

A decorative header consisting of two horizontal lines. The top line is a thick, solid green bar. The bottom line is a thin, light blue bar.

Introductory comments and boilerplate

Harvey Mudd College

CS 60

Principles of Computer Science
Spring 2003

Bob Keller, Professor
keller@cs.hmc.edu
621-8483

CS 60 Home Page

<http://cs.hmc.edu/courses/current/cs60>

- links to the syllabus and other good stuff
- will be updated constantly as we go

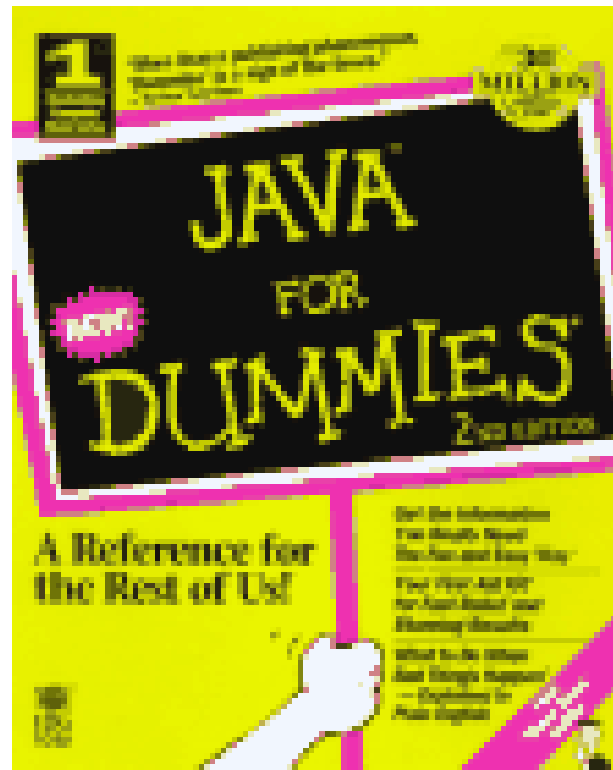
My Office Hours (1249 Olin):

- Note: 1249 is in the southwest corner of Olin
- Tuesday and Thursday, 1-3, Wed. 6:30-7:30, and
- By drop-in (frequently available)
- Look for "IN" vs. "OUT" letters
- By appointment:
 - email keller@cs.hmc.edu
 - phone 621-8483
- Crisis center: 621-2373

Text

- *Computer Science: Abstraction to Implementation* (aka "*Computer Science for Smart People*"*) by Robert M. Keller
- Available from CS Department office, three options:
 - \$30 for the 2001 edition (recommended)
 - web addition:
<http://www.cs.hmc.edu/~keller/cs60book>
Do not print major portions of book on HMC printers.
- Reading Assignment:
 - Chapters 1 - 3

*Boycott demeaning books!



Auxiliary Text

- Some kind of Java reference, e.g. what you used in CS 5 or equivalent. Or check any bookstore for something that looks appealing.
- Won't need Java for a couple of weeks.

Help with Computer Account

- You will be given an account on `turing.cs.hmc.edu`.
- For problems with your account, you will need to contact either:
 - our system administrator,
 - Andrew McDonnell (`amcdonne@cs.hmc.edu`).
 - or one of our staff members:
 - `staffnow@cs.hmc.edu`
- I (Bob Keller) don't have the privileges necessary to set your password, quota, etc.

How/Where to Login

- Room B102 is best
- Remote is possible, however:
 - Must use **secure** ssh client, **not telnet**
 - For further information please see:

http://www.cs.hmc.edu/tech_docs/qref/ssh.html

This will tell you how to get free client for your machine.

Strong Recommendation

- As quickly as possible:
 - Learn your way around UNIX
 - Learn to use Emacs (text editor)
 - Emacs is a powerful tool.
 - Using it can have life-long benefits.

Definition of Computer Science (CS)

Computer science involves synthesis and analysis of:

- Algorithms
- Information representations
- Communication processes
- Resource allocation methods
- Languages for all of the above

Role of CS

Computer science provides
the *logical infrastructure* for
the information-based society
(That's us, folks!).

