the program will run
<code>java.io</code>	reading/writing streams, files, pipes
<code>java.lang</code>	defines <code>Object</code> , <code>Class</code> , <code>Thread</code> , <code>Math</code> , <code>Boolean</code> , <code>Character</code> , <code>Integer</code> , <code>Float</code> , <code>Double</code> , <code>String</code>
<code>java.net</code>	used to support network programming
<code>java.util</code>	includes <code>BitSet</code> , <code>Date</code> , <code>Dictionary</code> , <code>HashTable</code> , <code>Random</code> , <code>Stack</code> , <code>Vector</code>

- These packages are documented at: <http://java.sun.com/products/jdk/1.0.2/api/>.

3.2 Handling strings

- In the package `java.lang`, there are two classes that store and manipulate character data: `String` is used for immutable strings and `StringBuffer` is used for mutable strings.
- Objects of the class `String` are usually more efficiently handled than `StringBuffers`, and they can be shared.
- `String` includes the following methods:
 - `length`, `charAt`, `toLowerCase`, `toUpperCase`, `equals`, `equalsIgnoreCase`, `compareTo`, `startsWith`, `endsWith`, `indexOf`, `lastIndexOf`, `substring`, `trim`, `concat`, `replace`
- `StringBuffer` includes the following methods:
 - `length`, `capacity`, `charAt`, `setCharAt`, `insert`, `append`, `toString`

3.3 An example of processing a string

- Here is a method that uses the classes `String` and `StringBuffer`:

```
public class StringExtras {
    public static String reverse(String source) {
        int charNum;
        int numChars = source.length();
        StringBuffer temp = new StringBuffer(numChars);
        for (charNum = (numChars-1); charNum>=0; charNum--) {
            temp.append(source.charAt(charNum));
        }
        return temp.toString();
    }
    ...
}
```

3.4 Providing a GUI

- The package `java.awt` consists of many classes forming the Abstract Window Toolkit (AWT). It provides GUI components such as: buttons, checkboxes, lists, menus and text areas.
- It also includes containers (such as windows and menu bars), and higher-level components (such as dialog boxes, including dialog boxes for opening or saving files).

- There are also classes for basic drawing operations, and for manipulating images, fonts and colours, and for handling events such as mouse clicks.

3.5 An example of providing a GUI

- Suppose we want to construct a Java applet that has a button and a textfield such that each time the button is pressed the textfield is updated to show the current date and time.
- With JDK 1.0.x, we can override the method `handleEvent` of the class `java.awt.Component`:

```
import java.applet.Applet; import java.awt.*; import java.util.Date;
public class ButtonClock extends Applet {
    private TextField tTextField = new TextField("hello", 35);
    private Button tButton = new Button("get date and time");
    public void init() {
        add(tButton); add(tTextField);
    }
    public boolean handleEvent(Event rEvent) {
        if (rEvent.target.equals(tButton) && rEvent.id == Event.ACTION_EVENT) {
            Date tDate = new Date(); tTextField.setText(tDate.toString());
            return true;
        }
        return false;
    }
}
```

- JDK 1.1.x has a better way of handling events. Here is a better version of `ButtonClock`:

```
import java.applet.Applet; import java.awt.*; import java.util.Date;
import java.awt.event.*;
public class ButtonClock11 extends Applet {
    private TextField tTextField = new TextField("hello", 35);
    private Button tButton = new Button("get date and time");
    private ButtonListener tButtonListener = new ButtonListener(tTextField);
    public void init() {
        tButton.addActionListener(tButtonListener);
        add(tButton); add(tTextField);
    }
}
class ButtonListener implements ActionListener {
    public ButtonListener(TextField rTextField) {
        iTextField = rTextField;
    }
    public void actionPerformed(ActionEvent rActionEvent) {
        Date tDate = new Date(); iTextField.setText(tDate.toString());
    }
    private TextField iTextField;
}
```

- Some details about converting programs to using the AWT API of JDK 1.1.x are at:
<http://www.javasoft.com/products/JDK/1.1/docs/guide/awt/HowToUpgrade.html>
<http://java.sun.com/products/JDK/1.1/docs/guide/awt/designspec/events.html>

3.6 More information about `ButtonClock`

- With an appropriate Java-aware WWW browser, you can execute the above applets via:
<http://www.dur.ac.uk/~dcl0bjc/Java/a.taste.of.java/code/ButtonClock10x.html>
<http://www.dur.ac.uk/~dcl0bjc/Java/a.taste.of.java/code/ButtonClock11x.html>

- A more involved example of Java source code that creates a user interface can be displayed by accessing the URL <http://www.dur.ac.uk/~dcl0bjc/Java/a.taste.of.java/code/SAT.java>. Because of the way in which it has been written, this source code can be run either as a Java applet or a Java application. You can run the SAT applet by accessing the URL: <http://www.dur.ac.uk/~dcl0bjc/Java/a.taste.of.java/code/SAT.html>.

