

Welcome!

Principles of Computer Science

CPS 1

LSRC B101

M, W, F 1:10-2:00

Professor Dietolf (Dee) Ramm

<http://www.cs.duke.edu/courses/fall03/csp001>

CPS 001

1.1

Today's topics

- What is this course about?
- How are we going to learn that?
- Who is this guy talking to us?
- Where do we from here?
- An overview of computer science

- Upcoming
 - The World Wide Web and HTML
 - Problem Solving

CPS 001

1.2

Course Information

“A survey of the great ideas of computer science along with experience with programming, the theoretical foundations of computer science, how computer systems are organized and work, and the applications of computers including their effect on society.”

• Grading Breakdown

Assessment	Weight (approx)
Labs	10%
Lab Final	5%
In-class	5%
Quiz/Assign	23%
Project	17%
Midterm	15%
Final	25%

- Scores on absolute scale
- **No make-ups, no late submissions**
- Important Dates
 - Midterm 10/1
 - Final 12/13 2pm-5 !!!!
 - Wednesday Quizzes
- Let me know ASAP if you have any concerns

CPS 001

1.3

Administrivia

- Read Blackboard web page **regularly**
<https://courses.duke.edu>
- Read discussion forums

- Reading posted on Blackboard page due by Friday class time
- Labs start on Monday, September 1 (Labor Day)

- Collaboration policy
- Late policy

CPS 001

1.4

Frequently Asked Questions

- What is the prerequisite?
 - ↗ High school algebra (?)
- How does this course fit into the curriculum?
 - ↗ A survey, service course designed for non-majors
 - ↗ CPS 6 is more programming oriented
 - ↗ Satisfies QID (M) and STS requirements
- Why take this course?
 - ↗ Computers are interesting, useful, and ubiquitous
 - ↗ Pure entertainment
- Why doesn't this course teach anything *practical*?

CPS 001

1.5

On the subject of questions...

- Did you ask any good questions today?
 - ↗ *Ideas and Information* by Nobel prize winning physicist Arno Penzias
 - ↗ Questions which illuminate help nourish ideas
 - ↗ Children are born curious
 - ↗ Fear of public displays of ignorance prevents learning
- Participate in class
- Go to office hours
- Make study groups with your classmates

CPS 001

1.6

Getting help

- Contact Information
 - ↗ Email: ramm@cs.duke.edu
 - ↗ Phone: 660-6532
 - ↗ Office Hours in D226 Levine Science Research Center
 - Wed 10:00-11:00, Thu 2:30-3:30
 - When my door is open
 - By appointment
- TAs
 - ↗ Grad TA: Vijeta Johri (vi_jeta@cs.duke.edu)
 - Office hours TBA
 - D328 LSRC, 660-6588
 - ↗ Head UTA: tba
- And a whole gang of UTAs

CPS 001

1.7

Who are you?

- Class
 - ↗ Frosh / Soph / Jun / Sen / Other
- Major
 - ↗ Fuzzy / Techie / Undecided
- Box at home?
 - ↗ Mac / PC / Linux / Other
- Programmed before
 - ↗ Never / HTML / Javascript / Java / C / C++ / Lisp / Other

CPS 001

1.8

Stories

- Who is Marc Andreessen and what did he do (21 years old)?
- Who is Claude Shannon and what did he do (21 years old)?
- Who is Linus Torvalds and what did he do (21 years old)?
- Who is Steve Jobs and what did he do (21 years old)?
- Who is Ada Byron and did she do (27 years old)?
- Who is Jim Ellis and what did he do (23 years old)?
- Who is Alan Biermann and what did he do (51 years old)?

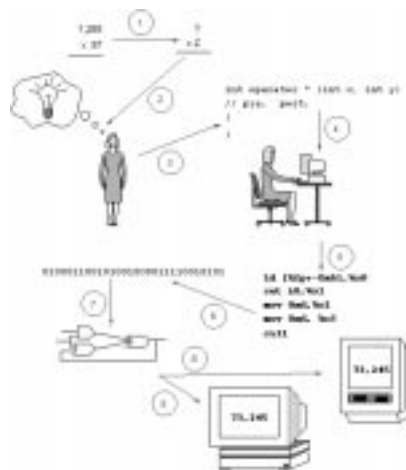
Other names: Alan Turing, Edsger Dijkstra,... (more?)

Precise, logical thinking

- Breaking down a task into *unambiguous* steps
- Computers are *deterministic*
- Algorithm: *a set of steps that defines how a task is performed*
- Debugging
 - Programs will rarely work the first time one writes them
 - Systematic approach to detecting, diagnosing, and fixing errors
 - Debugging skills are useful in many parts of life

Creating a Program

- Specify the problem
 - remove ambiguities
 - identify constraints
- Develop algorithms, design classes, design software architecture
- Implement program
 - revisit design
 - test, code, debug
 - revisit design
- Documentation, testing, maintenance of program
- A *programming language* is a way to describe an algorithm.



Quotations about Computer Science

"Computer science has such intimate relations with so many other subjects that it is hard to see it as a thing unto itself"

- Marvin Minsky, 1979

"It has often been said that a person does not really understand something until he teaches it to someone else"

- Donald Knuth

"Actually, a person does not really understand something until he can teach it to a computer"

- Judith Gal-Ezer and David Harel

Algorithms as Cornerstone of CS

- Step-by-step process that solves a problem
 - more precise than a recipe
 - eventually stops with an answer
 - general process rather than specific to a computer or to a programming language
- Searching: for phone number of G. Samsa, whose number is 929-9338, or for the person whose number is 489-6569
 - Are these searches different?
- If the phone book has 8 million numbers in it (why are there only 7.9 million phone numbers with area code 212?)
 - How many queries to find phone number of G. Samsa?
 - How many queries to find person with number 929-9338
 - What about IP addresses?

