CSE 142
Computer Programming I

Overview and Welcome

Hal Perkins and Susan Eggers
Spring Quarter 2000
Slides based on those of Martin Dickey and John Zahorjan, Winter 2000, and previous quarters.

Today's Outline

- What Is 142?
- What is programming?
- Should you be here?
- What to expect
  (workload, grades, difficulty, fun, ...)
- Course organization
- First Assignment

Can't get in?

- Some new spaces will open up this week!
- History shows that many students drop 142 during the first two weeks of the course
- All you can do is keep trying
  - No waiting list, no lottery
  - Matriculated undergrads have priority over grads and non-matriculated students
- Instructors do not have entry codes

What to do until then...

- You are welcome to attend this week and do the first assignments, but… we cannot guarantee you will get in.
- Go to some quiz section on Thursday
- CSE (the Computer Science and Engineering Department) has undergraduate advisors in Sieg 114.
  - See them for all registration advice and signatures (but not entry codes)
  - See them for information about becoming a major
  - See them if you get discouraged and want to drop

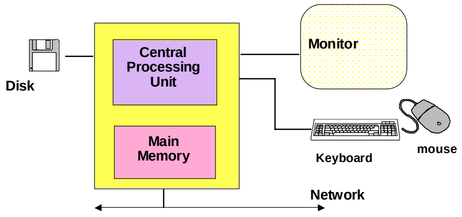
CSE/ENGR 142
Computer Programming I

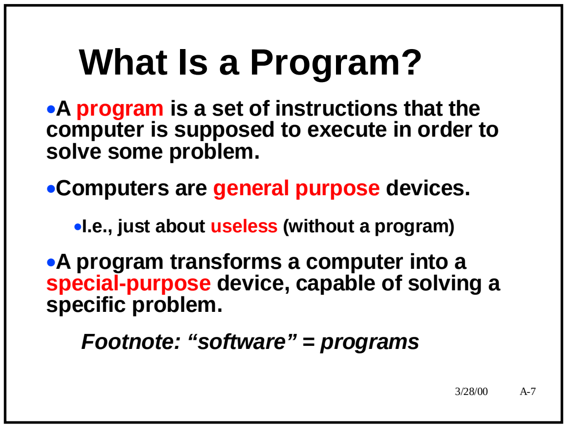
UW Catalog Description:

> *Basic programming-in-the-small abilities and concepts. Highlights include procedural and functional abstraction with simple built-in data type manipulation. Basic abilities of writing, executing and debugging programs.*

Note: It doesn't say C (nor Java, FORTRAN, Pascal, ...)

What's a Computer?

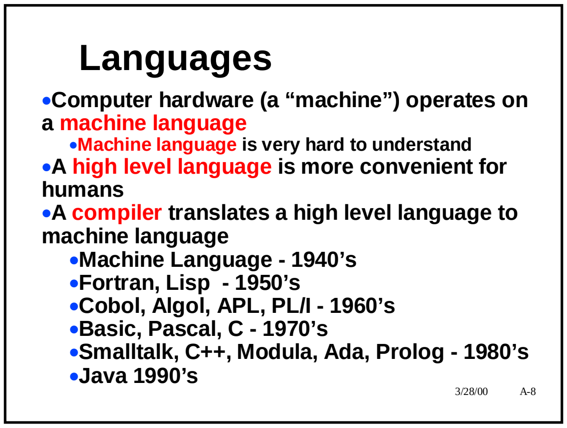




What Is a Program?

