



THE UNIVERSITY of
NEW MEXICO



CS-259

Data Structures with Java

Welcome

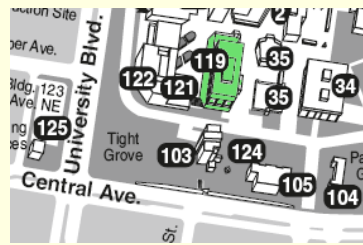
Instructor: **Joel Castellanos**

e-mail: joel@unm.edu

web: <http://cs.unm.edu/~joel/>

Course website:

<http://cs.unm.edu/~joel/cs259/>



8/28/2009

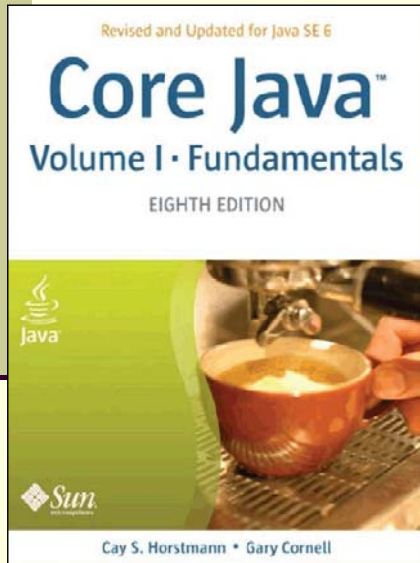
Introductions

- Your Name
- Your Background in Computer Programming: quickBasic, Visual Basic FORTRAN, C, C++, or hours and hours of WoW.
- UNM major, intended major, just taking one class....
- Why you are taking this class and what you expect to get from it.



2 of 19

Course Resources



Class website:

<http://cs.unm.edu/~joel/cs259/>

- Syllabus
- Lecture Notes
- Supplemental readings

Textbook:

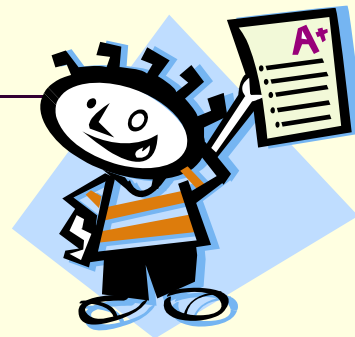
Core Java, Volume I-
Fundamentals (8th Edition) by
Cay S. Horstmann, and Gary
Cornell.

WebCT: <http://vista.unm.edu/>

- Grades
- Homework assignments

Grades

Midterm Exam	15%
Final Exam	15%
i-clicker quizzes	10%
Labs	30%
Projects	30%



- Attendance is required for both i-clicker and lab grades.



- We will use the i>clicker for quizzes every lecture.
- Your i>clicker needs to be registered with your name, and UNM NetID.
- If you have already registered your i>clicker last semester on the web for a different class, then you do not need to register again.
- Register your name as it appears in your UNM NetID.
- One i>clicker can be registered by different people in different classes.
- If your i>clicker registration number has rubbed off, see me after class.

5

i>clicker Web Registration

- <http://www.iclicker.com/registration/>



Register Your i>clicker

First Name:

Last Name:

Student ID:

Remote ID:



UNM NetID
(no @unm.edu)

**NOT Student ID:
100838716**

6



Quiz Question #1:

Do you have your i-clicker?

- a) Yes – I am ready to go.
- b) I bought one from the bookstore, but forgot it.
- c) My dog ate it.
- d) No – I did not get one yet.
- e) What is an i-clicker anyway?

7

Who Should Take This Course

- Someone who has studied Java in school, on a job or is self-taught, but does not feel ready to skip CS-152 and take CS-251.
- Someone who has solid programming experience, but does not know anything about Java. A person who knows C or FORTRAN well enough to write a program that calculates the first 50 prime numbers would probably be bored in CS-152, but, without any Java experience, would be lost on day-one of CS-251.
- An engineering, science or math student with junior or senior standing who generally gets As or Bs. Even if that student has never taken a programming class, he or she would probably find this class more interesting and appropriately challenging than the CS-152 & CS-251 series.
- Calculus is not required or used in this class, but a student, straight out of high school, who found AP calculus and AP physics fairly easy would be much happier in this class than in CS-152 & CS-251.

8

Homework: Due Wed, Aug 26

Read Chapter 1

- Java as a Programming Platform.
- Java Buzzwords.
- Java Applets and the Internet.
- A Short History of Java.
- ★ ■ Common Misconceptions about Java.



Wednesday: i>clicker quiz on the reading.

