Index of /~cs61b/lectures/01

Name	Last modified	Size	Description
Parent Directory	24-Sep-98 21:22	-	
HelloWorld.java	24-Aug-98 08:23	1k	

CS61b: Lecture 1
Monday, August 24

Prof. Kathy Yelick E-mail: yelick@cory
Prof. Jonathan Shewchuck E-mail: jrs@cory
Profs and TAs E-mail: cs61b@pasteur

GOALS OF THIS COURSE

1) Writing good code **
2) Understanding abstraction, especially data abstraction
 **special emphasis this year
3) Designing efficient data structures and algorithms that use them
4) Learn Java

A DESIGN PROBLEM

You've just been hired by Inktomi, a company that builds Internet infrastructure. They need a data structure to look up URLs by keyword.
  * There are about 10 million documents on the Web (doubles every 5 months)
  * Inktomi has decided that ~100 thousand keywords are enough

What data structure should you use?
  How long will a lookup take?
  What about adding new keyword/URL pairs?
  What if we're worried about memory as well as time?

Note: good "content-based" retrieval is an open research problem.

HANDOUTS (4, also available from http://www-inst.eecs.berkeley.edu/~cs61b)

1) General Information
2) Lecture Notes #1 (this handout)
3) Homework 1 (due this Friday)
4) Lab 1 (Due during assigned lab. No preparation required this week.)

THE LANGUAGE OF OBJECT-ORIENTED PROGRAMMING

Objects: some data and operations that manipulate that data
Classes: factories for "generating" objects
Methods: an operation on an object or a class
Variables: names for the data in objects, classes, and methods
Object-Oriented: uses data abstraction (classes) and inheritance

ANATOMY OF A SMALL Java PROGRAM

```
/* In a file called HelloWorld.java */

class HelloWorld {
    public static void main (String[] args) {
       System.out.println("Hello, world");
    }
}
```

1. The Hello world class is a "factory" for HelloWorld objects.
2. The class knows how to do the "main" operation.
3. Calls the System class to get an output stream and do a print
   operation on it using the "Hello, world" string.

The classes are:
The objects are:
The methods are:

JAVA

Some differences you'll notice between Java and Scheme (or Java and Matlab)

1. Everthing in Java has a type, and you have to "declare" it.

   e.g., int x = 1;   vs. (let ((x 1)) ...)

   Can't use x without saying it's an int.

2. Java has a compiler.

```
       Scheme                 Java             Pascal, C, C++...
       ------                 ----             -----------------
 Scheme program (.scm)  Java program (.java)     C program (.c, .cc,...)
          |                      |                   |
          |  eval                | java              | gcc
          |                      v                   v
          v                .class files          .o files
        Answer                   |                   |
                                 | javac             | link
                                 v                   v
                               Answer            executable
                                                     |  run
                                                     v
                                                 Answer
```

ENROLLMENT:

Labs (in 277 Soda) and discussions sections start this week, i.e., today.
You need to attend your assigned lab (from Telebears scheduling) to:
1) secure your spot in the class
2) get an account form (needed for HW1 and Lab1)

You may only attend a lab for which you are officially enrolled.
If you are on the waiting list, you may only attend:
1) lab 21 8-10am on Tuesday
2) lab 22 8-10am on Wednesay
3) temporary lab A, Monday 9-11pm
4) temporary lab B, Tuesday 8-10pm

PREREQUISITES:

About 20 students currently enrolled have not satisfied the prereqs (cs61a or E77). You must drop the class or receive an F in the class. See the drop list outside of 379 Soda.

If you have not taken 61a or E77 at Berkeley, but feel you have taken something similar, fill out an appeal form in 393 Soda.

UNAUTHORIZED COLLABORATION

...also known as cheating, will be reported to the Office of Student Conduct

a) Helping someone interpret an error, for example, is encouraged, but follow the "No Code Rule:" Never have a copy of someone else's program in your possession and never give your program to someone else.
b) People who give their programs to others will also be reported.
c) If you receive a significant idea from someone, acknowledge it.
d) These rules also apply to groups.
e) It applies to homeworks and projects as well as exams.