Welcome to CS 126!

COS 126 Lecture 1: Introduction

Introductory survey course
- no prerequisites
- basic principles of computer science
- learn to use computers effectively
- check FAQs on web

Topics introduced:
- hardware and software systems
- programming in C and other languages
- algorithms and data structures
- theory of computation
- applications to solving scientific problems

#include <stdio.h>
main()
{
    printf("This is a C program\n");
}

Q. How did the computer scientist die in the shower?
A. The instructions on the shampoo said "Lather, Rinse, Repeat"
**Outline**

- **Administrivia**
  - What is “computer science”?
    - What it’s not
    - Why we learn it
    - Syllabus (long answer)
  - An example
    - A simple machine
    - “Science” behind it
  - Conclusion
    - CS is about abstractions (short answer)

**The Usual Suspects**

- Randy Wang (rywang@cs)
- Lisa Worthington (lworthin@cs)
- Spyridon Triantafyllis (strianta@cs)
- Jie Chen (jennifer@cs)
- Petru Chebeleu (chebeleu@cs)
- Ben Gum (gum@cs)
- Alexey Lvov (lvov@math)
To Get Started

• Visit course web page:
  - [http://www.cs.princeton.edu/courses/cs126](http://www.cs.princeton.edu/courses/cs126)

• Get course packet from Pequod (ready by 9/22?):
  - for more general information

• Go to lab tomorrow (9/17, 10-11:50, 1:30-3:30, CS101)
  - to get on-line

• Decide which precept to go to
  - visit course page for preceptor assignment
  - contact tmhill@cs to make time changes

• Go to precept on Monday (9/20)
  - to get remaining questions answered
Tips

• “CS126 survival guide”
• More...
  - Come to lectures and precepts
  - Do readings, exercises, as well as program assignments
  - Find a “system” that works best for you
  - Read, understand, and borrow from example code before writing your own

Outline

• Administrivia
• What is “computer science”?
• An example
• Conclusion
What Is CS?

• (Why don’t we call chemistry “test tube science”?)
• What CS is not
  - CS is not programming, just as
  - Biology is not about learning to use a microscope
  - Programming is merely a tool
• Why we learn it
  - Appreciate underlying principles and limitations
  - “Meta-learning”: learning how to learn
• What is it?
  - Syllabus (long answer)
Outline

• Administrivia

• What is “computer science”?

• An example
  • How to make a simple machine
  • What we can do with it
  • “Science” behind it

• Conclusion
A Simple Machine

• Want
  - a machine that outputs a random sequence of 0s and 1s

• Some basic terms
  - a bit: a student who’s either male or female
  - a storage element (cell): a seat that can hold one student
  - a register: a whole row of seats
  - a shift register: when clock strikes, stand up and take the seat to your right
  - a “linear feedback shift register”: ...
What Is It Good For?
Message Encryption

- Use LFBSR as a component in an encryption/decryption machine
- Cool detail: “+” and “-” can be xor; so same machines!
Now the “Science” Behind It

• Are the bits really “random”?
• How long would it take before the bit pattern repeat itself?
• Will the machine work equally well if I xor the 10th and 4th bits?
• How many cells do I need for my LFBSR if I want to guarantee a certain degree of security?
Outline

- Administrivia
- What is “computer science”?
- An example
- Conclusion
  - CS is about abstractions (short answer)
Abstractions Involving LFBSR

- Bigger boxes made of smaller ones, hide details behind interfaces
- "Science" at each step for design decisions

Diagram:

```
Cell \xor \rightarrow LFBSR
\rightarrow Random Bits
\rightarrow Encrypted Message
\rightarrow Encryption
\rightarrow Clear Text
Encrypted Message \rightarrow eShopping \rightarrow Customer Orders
```

Computer Systems and Abstract Machines

- Layers of abstraction
  - precisely define a simple machine
  - use it to build a more complex one
  - develop complex systems by building increasingly more complicated machines
  - improve systems by substituting new (better) implementations of abstract machines at any level

LFBSR layers of abstraction
- simple piece of hardware
- converts fill to "random" bits
- can use "random" bits for encryption
- can use encryption for internet commerce

Computer layers of abstraction
- complex piece of hardware
  - CPU, keyboard, printer, storage device
  - machine language programming
  - software systems
    - editor (emacs): create, modify files
    - compiler (cc): transform program to machine instructions
    - operating system (Unix): invoke programs
    - windowing system (X): illusion of multiple computer systems