9

Java Rocks



- Advantages for Large Programs
- JUnit Testing
- JavaDocs
- Object-Oriented
- Platform-independent
- Secure
- Robust (emphasis on early checking for possible errors)
- Multithreaded
- Well suited for Automated Optimization



10

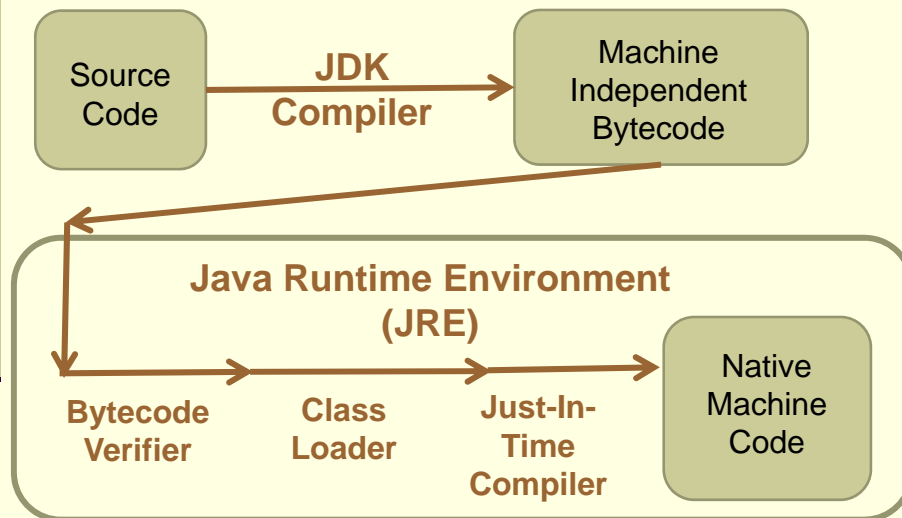
Compilation

A compiler translates human readable *source code* to *machine instructions*.

Java Source Code	Machine Instruction Mnemonics
<code>area = 0.5 * base * height;</code>	Load 0.5 Load 0x000017AB0 Multiply Load 0x000042331 Multiply Store 0x000112355

11

Java Compilation Process



12

Java Performance - Wikipedia

Programs written in Java have had a reputation for being slower and requiring more memory than those written in natively compiled languages such as C or C++.

However, Java programs' execution speed has improved significantly due to introduction of Just-In Time compilation (in 1997/1998 for Java 1.1), language features supporting better code analysis, and optimizations in the Java Virtual Machine itself (such as HotSpot becoming the default for Sun JVM in 2000).

13

Micro Benchmarks vs. Production Code

C/C++ (wins on small)

- Hand optimized, Pointers are quick
- Native floating-point operations.
- No error checking
- Quick startup

Java (often wins on big)

- C-style pointers make optimization hard in languages that support them (particularly fine grain parallelism)
- Adaptive optimization is impossible in fully compiled code.
- Auto *Garbage Collection* beats application programmer GC!

14

Structured Programming

```
//Find Prime Factors.
while (testNumber <= endTest)
{ if (n % testNumber == 0)
  { //testNumber divides n.
    System.out.println(testNumber);
    n = n / testNumber;
  }
  else
  { //testNumber does not divide n.
    testNumber++;
  }
}
```

15

OOP: Object-Oriented Programming

Do not be afraid of OOP.

Small Java programs are often not object-oriented.

OOP is a superset of structured programming .

Inside each method (a.k.a. subroutine) OOP uses structured programming style.

An object-oriented program may be viewed as a *collection of cooperating objects*,

In a conventional model, a program is seen as a list of tasks (subroutines) to perform in some order.

16

Object-Oriented Programming – part 2

- In OOP, each object is capable of receiving messages, processing data, and sending messages to other objects and can be viewed as an independent 'machine'.
- The actions (or "operators") on these objects are often part of the object.
- For example, in an employee database, there might be an Employee object with
 - **Fields** such as firstName, lastName, salary, ...
 - **Methods** (actions) such as alphabetize that looks at the lexicographical order of the last name, and if they are the same, then looks at the lexicographical order of the first name.

17

Object-Oriented Programming – part 3

- OOP is a programming paradigm that uses "objects" and their interactions to design computer programs.
- As software became increasingly complex, quality was often compromised and reuse became critical.
- OOP was, in part, developed to address these common problems by strongly emphasizing discrete, reusable units of programming logic.
 - Information hiding
 - Data abstraction
 - Encapsulation
 - Modularity
 - Inheritance

18

Class Ideas

- What is Java?
- What is Computer Programming?
- What is Object-Oriented Programming?

