Introduction to Java™
Version 3.0

Copyright (c) 1997,1998,1999 N Afshartous - All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization.

Trademarks: Java, JDK, Java Development Kit are trademarks of Sun Microsystems. Mosaic, NCSA Mosaic are proprietary trademarks of the University of Illinois.

Disclaimer: Although great care has been taken to ensure the accuracy and quality of these materials, all material is provided without any warranty whatsoever, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.
About the author: Niki Afshartous has been teaching corporate technical training courses for five years. Previously he was at the MIT Lincoln Laboratory as a systems programmer. He holds a Ph.D. degree in computer science from New York University where his research focus was on concurrent programming languages.

Dedication: For Jenny

Acknowledgement: Thanks to Maki Takeuchi for carefully proof reading earlier versions of this course.

Comments, questions? Don’t hesitate to contact the author.
Email: afshar@cs.nyu.edu
WWW: www.cs.nyu.edu/afshar
Contents

1. Introduction
2. Types and Variables
3. Statements and Control Flow
4. Reading Input
5. Classes and Objects
6. Arrays
7. Methods
8. Scope and Lifetime
9. Utility classes
10. Introduction to Object-Oriented Analysis and Design
Chapter 1: Introduction

After this chapter you will be able to:

- Describe the history of the Web
- Identify the impact of Java™ on the Web
- List the design goals of the Java™ team
History of WWW

• (Early nineties) Web was invented by Tim Berners-Lee who applied hypertext to make the internet user-friendly

• (1994) Most people were using the non-commercial web-browser Mosaic. Mosaic was partially developed by Marc Andressen for $6.85/hr as an undergrad on work-study.

• Later Andressen co-founded Netscape

• (1995) Microsoft holds a major press conference to announce an Internet strategy
Initial Architecture of WWW

- HTML - Hypertext Markup Language
- CGI - Common Gateway Interface
- WWW limitation: only server side execution using CGI.
Enter Java™

• Sun introduced Java™ at the Sun World conference on May 25, 1995
• Java™ initially made client-side execution on the Web possible!
• Currently, the focus has been on server-side Java™
Terms

• **Applet** - A Java™ program that executes in a virtual machine embedded in web browser

• **Servlet** - A Java™ program that executes in a virtual machine embedded in a web server
WWW + Client-Side Java™

- A Web page can be downloaded along with Java™ bytecode
- The bytecode is executed by a virtual machine embedded inside the web-browser
WWW + Server-Side Java™

- With Java™ on the server, no virtual machine is needed in the browser
Javascript

• Netscape subsequently released Javascript (originally called Livescript)
• The Java™ name was licensed from Sun Micro
• Javascript also facilitates client-side execution on the web
• Javascript is embedded within HTML
Architecture of WWW + Javascript

• A Web page can contain both HTML and Javascript

• The Javascript code is executed by an interpreter embedded inside the web-browser
Java™ ............. VS .......... Javascript

- Created by Sun Micro
- Can be used stand-alone or with the web
- Source compiled into bytecode
- Object-oriented
- Has facilities to support large applications
- Variables and expressions are statically typed
- Has threads for concurrent programming
- Fast exec but slow download

- Created by Netscape
- Tied to the web
- Source directly interpreted
- Object-based (no inheritance)
- No facilities for large applications
- Variables and expressions are dynamically typed
- Limited support for threads
- Slow exec but fast download
Summary of Java™ and Javascript

• Java™ is a powerful general purpose programming language
  – web server
  – web client
  – stand-alone

• Javascript facilitates making web pages more dynamic
  – web client
    • validating form data
Java™ Design Goals

• Simple
• Object-oriented
• Distributed
• Robust and secure
• Architecture neutral and portable
• Interpreted
• High-performance
• Multithreaded
• Dynamic
Simple

• $\text{Java}^{\text{TM}}$ designed as close as possible to C++ to look familiar

• $\text{Java}^{\text{TM}}$ omits confusing and complex features of C++
  – operator overloading
  – multiple inheritance
  – extensive automatic coercion's

• $\text{Java}^{\text{TM}}$ incorporates:
  – garbage collection
Object-oriented

- Focuses design on data and the operations on the data
- Helps to provide reusable software components
Distributed

- Java™ has library support for TCP/IP protocols
  - http
  - ftp
- Java™ applications can access remote objects by using URLs
- JDBC (Java Database Connectivity) kit allows access to remote databases
Architecture neutral

• Java™ compiler generates an architecture neutral object file format (bytecode)
• Bytecode can be executed on any platform where there is a virtual machine
Robust and secure

• Compile-time type checking
• Run-time checking (i.e. array bounds)
• Hard to write viruses in Java™ (no pointers)
• Bytecode is verified before it is run
Interpreted

• Java™ bytecode executed by an interpreter (virtual machine VM)

• VM can be stand-alone or embedded inside a web browser

• Other interpreted languages
  – scripts
  – Lisp
  – SmallTalk
  – Basic (Visual Basic)
Interpreted cont,

source (.java) → compiler → bytecode (.class)

interpreter (virtual machine)
High-performance

- Interpreting bytecode is more efficient than interpreting a source file (script).
- Just-in-time compilation (JIT) makes Java™ even faster. A virtual machine with JIT will translate bytecode into native executable format.
Multithreaded

- Java™ has threads and primitives for synchronization.
- Having threads facilitates writing programs that incorporate animation.
Dynamic

- Java™ was designed to adapt to an evolving environment.
- New code can be dynamically linked into a running system.
Free!

- Java Development Kit available from Sun Microsystems (java.sun.com)
- Consists of:
  - `javal`: compiler
  - `java`: virtual machine (bytecode interpreter)
  - `javadoc`: a program that generates HTML doc from source code comments
  - other miscellaneous programs
References

• The Java Language: An Overview (a.k.a. “The Java™ Whitepaper”)
at java.sun.com