CS Characteristics and Contrasts

- CS not a “study of nature” as such
- *We create what we study*
“The best way to predict the future is to invent it.”
Alan Kay
- “Abstractions” are often a product
(e.g. □Java awt: *Abstract Window Toolkit*)

A Misconception

- Computer Science is just the study of computers

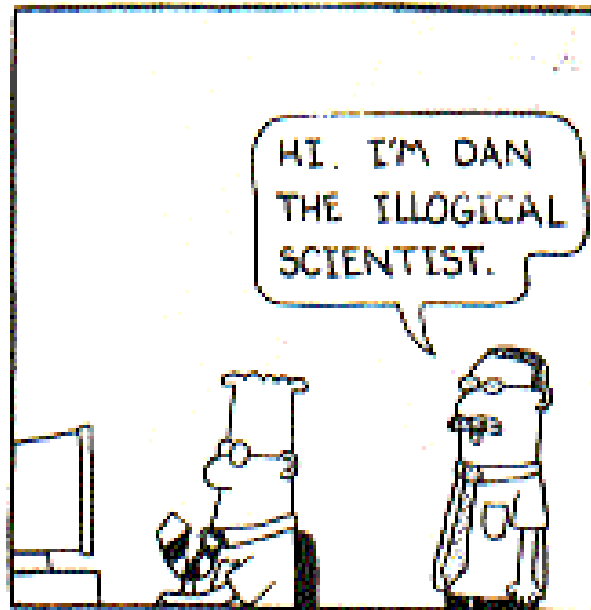
Analog: Surgery is "knife science".

- There is *some* of that, but CS is more generally about information and computation.

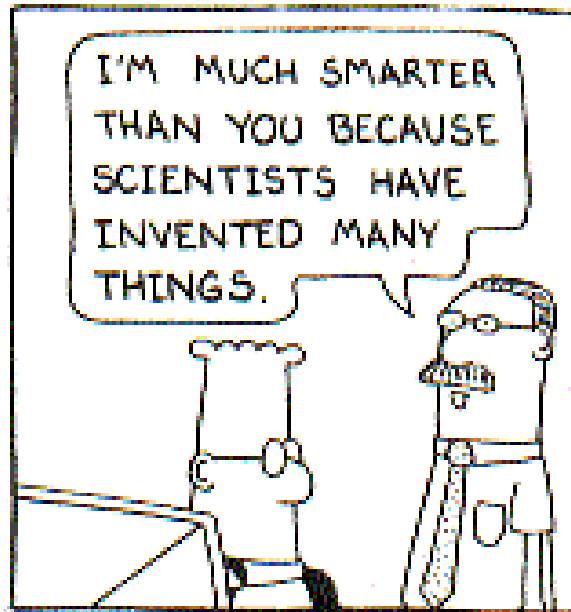
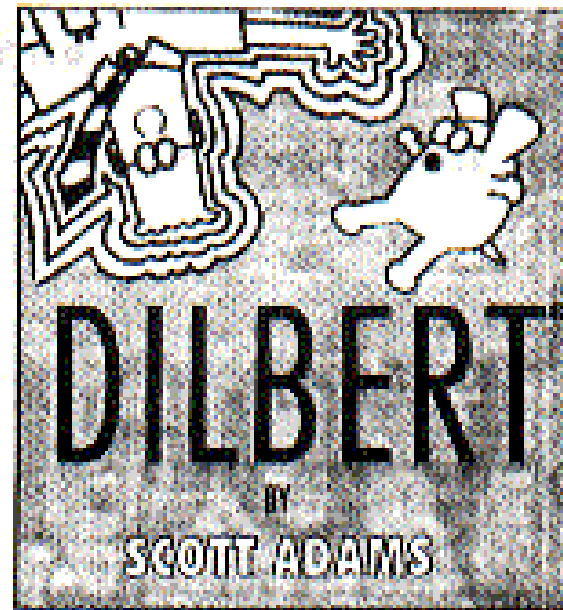
Misconceptions (continued)

- Computer Science is just a "service" to other fields, not a "real science".

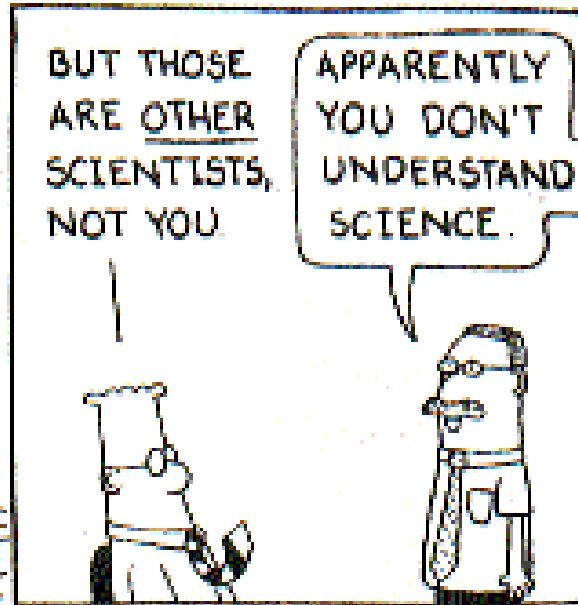
Computer Science is an independent intellectual discipline that *happens* to enjoy applications to many other disciplines.



