## Announcements!

- Make sure to get a copy of the handout. It should have three parts:
  - ➤Course Administration (general information)
  - ➤Tentative Schedule
  - ➤Homework 0 Starter Sheet
- Quiz Section AG (Thursday 1:10–2:10) has moved to EE1 025



# CSE 142
# Computer Programming I

## Overview and Welcome

Slides adapted from those of Hal Perkins and Susan Eggers, Spring 2000, and previous quarters.



## Your Instructor

- Isaac Kunen
- email: zook@cs.washington.edu
- Tentative office hours:
  - ➤Tuesday, 4:00
  - ➤Wednesday, 2:00
  - ➤Or by appointment
- Office: 226c Sieg Hall



## The Complete Staff

- Instructor:
  - ➤Isaac Kunen
- Course Administrator
  - ➤Melissa Albin
- Course Technologist/ Webmaster/Wizard
  - ➤Dan Boren
- Head TA
  - ➤Justin Campbell
- Teaching Assistants
  - ➤Justin Goshi
  - ➤Karen Liu
  - ➤David Chang
  - ➤…
- Consultants
  - ➤TBA



## Today's Outline

- What is 142?
- What is programming?
- Should you be here?
- What to expect
- Course organization
- First assignment



## If you can't get in...

- Many students will drop this first week
- Spaces will open up, however...
- We cannot guarantee you a space
- Just keep trying
  - ➤There is no waiting list or lottery
  - ➤Matriculated undergrads have priority over non-matriculated students and grads
- Instructors do not have entry codes

## What to do if you're not in

- Attend class this week
- Go to some quiz section on Thursday
- Do the first homework (HW0)



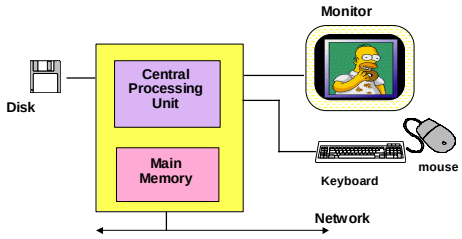
## What is CSE 142?

- 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.
- It doesn't say C, but...



## What is a Computer?






## What is a program?

- A computer is a general purpose machine, but is useless without a program
- A program is a set of instructions that tells the computer what to do
- The program turns the general purpose machine into a special purpose machine
- Any piece of software is a program



## Languages

- A computer (a "machine") operates on machine language
  - Machine language is very hard for people to understand
- A high-level language is more convenient for humans
  - *Lots* of languages: Fortran, Lisp, Algol, Pascal, C, Smalltalk, C++, Modula, Java,…



## C Is Not the Main Point…

- The fundamental concepts are important!
  - variables
  - types
  - expressions
  - flow of control
  - abstraction
  - modularity
  - encapsulation… and the list goes on!

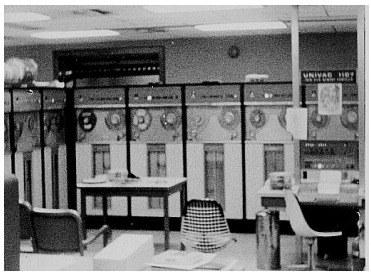
## C Is Not the Main Point...

- Fundamental Skills
  - ➤Formalizing problems
  - ➤Formalizing solutions
  - ➤Debugging
  - ➤Writing “clean” code
  - ➤etc.



## Computers in the 60’s

- Big
- Slow
- Expensive





## Computers Today

- Small
- Fast
- Cheap
- Aerodynamic



## If Cars Had Improved Like Computers...

- A typical car would cost $5.00
- It would get 40,000km per gallon
- It would crash a lot more often

(Estimates by Woodall, 1997)



## Is the Revolution Over?

- AMD Athlon has 22 million transistors
- 30-300 million transistors per chip is certainly possible
  - ➤Moore’s law
- Disks are getting larger, networks are getting faster
- Prices are going down!



## What About Software?

- Software we use today would not be possible 10–15 years ago.
  - ➤Internet browsers, 3-D games, e-commerce
- Huge improvements in
  - ➤Handwriting and speech recognition
  - ➤Computer graphics
  - ➤Digital consumer goods
    - – Cell phones, DVD, MP3, Internet Telephony, etc.

## Why are we here?

- Computers are changing the way we live and do work
- Computers are now a part of most aspects of our lives
- Programming lets *you* take control of the technology that surrounds us



## Why Are *You* Here?

- "I know computing is important, and I want a good introduction."
- "I'm just curious."
- "I have this computer, and I want to do *X,* but I can't find software to do *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 the course…
  - ➤You can go directly to CSE 143 and get credit for 142 if you do well.
    - Go there ASAP to check it out: Sieg 134, 8:00am, MWF
  - ➤You may find this course boring and time consuming if you choose to stay
  - ➤If you stay, please participate!