3.7 JDK 1.1.x

- The final release of JDK 1.1.2 is now available for Windows 95/NT and Solaris (sparc and x86). A version of JDK 1.1.x for the Mac is planned for Q2 1997. For more details about JDK 1.1.x, see: <http://java.sun.com/products/jdk/1.1/>.
- The new features of JDK 1.1.x include: internationalization, improved security, signed Applets, AWT enhancements, JavaBeans API, networking enhancements, IO enhancements, JAR (Java ARchive), Remote Method Invocation, object serialization, reflection, JDBC (Java DataBase Connectivity), inner classes, more uses for the `final` keyword, a new native method interface, `Byte`, `Short`, `Void`, `BigInteger` and `BigDecimal` classes, and performance enhancements. The package `java.io` now has character streams. All the new APIs form part of the Core API of JDK 1.1.x. For details, see: <http://java.sun.com/docs/books/javaprogram/1.1Update.html> and <http://java.sun.com/products/jdk/1.1/docs/relnotes/features.html>.

3.8 Other APIs

- **Internet Foundation Classes (IFC)**: In December 1996, Netscape released an API called the *Internet Foundation Classes*. It includes drag-and-drop, built-in animation, overlapping resizable windows. For more details, see: <http://developer.netscape.com/library/ifc/>.
- **Application Foundation Class (AFC)**: In April 1997, Microsoft released an API called the *Application Foundation Class*. Written in Java and built in terms of the AWT, it includes GUI components such as toolbars, tabbed dialog boxes and tree controls. Microsoft say that the AFC may be redistributed free by application developers with their products on any Java-capable platform. The final release is scheduled for later in the year. This is the same schedule as Internet Explorer 4.0. For more details, see: <http://www.microsoft.com/java/resource/afc.htm>.
- **Java Foundation Classes (JFC)**: In April 1997, Sun, Netscape and IBM announced that they are to release an API called the *Java Foundation Classes*. They plan to “bring together the best of AWT with the best of IFC to create a unified set of APIs for developing robust, cross-platform Java applications”. The JFC is “scheduled to be available in the next release of the Java Development Kit”: presumably this means JDK 1.2. For more details, see: <http://java.sun.com/products/jfc/>.
- **Standard Extensions**: Besides the Core API, Sun is also defining and publishing *Standard Extensions*. A broad overview of the current and forthcoming APIs for Java, both for Core libraries and for Standard Extension libraries, can be seen at: <http://java.sun.com/products/api-overview/>.

4 Other points

4.1 Security issues

- Java ensures some problems do not arise: no pointer arithmetic, restricted forms of casting, no indexing outside arrays, no heap and stack overflows. A Java application has unrestricted access to local files, to the network, and to system calls. However, the environment of an applet is controlled by the WWW browser: there is no access to local files in Netscape Navigator, whereas HotJava users can configure read and write access.
- Security is a moving target as loopholes get discovered and fixed. For more details, see: <http://java.sun.com/sfaq/index.html>. For other papers, see: <http://java.sun.com/security/>.
- The Open Software Foundation has several papers on Java including one entitled *Security Features of Java and HotJava*. See: http://www.osf.org/mall/web/java_rpt.htm.
- Princeton University's Department of Computer Science has detected many of Java's security problems. See *News* and *Frequently Asked Questions* at: <http://www.cs.princeton.edu/sip/>. David Hopwood of Oxford University has detected other bugs.
- For details about *Java Security: Hostile Applets, Holes and Antidotes* by Gary McGraw and Edward Felten (ISBN 0-471-17842-X, \$19.95), see: <http://www.rstcorp.com/java-security.html>.