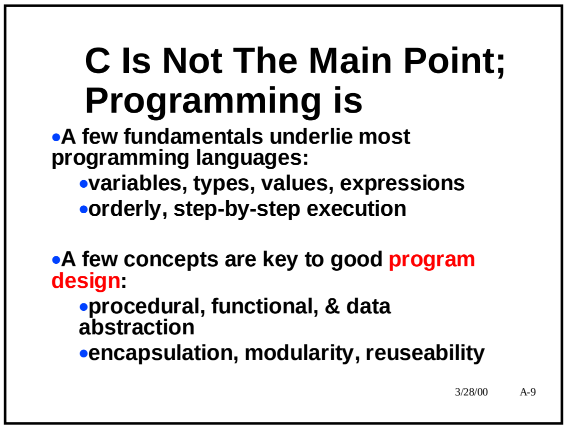
- A program is a set of instructions that the computer is supposed to execute in order to solve some problem.
- Computers are general purpose devices.
  - I.e., just about useless (without a program)
- A program transforms a computer into a special-purpose device, capable of solving a specific problem.

Footnote: "software" = programs



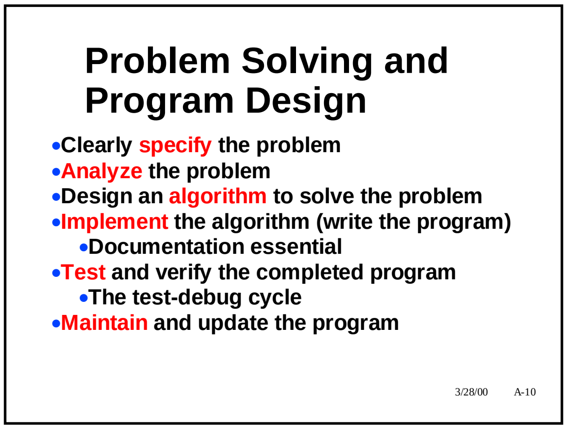
Languages

- Computer hardware (a "machine") operates on a machine language
  - Machine language is very hard to understand
- A high level language is more convenient for humans
- A compiler translates a high level language to machine language
  - Machine Language - 1940's
  - Fortran, Lisp - 1950's
  - Cobol, Algol, APL, PL/I - 1960's
  - Basic, Pascal, C - 1970's
  - Smalltalk, C++, Modula, Ada, Prolog - 1980's
  - Java 1990's



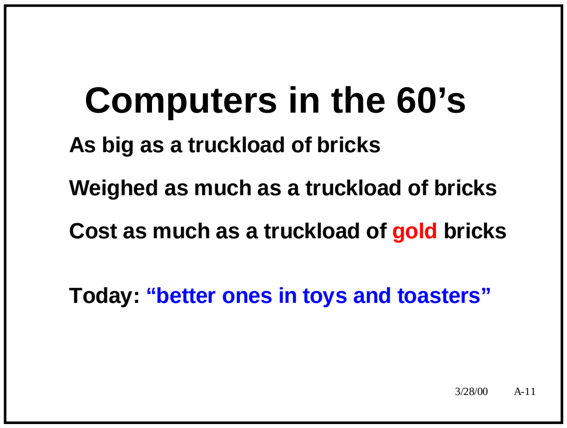
C Is Not The Main Point; Programming is

- A few fundamentals underlie most programming languages:
  - variables, types, values, expressions
  - orderly, step-by-step execution
- A few concepts are key to good program design:
  - procedural, functional, & data abstraction
  - encapsulation, modularity, reuseability



Problem Solving and Program Design

- Clearly specify the problem
- Analyze the problem
- Design an algorithm to solve the problem
- Implement the algorithm (write the program)
  - Documentation essential
- Test and verify the completed program
  - The test-debug cycle
- Maintain and update the program



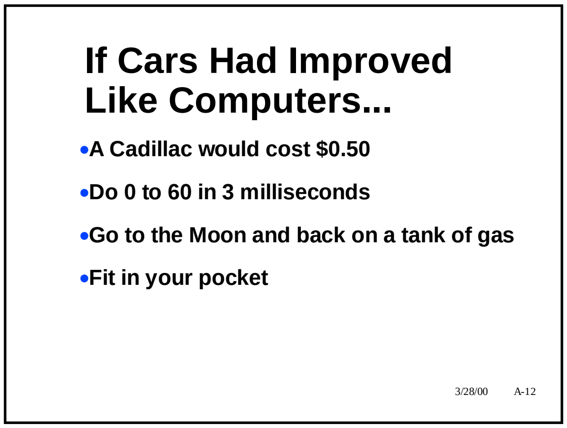
Computers in the 60's

As big as a truckload of bricks

Weighed as much as a truckload of bricks

Cost as much as a truckload of gold bricks

Today: "better ones in toys and toasters"



If Cars Had Improved Like Computers...

