

Welcome!

Computer Science Fundamentals

CPS 1

LSRC B101

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

Professor: Jeffrey Forbes

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
 - Intro to Java

CPS 001

1.2

Administrivia

- Read webpage *regularly*
<http://www.cs.duke.edu/education/courses/fall01/cps001>
- Read newsgroup
<news:duke.cs.cps001>

- Overview handout

CPS 001

1.3

Frequently Asked Questions

- What is the prerequisite?
 - High school algebra (?)
- How does this course fit into the curricula?
 - A survey, service course designed for non-majors
 - CPS 4 and CPS 6 are more programming oriented
- Why take this course?
 - Computers are interesting, useful, and ubiquitous
 - Pure entertainment

CPS 001

1.4

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**

Getting help

- **Contact Information**
 - ↗ **Email:** forbes@cs.duke.edu
 - ↗ **Phone:** 660-6550
 - ↗ **Office Hours in D235 Levine Science Research Center**
 - M,W 2:00-3:00
 - When my door is open
 - By appointment
- **Graduate TA**
 - ↗ **Donald Onyango (dco@cs)**
 - Office hours W 9-11am
 - D103 LSRC, 660-6501
- **And a whole gang of UTAs**

Who are you?

- Show of hands, height indicates magnitude
- **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

Stories

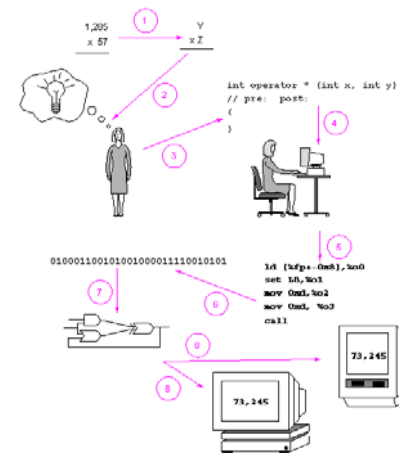
- **Who is Shawn Fanning and what did he do (19 years old)?**
- **Who is Marc Andreessen and what did he do (21 years old)?**
- **Who is Claude Shannon and what did he do (21 years old)?**
- **Who is Linus Torvalds and what did he do (21 years old)?**
- **Who is Dmitry Sklyarov and what did he do (26 years old)?**
- **Who is Tim Berners-Lee and what did he do (35 years old)?**
- **Who is Alan Biermann and what did he do (51 years old)?**

Precise, logical thinking

- Breaking down a task into *unambiguous* steps
- Computers are *deterministic*
- 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.



The Ice Cream Story

"This is the second time I have written you, and I don't blame you for not answering me, because I kind of sounded crazy, but it is a fact that we have a tradition in our family of ice cream for dessert after dinner each night. But the kind of ice cream varies so, every night, after we've eaten, the whole family votes on which kind of ice cream we should have and I drive down to the store to get it. It's also a fact that I recently purchased a new car and since then my trips to the store have created a problem. You see, every time I buy vanilla ice cream, when I start back from the store my car won't start. If I get any other kind of ice cream, the car starts just fine. I want you to know I'm serious about this question, no matter how silly it sounds: 'What is there about your car that makes it not start when I get vanilla ice cream, and easy to start whenever get any other kind?'"

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

1.13

Sorting Experiment: why do we sort?

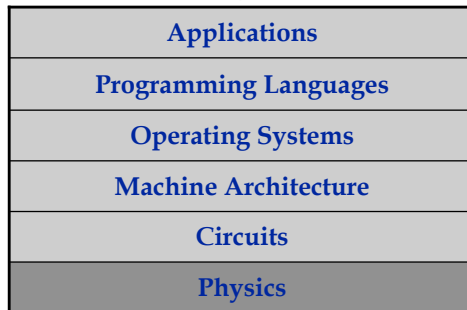
- Groups of four people are given a bag containing strips of paper
 - on each piece of paper is an 8-15 letter English word
 - create a sorted list of all the words in the bag
 - there are 100 words in a bag
- What issues arise in developing an algorithm for this sort?
 -
 -
- Can you write a description 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

1.14

Layers of abstraction

The User:



The Result:



CPS 001

1.15

Survey the field

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

CPS 001

1.16

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

CPS 001

1.17

Abstraction, Complexity, Models

- **What is an integer?**
 - ↗ In mathematics we can define integers easily, infinite set of numbers and operations on the numbers (e.g., +, -, *, /) {...-3, -2, -1, 0, 1, 2, 3, ...}
 - ↗ In programming, finite memory of computer imposes a limit on the magnitude of integers.
 - Possible to program with effectively infinite integers (as large as computation and memory permit) at the expense of efficiency
 - At some point addition is implemented with hardware, but that's not a concern to those writing software (or is it?)
 - C++ doesn't require specific size for integers, Java does
- **Floating-point numbers have an IEEE standard, required because it's more expensive to do arithmetic with 3.14159 than with 2**

CPS 001

1.18

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

CPS 001

1.19

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?**

CPS 001

1.20

Questions you will be able to answer

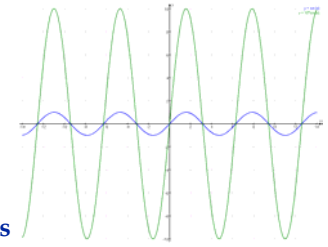
- Vendor tries to sell you a system that will check all of your systems and procedures to see if they are correct.
 - A good deal?
- Programmer tells you that to optimize the routing of your sales personnel is beyond the power of today's computers.
 - Do you believe her?
- Computer consultant demonstrates 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?

CPS 001

1.21

What is digital?

- What's the difference between
 - Rolex and Timex?
 - VCR tape and DVD?
- Sampling analog music for CD's
 - $44,100 \text{ samples/channel/second} * 2 \text{ channels} * 2 \text{ bytes/sample} * 74 \text{ minutes} * 60 \text{ seconds/minute} = 783 \text{ million bytes}$
- How does MP3 help?



CPS 001

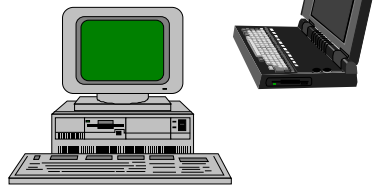
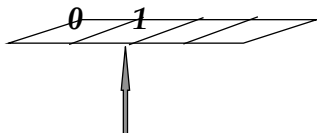
1.22

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"

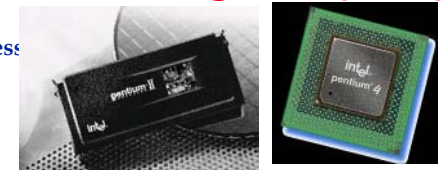
CPS 001

1.23

Chips, Central Processing Unit (CPU)

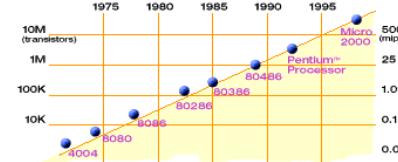
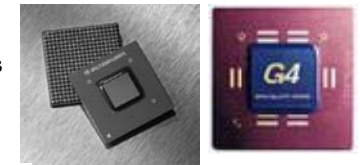
- CPU chips/Microprocessors

- 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



CPS 001

1.24

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?
 3. Name an occupation or business that has not been affected by the World Wide Web

- First question you should be asking is: *What is a computer?*
 - Define computer as a microprocessor/CPU.

- Extra credit for anyone who can go a day without using any computers.