4.2 Other ports of the JDK

- HP has ported the JDK to HP-UX 10.01, 10.10 and 10.20 for the HP 9000 systems: <http://www.hp.com/gsyinternet/hpjdk/>.
HP has a pre-release version of JDK 1.1.2 for the HP 3000 systems: <http://jazz.external.hp.com/src/java/>.
- Look at: <http://www.blackdown.org/java-linux.html> for details of how to obtain the port of the JDK to Linux. This WWW page also contains a Java-Linux HOWTO and details about how to run HotJava, Java Workshop, Jeeves and the JDBC with the Linux port.
- IBM has released ports of the JDK for OS/2, OS/390, AIX and Windows 3.1, and work continues on a port to OS/400. The port of JDK 1.0.2 to Windows 3.1 is called *ADK for Win 3.1*: <http://ncc.hursley.ibm.com/javainfo/hurindex.html>. IBM are unlikely to upgrade ADK to JDK 1.1.x. IBM's main Java page is at: <http://www.ibm.com/Java/>.
- SGI has ported JDK 1.1.2: http://www.sgi.com/Fun/Free_webtools.html.
- The Open Software Foundation has ported JDK 1.0.2 to 7 platforms: <http://www.gr.osf.org/java/>.
- For details of other ports, see: <http://java.sun.com/products/jdk/jdk-ports.html>.

4.3 Cloning the JDK

- One clone of JDK 1.1.1 is *Kaffe* from Tim Wilkinson. Kaffe currently interprets bytecodes on Intel, SPARC, the PowerPC, Motorola's 68K, MIPS and Alpha. Supported OS include SunOS 4.1.x, Solaris 2.x, Irix, Linux, Windows 95 and AmigaOS. On most platforms, Kaffe also has a just-in-time compilation system which translates the bytecode to native machine code on a method-by-method basis as the application is executed. "This has demonstrated speedups of 10 times that of Sun's JDK, and is general[ly] about half the speed of optimised C code." See: <http://www.kaffe.org>.
- Another clone of the JDK is *guavac* produced by Effective Edge Technologies. Written in C++, it "has been tested under Solaris 2.x, SunOS 4.x, DEC OSF, Iris 5.3, HP/UX 9, and Linux 1.x using gcc-2.7.2 and libg++-2.7.1 and binutils-2.6, but other operating systems

should work without any major changes”. See: <ftp://summit.stanford.edu/pub/guavac> or: <http://www.cs.berkeley.edu/~engberg/guavac/>.

- Andreas Krall has released version 0.1 of *CACAO*, a 64 bit JIT system for an Alpha running Linux or Digital Unix. See: <http://www.complang.tuwien.ac.at/java/cacao/>.

4.4 Tools to support Java

- Two different modes for emacs are at the following URLs:
<ftp://ftp.python.org/pub/emacs/index.html>
<ftp://sunsite.doc.ic.ac.uk/packages/java/pub/java/contrib/emacs>
- A debugger (jdb) and a documentation tool (javadoc) form part of Sun’s JDK.
- Sun’s *Java Workshop* (written in Java) for Solaris 2.x and Windows 95/NT includes: Project Manager, Visual Java (GUI builder), Source Editor, Build Manager, Debugger, Source Browser, Project Tester, Portfolio Manager. For more details (including a free 30-day trial offer), see: <http://www.sun.com/workshop/java/>.
Until 31st March 1997, the list price is reduced from \$295 to \$99. In the UK last year, there was an educational offer of £810 for a 25-user pack.
- HP is working with Sun to provide Sun’s *Java Workshop* for HP-UX at \$99 in Q2 1997: <http://www.hp.com/csopress/97mar12g.html>.
- For reviews of Java integrated development environments, see:
<http://www.andromeda.com/people/ddyer/java/Reviews.html>,
<http://www.javaworld.com/javaworld/jw-06-1997/jw-06-devtools-review.html>
and <http://www.gamelan.com/special/javaides.html>.

4.5 Some Integrated Development Environments

- *Borland C++ 5.0* includes Java development tools such as *AppExpert*, a visual debugger, a project manager and *AppAccelerator* (a just-in-time compiler). Borland is producing a complete product called *JBuilder* (previously called *Latte*). Borland has released a preview version (which complies with JDK 1.1). See: <http://www.borland.com/jbuilder/>.
- Symantec has released *Café* for Windows 95/NT and the Macintosh. Based in part on the *Symantec C++ 7.2 IDDE*, it includes a Java compiler, browser, Java-aware editor, class editor, hierarchy editor and debugger. More recently, they have released for the same platforms “a complete Rapid Application Development environment for Java” called *Visual Café*. For Windows 95/NT, there is also a *Pro* version which provides support for database servers. See: <http://cafe.symantec.com>.
- Rogue Wave has an IDE called *JFactory*. It is available for Windows 95/NT, Solaris 2.x (sparc), HP-UX 10.x, OS/2. See: <http://www.roguewave.com/products/java.html>.
- For Windows 95/NT, there is an IDE for Java called *Kawa*. For more details, see: <http://www.tek-tools.com/kawa/>.
- Metrowerks has support for Java in their development environment *CodeWarrior* for Windows 95/NT and the Macintosh. See: <http://www.metrowerks.com/products/>.
- Asymetrix has an IDE for Windows 95/NT called *SuperCede*. They claim to have “the fastest and easiest-to-use” IDE. It is available free for 15 days and then costs \$109.95. The compiler can also generate x86 code. See: <http://www.supercede.com>.
- Microsoft has a product for Windows 95/NT called *Visual J++*. Version 1.1 claims to compile at over 10,000 lines per second, and includes wizards to help developers reuse existing ActiveX controls and to create new ones. See: <http://www.microsoft.com/visualj/>.