- A Cadillac would cost $0.50
- Do 0 to 60 in 3 milliseconds
- Go to the Moon and back on a tank of gas
- Fit in your pocket

Is The Revolution Over?

- Intel Pentium II has 7.5 million transistors
- 30-300 million transistors per chip easily foreseeable
- 10x faster clock speeds, 100x faster throughput conceivable
  - "Moore's Law"
- Advances also in memory, magnetic (disk) and optical (CD) storage, networking, etc.
- Yet prices aren't rising!

What about software?

- Major software-based products literally unimaginable 10 - 15 years ago
  - desktop publishing, Internet browsers, 3-D games, Web audio and video, e-commerce
- Big improvements in
  - handwriting and speech recognition
  - computer animation, graphics, vision
  - digital consumer products
    - cell phones, CD-ROM and DVD, etc.

Why Are We Here Today?

- Computers are changing the way science and engineering is done
- Computers will continue to change all our lives
- Programming is a key enabling technology
- *That's the Big Picture. What about you?*

Why Are <u>You</u> Here?

- "I know computing is important, and I need basic expertise."
- "I'm just curious."
- "I have this computer and I want to do X but I can't get software that does X."
- "It's a requirement for my major."
- "I want a career in computing."

Should you be here??

- If you already know C and the contents of this course...
  - You <u>can</u> go directly to CSE 143 (142 credit available if you do well in 143)
    - Go there *today* to check it out: Guggenheim 224, 2:30 pm MWF
  - This course may be boring but will still be time-consuming. You'll have to do things "our way."
  - If you stay, please participate!

Should you be here??

- If you are a complete novice to programming…
  - Prior programming experience is NOT required!
- But...programming a computer is very different from simply using one.
- Being comfortable or even expert with computer applications does not prepare you for programming!

So What is Programming Like?

- It’s really hard to describe!
- Many similarities to solving “word problems” in math
  - Translate a problem description into a formal solution
  - Symbol manipulation an integral part
- Some people describe it as “puzzle solving”
- A mix of high-level creativity and low-level picky details

What To Expect

- Grades:
  - Class average just below 3.0
  - Always some 4.0’s, always some 0.0's
- Is this a tough course?
  - Contents are challenging
  - Projects can be time-consuming
  - Cramming won’t work -- must keep up
- Fun?
  - Absolutely!

Advice for New Programmers

- ➔Keep up with the course day-by-day
- ➔Seek help early and often:
  - ➔TA, instructor office hours
  - ➔consultants in IPL
  - ➔ undergrad advisors in Sieg 114
    - ➔Some special tutoring is available
- ➔Consider joining a "low-background" section

The UW Drop Policy

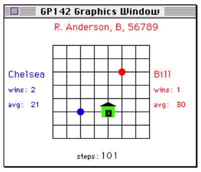
- Historically, 10%-15% of CSE/ENGR 142 enrollees dropped the course
  - – Most drops were after the 10th day under the old drop policy
- It’s very hard to judge how challenging this course is by its first two weeks
- Unfortunately, you must drop by 10th day !
  - – Once per year you get a “free” drop.
  - – Also possible to change status to noncredit until week 7 of the quarter.

