

Welcome (back) to CS1007!

*Introduction to Computer Science
(Programming in Java)*

Section 001: TR 2.40pm - 3.55pm 417 Int'l Affairs Building
Section 002: TR 11.00am - 12.15pm 209 Havemeyer Hall

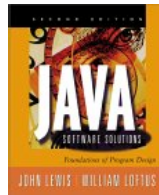
Professor Elizabeth Sklar

email: sklar@cs.columbia.edu
phone: 939 7021
office: 464 Computer Science Bldg

1

textbook.

Java Software Solutions: Foundations of Program Design.



by Lewis and Loftus
Addison Wesley
JavaPlace edition, 2001
ISBN 0-201-75052-X (ON BACK COVER)
available at Labyrinth Books
(112th, between Broadway & Amsterdam), from Sept 4

3

Please attend one of the AcIS training sessions:

“Introduction to UNIX for Java Programming”

day	date	time	room
Friday	Sept 7	10am-noon	252 ET
Friday	Sept 7	12.30pm-2.30pm	252 ET
Friday	Sept 7	2.30pm-4.30pm	252 ET
Monday	Sept 10	5.00pm-7.00pm	306 Butler
Tuesday	Sept 11	3.30pm-5.30pm	306 Butler
Wednesday	Sept 12	5.00pm-7.00pm	306 Butler
Friday	Sept 14	12.30pm-2.30pm	252 ET
Friday	Sept 14	2.30pm-4.30pm	252 ET
Monday	Sept 17	5.00pm-7.00pm	306 Butler
Wednesday	Sept 19	5.00pm-7.00pm	306 Butler

No need to register — just show up!

5

last time...

- course objectives
- policies
- academic integrity
- resources
 - WEB PAGE: <http://www.cs.columbia.edu/~sklar/cs1007>

2

assessment (updated dates!).

- homework assignments (40% total)
 - homework #1 out: TUESDAY SEP 11
 - homework #1 due: TUESDAY SEP 18
- 3 exams (60% total)
 - midterm I: 15% — THURSDAY OCT 4
 - midterm II: 20% — THURSDAY NOV 1
 - final exam: 25% — during exam period (DEC 14-21)

4

today's topics.

- computer basics
- creating your first program
- editing, compiling, linking, running
- applets and applications
- reading: *ch 1.1-1.5*

6

computer basics, 1: 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...
→ Artificial Intelligence, Cognitive Science

7

computer basics, 2: components.

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

8

computer basics, 3: 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*

9

computer basics, 4: 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 ⇒ *base 2, or binary*

10

computer basics, 5: assembly language.

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

11

computer basics, 6: 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*
 - then
 - (2) *linked* into *executable code*

12

computer basics, 7: languages example.

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

13

computer basics, 9: Internet.

- an autocratic system based on the *client – server* model
 - *server* tells client what to do
 - *client* does it
- commands may be executed at either the client or the server level
- Java tries to maximize execution at the client level
- Java runs inside a virtual machine (“JVM”) (to make it portable)

15

Java, 2: your first application.

“hello world”

- typical first program in any language
- output only (on input)

17

computer basics, 8: 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*

14

Java, 1: execution model.

- *application*:
client and server are the same computer
- *applet*:
server sends applet to client,
applet code runs on client
- applet example: Tron
<http://www.demo.cs.brandeis.edu/tron>

16

Java, 3: the application source code.

file name = hello.java

```
public class hello {  
  
    public static void main ( String[] args ) {  
  
        System.out.println( "hello world!\n" );  
    }  
}
```

18

Java, 4: 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

19

Java, 6: 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
```

21

Java, 8: the applet source code.

file name = hi.java

```
import java.applet.Applet;
import java.awt.*;

public class hi extends Applet {

    public void paint ( Graphics g ) {

        g.drawString( "hello world!",10,10 );
    }

}
```

23

Java, 5: try it yourself.

- log into CUNIX
- create the application source code file,
 - using the *emacs* (or *ejava*) editor
- compile the source code,
 - using the *javac* command
- execute an application using the *java* command
- execute an applet using the *appletviewer* command
 - OR a browser, like Netscape

20

Java, 7: quick and dirty *emacs*

- at the UNIX prompt: `unix> ejava 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!

22

Java, 9: the HTML source

file name = hi.html

```
<html>
<title>
sample applet page
</title>

the applet will be shown below...

<applet code="hi.class" width=400 height=400>
</applet>

</html>
```

24

to do.

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

Have a good weekend!