Slide 1

## Administrivia

- Reminder: Homework 1 due today.

Slide 2

## More “About This Course”

- Course has had several names — “Principles of Algorithm Design”, “Principles of Programming”, now “Principles of Computer Science”.
- Different names reflect an aspect that makes it difficult, namely that ideally it teaches both conceptual skills (“algorithmic thinking”) and practical ones (“programming”, plus use of command-line tools).
- At this point mostly it’s been about the latter. Switch gears here and look at conceptual level, using review of binary numbers as a source of examples.

Slide 3

## Binary Numbers

- We humans usually use the decimal (base 10) number system, but other (positive integer) bases work too. (Well, maybe not base 1.) Binary (base 2) is more widely used in computers because it makes the hardware simpler.
- In base 10, there are ten possible digits, with values 0 through 9.
  In base 2, there are 2 possible digits (*bits*), with values 0 and 1.
- In base 10, $1010$ means what? What about in base 2?

Slide 4

## Converting Between Bases

- Converting from another base to base 10 is easy if tedious (just use definition).
- Converting from base 10 to another base? Let's try to develop an "algorithm" (procedure) for that . . .

**Slide 5**

## Decimal to Binary, Take 1

- One way is to first find the highest power of 2 smaller than or equal to the number, write that down, subtract it from the number, and continue:
  1. If $n = 0$, stop.
  2. Find largest $p$ such that $2^p \leq n$.
  3. Write a 1 in the $p$-th output position.
  4. Subtract $2^p$ from $n$.
  5. Go back to first step.
- Is this okay? What's not quite right about it? (We don't say what to put in the positions that don't have ones in them.)
- (Example.)

**Slide 6**

## Decimal to Binary, Take 2

- Another way produces the answer from right to left rather than left to right, repeatedly dividing by 2 (again $n$ will be the number we want to convert):
  1. If $n = 0$, stop.
  2. Divide $n$ by 2, giving quotient $q$ and remainder $r$.
  3. Write down $r$.
  4. Set $n$ equal to $q$.
  5. Go back to first step.
- Is this okay? What's not quite right about it? (We don't say to write down the remainders from right to left.)
- (Example.)

Slide 7

## Recap

- Key ideas here — break problem down into a sequence of steps that we hope don't require much intelligence, just an ability to calculate, with some decision-making and repeating.
- Before moving on/back to programming, a little more about different number bases and how binary numbers are used to represent data . . .

Slide 8

## Octal and Hexadecimal Numbers

- Binary numbers are convenient for computer hardware, but cumbersome for humans to write. Octal (base 8) and hexadecimal (base 16) are more compact, and conversions between these bases and binary are straightforward.
- To convert binary to octal, group bits in groups of three (right to left), and convert each group to one octal digit using the same rules as for converting to decimal (base 10).
- Converting binary to hexadecimal is similar, but with groups of four bits. What to do with values greater than 9? represent using letters A through F (upper or lower case).
- (Examples.)

Slide 9

## Computer Representation of Integers

- Computers represent everything in terms of ones and zeros. For non-negative integers, you can probably guess how this works — number in binary. Fixed size (so we can only represent a limited range).
- How about negative numbers, though? No way to directly represent plus/minus. Various schemes are possible. The one most used now is "two's complement": Motivated by the idea that it would be nice if the way we add numbers didn't depend on their sign. So first let's talk about addition . . .

Slide 10

## Machine Arithmetic — Integer Addition and Negative Numbers

- Adding binary numbers works just like adding base-10 numbers — work from right to left, carry as needed. (Example.)
- Two's complement representation of negative numbers is chosen so that we easily get 0 when we add $-n$ and $n$.

  (To be continued.)

**Slide 11**

## Minute Essay

- What is $1011_2$ in base 10?
- What's the (base 10) value of the largest number you can represent with 4 bits? (E.g., the largest number you can represent with 2 bits is $11_2$, or $3_{10}$.)

**Slide 12**

## Minute Essay Answer

- $1011_2$ is $11_{10}$.
- 15 ($1111_2$).