S. Adams E-mail: SCOTTADAMS@AOL.COM



11-15-97 © 1997 United Feature Syndicate, Inc.



Richard P. Feynman, physicist:

Computer science is not as old as physics; it lags by a couple of hundred years. However, this does not mean that there is significantly less on the computer scientist's plate than on the physicist's: younger it may be, but it has had a far more intense upbringing!

Broad Goals of CS 60

- Exposure to a variety of important areas of computer science
- Logical thinking and techniques
- Specification and problem solving
- Programming practice in a variety of paradigms

Programming Paradigms

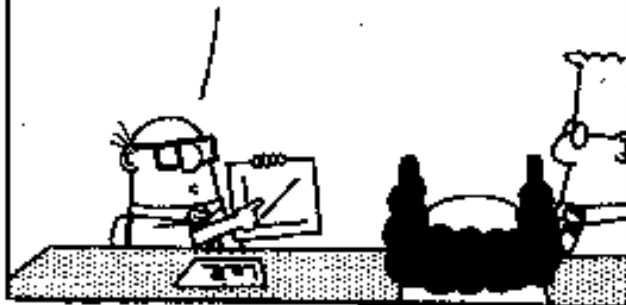
- Ones we will see:
 - Functional programming
 - Object-oriented programming
 - Logic programming
 - Assembly-language programming
- *because* these are important in thinking about software and hardware construction.
- "A language that doesn't affect the way you think about programming is not worth knowing."

Alan Perlis

What is a "paradigm" anyway?

- an example serving as a model
- a pattern
- (used to be) minimum change to make a call from a phone booth :-)

THIS METRIC SHOWS AN
EXCELLENT TREND IN THE
NUMBER OF DAYS SINCE
THE BEGINNING OF MY
PROJECT.



S. Adams E-mail: SCOTTADAMS@AOL.COM

DILBERT®

BY
SCOTT ADAMS



THAT GROWTH RATE
COMPARES FAVORABLY
WITH THE BEST
COMPANIES IN OUR
TIME ZONE.



I'M
WORKING
SMARTER,
NOT
HARDER.

IT'S A
WHOLE NEW
PARADIGM.



12/2/96 © 1996 United Feature Syndicate, Inc.

Why we write programs:

- to make a system or device that carries out some function
- to communicate with others
- to try ideas, to learn
- to *convince ourselves* that we understand (rather than just saying we do)

Old Chinese Proverb

- I hear, I forget.
- I see, I remember.
- I program, I understand.

My Best Advice

- CS 60 assignments are not cook-book; they demand a certain level of intellectual engagement. Therefore:
- Starting thinking about an assignment as soon as it is given.
- Let your sub-conscious mental processes help solve problems.
- Allow yourself space to experiment.

Getting Help

- I *welcome* you to come to my office for discussion of problems.
- The **grutors** will also be available for help.
- We (the grutors and I) *want* to receive your emailed questions: **Mail to: cs60help@cs.hmc.edu**
- *Get help as soon as possible when you feel you are not making progress; don't waste hours not knowing what to do.*
- There is no stigma attached to getting help or asking questions. It is **intended** that you will need to do so.

Submitting Assignments

- You will be given an account on our UNIX server

turing.cs.hmc.edu

- You can *only* submit homework from this account using the program:

`cs60submit your-filename`

CS 60 Home Page

<http://cs.hmc.edu/courses/current/cs60>

- links to the syllabus and other good stuff
- will be updated constantly as we go

Chapter One

- Gives an overview of the rest of the material
- Talks about *abstraction*

It may take some time to appreciate this;
perhaps the whole semester, or more.

“Truth be told, all software engineering is based on abstraction and abstract models.

Abstract thinking is, developmentally speaking, a more advanced and sophisticated mode of thought that takes years for children to acquire. There are some adults who never learn to cut free of concrete literalism.”

Larry Constantine
Object Magazine, Dec. 1996

Chapter Two

- Talks about *information structures*:
 - an abstract view of data structures
 - can be programmed directly in our *rex* language