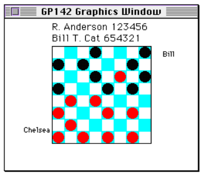
Course Organization

- Lectures 3 times a week
- Quiz section once a week
- Programming projects
  - In the lab or at home (with proper equipment)
  - Individual effort (not group projects)
  - Normally, Sunday night electronic turnin deadline, with paper copy due Monday
- Two midterm exams
- Final exam – Wednesday, June 7, 2000
  - May be a time change from original schedule!
- Other activities: non-programming HW, quizzes

Homework Can Be Fun

(Some examples from Autumn 1994)





Quiz Section

- Quiz section: once a week
  - Review, questions, exercises, quizzes and more
- Designated sections
  - "low-background": for students without previous programming experience
  - "high-background": for students with considerable experience
  - All sections have identical assignments, tests, and grading criteria
- Can request section swap in Wed. lecture
- Please memorize your student ID#, quiz section ID and your TA's name!

Textbook and materials

- Text: "*Problem Solving and Program Design in C*" - Hanly and Koffman
  - 3rd edition (2nd edition ok with minor adjustments)
  - "self-check" and "quick-check" exercises highly recommended (answers in book)
- Course Packets
  - Slides (based on last quarter's), reference material
    - Many students bring this to every lecture to take notes (recommended)
  - Buy at: Professional Copy & Print, 4200 U. Way

Final Exam (Comprehensive)

- Wednesday, June 7, 2000
- Times and rooms, but not the day, are likely to be different from the on-line Time Schedule (will be announced when we know the details)
- With permission you can move to the exam period other than the one you are scheduled for.
- If you have a problem with both times contact course administrator as soon as times are announced.
- *It will not be possible to take the final on any other day.*

142 Web Site

http://www.cs.washington.edu/education/courses/142

- Messages from class mailing list (read often)
- Homework projects
  - Instructions
  - Downloading
  - Turn-in
- Lecture schedule and current reading
- Lecture slides
- Tips, hints
- Office hours
- Exam information, lab schedules, etc. etc.

IPL:
Intro Programming Lab

- Sieg 323
- Pentium PC's running Windows NT
  - Microsoft Visual C++ Version 6.0
  - Web browsers
  - Electronic mail
- 142 consultants (posted hours)
- Visit today!

Course Staff

Here to help you succeed!

Instructors
  - You can go to either instructor's office hours
- TA's
  - Teach sections & grade homework
  - You can go to any TA's office hours
- Lab staff in IPL
  - Operator (front-desk)
  - 142 Consultants
- Teleconsultants: Get help at home!
- CSE undergrad advisors: Sieg 114
- Instructional technologist
- Course administrator: Special arrangements, fix bookkeeping problems, claim abandoned work, etc.

Mailing Lists & Newsgroups

- Announcements, tips, hints, place to ask questions and get answers
- uwash.class.cse142.* newsgroups for general discussions
- "cse142-announce" mailing list for announcements from course staff
  - Must subscribe first
    - send mail to: majordomo@cs.washington.edu
    - message text exactly as follows:

      subscribe cse142-announce
  - Details on the Web

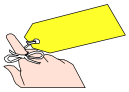
Homework # 0

- Due in 2 parts: This Friday(!) and Sunday/Monday
- Read Chapter 1 and handouts.
- Go to IPL (Sieg 323) and start learning the system. Be sure and read section 1.2 before going to lab.
- Start playing with the other software tools.
- There's lots to read during the quarter: Start going & keep going!

If you compute at home...

- Stay connected with Web and e-mail
- Get a compiler - MSVC++ 6.0 recommended
  - UW Bookstore has the "Standard" edition for <$50.
- Windows 95/98/NT/MSVC is our official platform
  - some support for others
- Do first project in IPL
  - just to become familiar with it
- Help on computing at home is on 142 web site
- Expect a few headaches (but worth it!)

Tutorials



- Optional tutorials, this week
- Hands-on sessions in the IPL to get you familiar with the system
  - Windows 95/98/NT, Web browser, basic MSVC, ...
  - Meant for people unfamiliar with the software
    - No advanced stuff
  - Can do assign. 0 (esp. part B) during tutorial
  - Seating: 1st come, 1st served
  - Length: about 1 hour

Location: IPL, Sieg 323
Time: TBA (check the Web)