- Microsoft's *Visual Studio 97* (previously known as *Boston*) contains the latest versions of Visual Basic (5.0), Visual C++ (5.0), Visual J++ (1.1), Visual FoxPro (5.0) and Visual InterDev (1.0). Java, C++ and InterDev all share the common development environment called *Developer Studio*. See: <http://www.microsoft.com/vstudio/>.
- Marimba has produced *Bongo*. It costs \$495. See: <http://www.marimba.com/products/>.
- For Windows 95/NT, Penumbra Software has produced *Mojo*. See: <http://www.penumbra.com/mojos.htm>.
- For the Macintosh, Roaster Technologies has produced *Roaster*. See: <http://www.roaster.com>.
- Visix Software has a product called *Vibe*. Versions are now available for Windows 95/NT, Solaris 2.5, Linux and AIX, with Irix "coming soon", betas for MacOS and OS/2 and plans for HP-UX 10.20 and Digital Unix. See: <http://www.visix.com/products/vibe/>.
- Sybase has released a second beta of *PowerJ* (also known as *Jato* and previously known as *Starbuck*) with a final release planned for mid-1997. This is a RAD Java development tool that includes JavaBeans and ActiveX component integration. PowerJ also includes *jdbcConnect* which offers direct access to databases and a Visual SQL query editor. See: <http://www.powersoft.com/products/internet/javatool/>.
- IBM has released (for Windows 95/NT) a beta of their RAD environment for Java called *VisualAge for Java*. For more details, see: <http://www.software.ibm.com/ad/vajava>.
- SGI has released an IDE called *Cosmo Code*. For more details, see: <http://cosmo.sgi.com/code/>.

4.6 Compiling other languages into bytecodes

- It is possible to translate other languages into bytecodes. Robert Tolksdorf has a list of translators that have been or are being produced: <http://grunge.cs.tu-berlin.de/~tolk/vmlanguages.html>. His list includes the following languages: Ada95, Basic, CLIPS, Eiffel, Lisp, Logo, NetRexx, Prolog and Scheme.

4.7 Books on Java

- A very comprehensive list of the large number of Java books being written is at: <http://lightyear.ncsa.uiuc.edu/~srp/java/javabooks.html>. Other lists of books are at: <http://www.statenislandonline.com/java/>, <http://sunsite.unc.edu/javafaq/books.html> and <http://java.sun.com/nav/read/books.html>.
- Although the book *Teach Yourself Java in 21 Days* by Laura Lemay and Charles L. Perkins has numerous errors, it is quite a good book for getting started with Java. However, it does not cover JDK 1.1.x. For details, see: <http://www.mcp.com/samsnet/books/030-4/030-4.html>. For my list of errors in this book, see: <http://www.dur.ac.uk/~dcl0bjc/Java/>.
- *Java in a Nutshell: A Desktop Quick Reference for Java Programmers* by David Flanagan (from O'Reilly and Associates) is excellent for those who know C++ (or possibly just C). The second edition of this book revised for JDK 1.1.x will appear in May 1997 and will cost about £14.95 or \$19.95. The book includes useful summaries of various aspects of the language and the packages. Its ISBN is 1-56592-262-X. See: <http://www.ora.com/catalog/javanut2/>. *Java in a Nutshell: Deluxe Edition* is the same book but with a CD-ROM containing the text of four other books. It will be published in June 1997, will have an ISBN of 1-56592-304-9, and will cost about \$69.95. See: <http://www.ora.com/catalog/javadeluxe/>. For other books on Java from O'Reilly, see: <http://www.ora.com/publishing/java/>.
- SunSoft Press has published some books on Java. See: <http://www.sun.com/smi/ssoftpress/>.

- Addison-Wesley (<http://www.aw.com/cp/javaseries.html>) has published several books in their Java series. The most important ones are:
0-201-63451-1: *The Java Language Specification*
0-201-63458-9: *The Java Class Libraries: An Annotated Reference*
Currently, these two books only cover JDK 1.0.2 and not JDK 1.1.x. The texts of some of the books of this series are available through the WWW at the URLs given below and some form part of Microsoft's Visual J++ product.