## Should You Be Here?

- If you are a complete programming novice
  - ➤Prior experience is *NOT* a prerequisite!
  - ➤Programming is very different from just using one
  - ➤Being an expert with most computer applications does not prepare you for programming



## What is Programming Like?

- Many similarities to solving word problems in math
  - ➤Translate a problem description into a formal language
  - ➤Develop a strategy for solving it
- Algorithmic thinking
- A mix of high-level creativity, and low-level picky details



## What to Expect

- Grades
  - ➤Class average just below a 3.0
  - ➤Always some 4.0's, always some 0.0's
- This is a tough course
  - ➤Contents are challenging
  - ➤Projects are time-consuming
  - ➤Cramming will not work—Practice will
- Fun? Absolutely!

## Advice for New Programmers

- Keep up with the course day-by-day
- Get help early and often
  - ➤TA and instructor office hours (we get lonely)
  - ➤Consultants
  - ➤Undergraduate advisors in Sieg 114
    - – Some tutoring is available
- If you are worried, 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
- The course *will* get harder as it goes on
- Unfortunately, you must drop by day 10!
  - ➤Once a year you get a "free" drop
  - ➤Can change to noncredit through week 7



## Course Organization

- Lecture 3 times per week
- Quiz sections once per week
- Programming projects and homework
- Quizzes
- Two midterms
- Final exam



## Quiz Section

- Once a week
  - ➤Review, ask question, take quizzes, etc.
- Special sections
  - ➤"low-background" section for students who have limited computing experience
  - ➤"high-background" section for students who have prior experience
  - ➤All sections have identical requirements



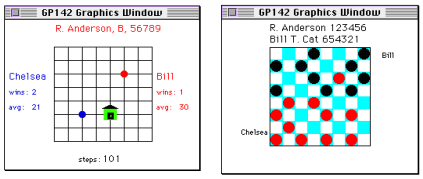
## Quiz section (contd.)

- Section swap requests this Wednesday in lecture
- Please memorize your student ID, quiz section ID, and your TA's name!



## Homework! :(

- Written work as well as programming
- Can be fun: (from Autumn 1994)



## Weekly Quizzes

- Short, 5-minute quizzes
- In quiz section (surprise!)
- Low stress



## Midterms and Final

- Two in-class midterms
  - ➤7th of July
  - ➤28th of July
- One in-class final (comprehensive)
  - ➤18th of August (last day of class)
  - ➤*Not possible to take the tests on any other days. Mark your calendar now!*



## Textbook and Materials

- Text: *Problem Solving & Program Design in C,* Hanly and Koffman
  - ➤3rd edition (2nd okay with minor adjustments)
  - ➤Exercises are very valuable
- Course Packets
  - ➤Last quarter's slides, reference material
  - ➤Buy at: Professional Copy & Print, 4200 University Way ($11.05 + Tax)



## 142 Web Site

**`http://www.cs.washington.edu/142`**

- Mailing list archive
- Homework assignments
- Lecture and reading schedule
- Lecture slides
- Office hours
- Lab schedules, *and much more!*



## Mailing Lists & Newsgroups

- Announcements, tips, hints, place to ask questions and get answers
- Newsgroup: **`uwash.class.cse142`**
- Subscribe to class mailing lists ASAP
  - ➤Mail: **`majordomo@cs.washington.edu`**
  - ➤Body of the message:

    **`subscribe cse142-announce`**

    **`subscribe cse142-section-XX`**

    (Where **`XX`** is your section)



## Programming Lab

- Mary Gates Hall CRC (Suite 131)
- Pentium PC's running Windows 98
  - ➤Microsoft Visual C++ Version 6.0
  - ➤Web browsers
  - ➤Email
  - ➤etc.
- 142 consultants (posted hours)
- Visit today!

## Instructional Staff

- *Here to help you*
- TA's
  - ➤Teach sections & grade homework
  - ➤See any TA
- Consultants at MGH
- Teleconsultants



## Other Staff

- CSE undergraduate advisors
- Instructional technologist
- Course administrator:
  - ➤Special arrangements
  - ➤Fix bookkeeping problems
  - ➤Claim abandoned work, etc.



## Homework #0

- Due in 2 parts: This Friday, and next Monday
- Read Chapter 1 and handouts
- Go to the Lab and start learning the system
- Start playing with the software tools
- Lots to read this quarter: Keep up!



## If You Compute At Home...

- Stay connected with web and email
- Get a compiler
  - ➤MSVC++ 6.0 recommended (≈$50 at UBS)
- Windows 95/98/NT/MSVC is our official platform
  - ➤*Some* support for others
- Get familiar with the CRC
- Help for working at home on web site



## Tutorials

- Optional tutorials this week
- Hands on sessions in Odegaard to get you familiar with the system
  - ➤Windows, Web browser, basic MSVCC
  - ➤Can do HW 0b during the tutorial
  - ➤Seating: 1st come, 1st served
  - ➤Length: about 1 hour
- Location and times TBA
  - ➤First one Tuesday 2:00–3:00, Collab. 1