Welcome (back) to CS1007!

Introduction to Computer Science in Java

Spring 2002

Section 001:	TR	2.40pm	-	3.55pm	301 Pupin
Section 002:	TR	11.00am	-	12.15pm	209 Havemeyer

Professor Elizabeth Sklar

email: *sklar@cs.columbia.edu*
web: *http://www.cs.columbia.edu/~sklar*
office: 464 Computer Science building (through Mudd)

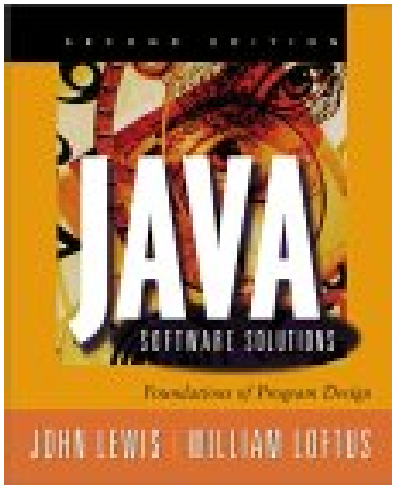
Class web page:
http://www.columbia.edu/~cs1007

last time...

- course objectives
- policies
- academic integrity
- resources
  - WEB PAGE: *http://www.columbia.edu/~cs1007*

textbook.

Java Software Solutions: Foundations of Program Design.
by Lewis and Loftus
Addison Wesley, JavaPlace edition, 2001
ISBN 0-201-75052-X



available at Labyrinth Books
(112th, between Broadway & Amsterdam)

assessment.

- out of 100 points
- 6 homework assignments (40 points total)
  - homework #1 (5 points) – due Thu Jan 31
  - homework #2 (7 points) – due Thu Feb 14
  - homework #3 (7 points) – due Thu Mar 7
  - homework #4 (7 points) – due Thu Mar 28
  - homework #5 (7 points) – due Thu Apr 11
  - homework #6 (7 points) – due Thu Apr 25
- 3 exams (60 points total)
  - midterm I (15 points) – Thu Feb 21
  - midterm II (20 points) – Thu Apr 4
  - final exam (25 points) – TBA (during exam period)
- *note that all dates are tentative and subject to change!*

recitations.

- sign up for a recitation (sign up sheet in class)

day	date	time	room
recitation #1	Mon	7.00pm – 8.00pm	252 ET
recitation #2	Mon	8.00pm – 9.00pm	252 ET
recitation #3	Tue	6.00pm – 7.00pm	407 Math
recitation #4	Tue	7.00pm – 8.00pm	407 Math
recitation #5	Tue	8.00pm – 9.00pm	407 Math
recitation #6	Wed	6.00pm – 7.00pm	252 ET
recitation #7	Wed	7.00pm – 8.00pm	252 ET
recitation #8	Wed	8.00pm – 9.00pm	252 ET

- ET = Engineering Terrace
- starting next week (week of Jan 28)
- this is where you will get back your homeworks

AcIS training sessions.

day	date	time	room
Fri	Jan 25	2.30pm – 4.30pm	252 ET
Fri	Jan 25	12.30pm – 2.30pm	252 ET
Mon	Jan 28	4.00pm – 6.00pm	252 ET
Tue	Jan 29	4.00pm – 6.00pm	252 ET
Fri	Feb 1	2.30pm – 4.30pm	252 ET
Fri	Feb 1	10.00am – 12.00pm	252 ET
Mon	Feb 4	4.00pm – 6.00pm	252 ET
Mon	Feb 4	12.00pm – 2.00pm	252 ET
Wed	Feb 6	4.30pm – 6.30pm	252 ET
Fri	Feb 8	2.30pm – 4.30pm	252 ET
Fri	Feb 8	12.30pm – 2.30pm	252 ET

- ET = Engineering Terrace

today's topics.

- computer basics
- creating your first program
- editing, compiling, linking, running
- `System.out`
- reading: *ch 1.1-1.5*

computer commands.

- computer follows commands
  commands = series of instructions
- you will learn how to *command* a computer
  command = program = write instructions
- you understand the commands,
  but does the computer?
  that's a question of cognition...
  $\rightarrow$ Artificial Intelligence, Cognitive Science

computer components.