4.8 Primary resources on Java

- The most important URL for information about Java is: <http://java.sun.com>.
- In particular, Sun's Java documentation is at: <http://java.sun.com/docs/>.
- A definition of the 1.0 language is at: <http://java.sun.com/docs/books/jls/html/>.
- A definition of the 1.0.2 API is at: <http://java.sun.com/products/jdk/1.0.2/api/>.
- A definition of the 1.1.x API is at: <http://java.sun.com/products/jdk/1.1/docs/api/packages.html>.
- A Java tutorial is at: <http://java.sun.com/docs/books/tutorial/>.
- To download the JDK, use: <http://java.sun.com/nav/download/>.
- During 1996, Sun had a forum for Java at: <http://java.sun.com/forum/securityForum.html>.
- Sun has produced a list of Java FAQs at: <http://java.sun.com/docs/faqindex.html>.

4.9 Other resources on Java

- Another Java FAQ is at: <http://sunsite.unc.edu/javafaq/javafaq.html>.
- An unofficial list of errata in the main Java documents is at:
<http://www2.vo.lu/homepages/gmid/java.htm>.
- A set of *How Do I* pages is available from: <http://www.digitalfocus.com/digitalfocus/faq>.
- Some useful papers on Java from ICON Computing are at: <http://www.iconcomp.com/papers/>.
- *Cafe Au Lait* contains news about Java. It is updated daily. See: <http://sunsite.unc.edu/javafaq/>.
- Mentor Software Solutions has a weekly summary of Java news called *DigitalEspresso* at:
<http://www.mentorsoft.com/DE/>.
- There is a weekly Java tutorial at: <http://netday.iworld.com/devforum/javajolt.html>.
- Probably the best monthly magazine about Java is available at: <http://www.javaworld.com>. Other monthly magazines are at: <http://www.javazine.pt> and <http://www.sigs.com/jro/>. A list of press reports is at: <http://www.javaworld.com/javaworld/netnews/netnews.index.html>.
- There are lists of Java resources at: <http://www.gamelan.com>, <http://www.java.co.uk>, http://mini.net/cetus/oo_java.html, <http://www.acl.lanl.gov/Java>, <http://java.motiv.co.uk> and <http://www.december.com/works/java/info.html>.
- The Usenet News newsgroups on Java are: `comp.lang.java.advocacy`, `comp.lang.java.announce`, `comp.lang.java.beans`, `comp.lang.java.databases`, `comp.lang.java.gui`, `comp.lang.java.help`, `comp.lang.java.machine`, `comp.lang.java.misc`, `comp.lang.java.programmer`, `comp.lang.java.security`, `comp.lang.java.softwaretools` and `comp.lang.java.tech`.

- A group called *iJUGGL*, the Independent Java Users Group for Greater London, has a WWW page at: <http://www.ijuggl.org.uk>.
- The programming languages subcommittee of ISO has formed a Java Study Group. For more details, see: <http://www.dkuug.dk/JTC1/SC22/JSG/>.

4.10 Not just a programming language

- Sun Microelectronics is producing a chip that can process Java bytecodes: *picoJAVA*. It will be suitable for cellular phones, printers and other consumer and peripheral markets. The *microJAVA* and *ultraJAVA* chips will be more advanced, e.g., handling graphics and multimedia. For more details, see: <http://www.sun.com/sparc/java/>.
- Sun has produced *JavaOS* which provides a standalone Java environment, i.e., a processor does not need an operating system (other than JavaOS) in order to run Java applications. It is currently available for sparc, x86 and ARM processors. For more details, see: <http://java.sun.com/products/javaos/>.
- Sun has produced a network computer called a *JavaStation*. This is a disc-less machine that understands Java. So, it has no disc drives, no floppy, no CD-ROM, no slots and no jumpers. For more details, see: <http://www.sun.com/javastation/>.
- Sun are to produce *JavaPC*: this is software that converts an old PC into a network computer. They say that JavaPC will be available in Autumn 1997 and will sell for under \$100. For more details, see: <http://java.sun.com/pr/1997/april/pr970402-07.html>.

5 Conclusions

- It is nice to have a programming language that supports object-oriented programming, exception handling and concurrency.
- Also, it is easy to build GUIs.
- Many of the problems with the C++ language are avoided.
- Programs are instantly portable across platforms.

The text of this overview is updated from time to time. The latest version is available at: <http://www.dur.ac.uk/~dcl0bjc/Java/>. This WWW page also contains other documents on Java including a tutorial on how to write Java programs and some notes comparing Java and C++.