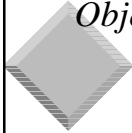



Core Java Course
Introduction to Java
(Chapter 1)


Dr. Jeff Blessing
MSOE



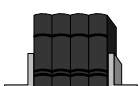
Java
Object-Oriented Programming
(CS-4811)



Prof. Jeffrey J. Blessing
Office: MSOE, CC-27a
Phone: 277-7194
Email: blessing@msoe.edu
<http://www.msoe.edu/~blessing>




Logistics



- ❖ Schedule
- ❖ Text, code, & notes:
 - ❖ www.horstmann.com/corejava.html
 - ❖ www.msoe.edu/~blessing/cs4811
- ❖ Compiler (JDK 1.3 compliant)
- ❖ Exams
 - ❖ Weekly Quizzes & programming assignments
 - ❖ Final Exam (week 11)


3



Ideal Candidates for the Course

- ☉ People with basic programming background
- ☉ Programmers coming from procedurally oriented languages (COBOL, Fortran, C)
- ☉ Visual Basic programmers
- ☉ Will sound familiar for C++ programmers
 - ❖ should focus on language differences
 - ❖ C/C++ programmers will have the best background for the course


4



Not So Ideal Candidates...

- ☹ People with no programming background
 - ❖ We do not have time to explain basics
- ☹ People with significant Java experience
- ☹ People interested in advanced Java features considered in the second course
 - ❖ Multithreaded programming
 - ❖ Distributed applications
 - ❖ JavaBeans

5



Agenda

- ❖ History of Java
- ❖ What is Java?
 - ❖ Major Java advantages and features
- ❖ Java misconceptions
 - ❖ Confusions about Java as programmer's universal remedy (or "magic/silver bullet")

6

History

- 1991 Small, portable language for consumer devices (Patrick Naughton, James Gosling)
 - ❖ "Green" project, intermediate code, VM
 - ❖ "Oak", Object-oriented, C++ based
- 1992 First product delivered - remote control
 - ❖ Sun is not interested!
- 1993 Unsuccessful attempts to market to others
- 1994 Internet gives birth to the Web
 - ❖ First Web browser from Mosaic (UIUC)

7

History (cont.)

- 1994 (cont.)
 - ❖ HotJava browser linked power of Java to the Internet
- 1995 Java adopted in Netscape 2.0
- 1996 Java officially released by Sun (JDK 1.0)
- 1997 Java has become more robust (JDK 1.1)
 - ❖ New libraries, new event model, printing
- 1998 Sun releases JDK 1.2 (renamed Java 2)

8

What is Java?

- ❖ Simpler than C++
 - ❖ eliminated "cryptic" syntax
 - ❖ improved language constructs (inheritance & polymorphism)
 - ❖ eliminated pointers, memory management, operator overloading (function overloading OK), templates, and global functions
- ❖ Not as simple as VB
- ❖ Small (40 - 205K run-time size)

9

What is Java? (cont.)

- ❖ An Object Oriented Language
 - ❖ everything is an object except for some *primitive* data types
- ❖ Internet friendly
 - ❖ native support for HTTP, FTP, URLs
- ❖ Safer to program
 - ❖ dynamic (run-time) type checking
 - ❖ built-in memory management

10

What is Java? (cont.)

- ❖ "Environmentally" safe
 - ❖ no memory corruption
 - ❖ no run-time stack overrunning
 - ❖ signed classes
- ❖ Secure
 - ❖ JVM filters all the resource access calls
 - ❖ Applets cannot access local resources at all
 - ❖ `SecurityManager` traps misbehaving apps

11

What is Java? (cont.)

- ❖ Portable and architecture independent
 - ❖ will work on any processor that has a Java interpreter - Java Virtual Machine
- ❖ Interpreted
 - ❖ *bytecode* files
 - ❖ *native* and *Just-In-Time* Compilers (JIT)
 - ☞ compilers may not be available for all platforms!

12

What is Java? (cont.)

- ❖ Integrated windowing & graphics
 - ❖ AWT (Abstract Windowing Toolkit)
 - ❖ Swing/JFC (Java Foundation Classes)
- ❖ Network capable
 - ❖ Client/Server model: *Applets/Servlets*
- ❖ Component Architecture
 - ❖ JavaBeans and EJBs allows for extensible software architectures

13

What is Java? (cont.)

- ❖ Real-time programming ready
 - ❖ support for multithreading
 - ☞ yet, not deterministic
- ❖ Productivity oriented
 - ❖ Programming chores are off-loaded to libraries
- ❖ Two different executable types
 - ❖ *Application* (i.e. Stand-alone)
 - ❖ *Applet* (i.e. Runs on a web page)

14

Misconceptions

- ❖ (Not) Easy to learn
 - ❖ extensive libraries requiring application/domain knowledge
- ❖ (Not) Easy programming environment
 - ❖ IDE (Integrated Development Environment) tools missing in the Java Development Kit (JDK)
 - ❖ hard to debug with `jdb` (Java Debugger)

15

Misconceptions (cont.)

- ❖ (Not) Universal Language
 - ❖ lacks many useful features (especially AWT)
 - ❖ support lags behind for some platforms
- ❖ (Not) Slow because it is interpreted
 - ❖ in many cases (networking) application is waiting for data to become available
 - ❖ JIT's make CPU-intensive processing much more efficient

16

Misconceptions (cont.)

- ❖ (Not) Will replace a PC with NC
 - ❖ most serious applications are large and would take long time to access over network
 - ❖ local storage is a necessity
 - ❖ writing high quality NC-compatible software can take years (Corel cancelled Java Office)

17



18