CPS 001

1.14

Sorting Experiment: why do we sort?

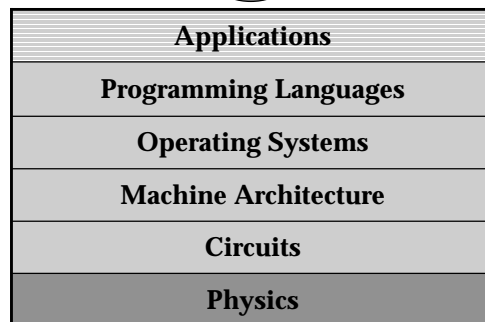
- Groups of four people are given a bag containing strips of paper
 - on each piece of paper is an 8-15 letter English word
 - create a sorted list of all the words in the bag
 - there are 100 words in a bag
- What issues arise in developing an algorithm for this sort?
 -
 -
- Can you write a description of an algorithm for others to follow?
 - Do you need a 1-800 support line for your algorithm?
 - Are you confident your algorithm works?

CPS 001

1.15

Layers of *abstraction*

The User:



The Result:



CPS 001

1.16

Survey the field

- Artificial intelligence
- Graphics/Multimedia
- Parallel Computation
- Programming Languages
- Systems
- Scientific Computing
- Theory
- User Interfaces

CPS 001

1.17

Themes and Concepts of CS

- Theory
 - ↗ properties of algorithms, how fast, how much memory
 - ↗ average case, worst case: sorting cards, words, exams
 - ↗ *provable* properties, in a mathematical sense
- Language
 - ↗ programming languages: C++, Java, C, Perl, Fortran, Lisp, Scheme, Visual BASIC, ML, ...
 - ↗ Assembly language, machine language,
 - ↗ Natural language such as English
- Architecture
 - ↗ Main memory, cache memory, disk, USB, SCSI, ...
 - ↗ pipeline, multi-processor

What's more difficult

1. Sketch artist vs. Your dog?
 - ↗ Generating a face
 - ↗ Recognizing a face
 2. Vacation planner vs. Super-librarian?
 - ↗ Finding the best route through cities
 - ↗ Alphabetize the books in the Library of Congress
- Fundamental ideas of computer science
 - ↗ Complexity
 - ↗ Computability

Complexity: What's hard, what's easy?

- What is a prime number?
 - ↗ 2, 3, 5, 7, 11, 13, ...
 - ↗ Largest prime?
- 48112959837082048697
- 671998030559713968361666935769
- How do we determine if these numbers are prime?
 - ↗ Test 3, 5, 7, ...
 - ↗ If we can test one million numbers a second, how long to check a 100 digit #?
- Why do we care?
- 671998030559713968361666935767 is not prime, I can prove it but I can't give you the factors.
- Finding factors is "hard", determining primality is "easy"
 - ↗ What does this mean?
 - ↗ Why do we care?
- Encryption depends on this relationship, without encryption and secure web transactions where would we be?

Questions you will be able to answer

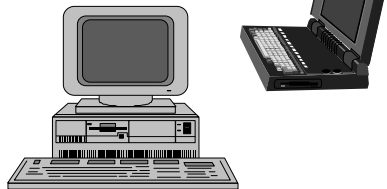
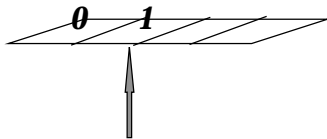
- Vendor tries to sell you a system that will check all of your systems and procedures to see if they are correct.
 - ↗ A good deal?
- Programmer tells you that to optimize the routing of your sales personnel is beyond the power of today's computers.
 - ↗ Do you believe her?
- Computer consultant demonstrates complicated management system with test data including a handful of employees.
 - ↗ Is the performance with this small set of data a good indicator of how the system will perform with all of your company data entered?

What is a computer?

- Turing machine: invented by Alan Turing in 1936 as a theoretical model

Mainframe, PC, laptop, supercomputer

infinite tape, moving tape-reader

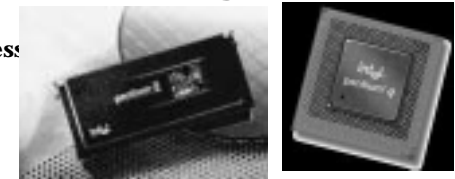


A computer is a computer, is a computer, Church-Turing Thesis, all have same “power”

Chips, Central Processing Unit (CPU)

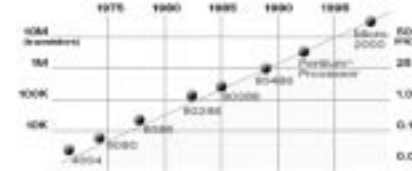
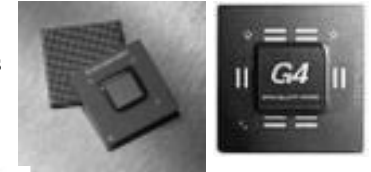
- CPU chips/Microprocess

- Pentium (top)
- G3/4 (bottom)
- Sound, video, ...



- Moore's Law

- chip “size” (# transistors) doubles every 12--18 months (formulated in 1965)
- 2,300 transistors Intel 4004, 42 million Pentium 4



Assignment 0

- Due in your first lab
- What would life be like without computers?
- Pick one 24 hour period over the next week
 1. Write down ALL interactions you have with a computer
 2. What would change in your life if all computers stopped working?
- Come up with *clean* computer science related joke
- First question you should be asking is: *What is a computer?*
 -