- computer = hardware + software
- a computer is organized into *logical units*:
  - input
  - output
  - memory
  - arithmetic and logic (ALU)
  - central processing (CPU)
  - secondary storage

computer instructions.

- set of instructions = *program*
- types of instructions:
  - machine language
  - assembly language
  - high-level language (e.g., C, C++, Java)
- program is *compiled* into machine language and then *executed (ran)*
- *executing (running) program = job = process = task*

machine language.

- lowest level
  – numeric
- computer is comprised of zillions of *switches* or *relays*
  – switches = ON or OFF
  – relays = OPEN or CLOSED
- hardware position is abstracted into software as 1's and 0's
- 1's and 0's $\Rightarrow$ *base 2*, or *binary*

assembly language.

- medium level, but still pretty low; i.e., hard to read and understand
- “English” words and abbreviations
- examples:
  LOAD
  ADD
  SHIFT
  STORE

high-level languages.

- examples: C, BASIC, FORTRAN, Pascal, C++, Java, LISP, Scheme
- even more like “English”
- high-level languages are
  1. *compiled* into machine language or *object code*
  2. *linked* into *executable code*
  3. *executed* or *ran* as programs

language examples.

- machine language:
  +1300042774
  +1400593419
  +1200274027
- assembly language:
  LOAD BASEPAY
  ADD OVERPAY
  STORE GROSSPAY
- high-level language:
  grossPay = basePay + overTimePay;

Java.

- Java is an *object-oriented* language: it is structured around *objects* and *methods*, where a method is an action or something you do with the object
- Java programs are divided into entities called *classes*
- some Java classes are *native*
  but you can also write classes yourself
- Java programs can run as *applications* or *applets*

your first application.

“hello world”

- typical first program in any language
- output only (no input)

the application source code.

file name = hello.java

```
/*-------------------------------------------------------
   EISklar, 11-Sep-01, hello.java

   This class demonstrates output from a Java application.
-------------------------------------------------------*/
public class hello {
    public static void main ( String[] args ) {
        System.out.println( "hello world!\n" );
    } // end of main()
} // end of class hello()
```

System.out.

- like filling in graph paper
- *methods*

```
System.out.println( )
System.out.print( )
```

- *arguments*
  - those things inside the parenthesis ( )
  - one or more Strings, separated by “+” ’s
  - escape sequences: \n, \t
  - also called *parameters*
- example

```
System.out.println( "The quick" + ", brown " + "fox" );
```

things to notice.

- Java is CASE sensitive
- punctuation is really important!
- *whitespace* doesn't matter for compilation
- *BUT* whitespace DOES matter for readability and your grade!
- file name is same as class name

try it yourself.

1. log into CUNIX
2. create the application source code file,
   using the *emacs* (or *ejava*) editor
3. compile the source code,
   using the *javac* command
4. execute the program using the *java* command

quick and dirty UNIX.

- UNIX is an operating system,
  – *Linux* is a version of UNIX
- command-line interface
  – commands have options, also called *switches*
- here are some commands:

```
ls      -- list the files in the current directory
cp      -- copy a file
mv      -- rename a file
rm      -- delete (remove) a file
cd      -- change directory
pwd     -- show the current directory
man     -- help
chmod   -- change file protections
```

quick and dirty *emacs*.

- at the UNIX prompt: `unix> emacs hello.java`
- emacs is a “control key” editor
- here are some commands:

```
Ctrl-B          -- move cursor back
Ctrl-F          -- move cursor forward
Ctrl-P          -- move cursor to previous line
Ctrl-N          -- move cursor to next line
Ctrl-D          -- delete character under cursor
Ctrl-X Ctrl-S   -- save the file
Ctrl-X Ctrl-C   -- exit emacs
Ctrl-H          -- help
ESC             -- escape! gets you out of trouble!
```

to do.

- get the textbook, and read chapter 1.1 – 1.5
- sign up for a rectiation
- attend one of the AcIS training sessions
- try logging into your CUNIX account
- check out the class web page:
  http://www.columbia.edu/~cs1007

Have a good weekend!