COMS 1003: Introduction to Computer Programming in C

Complex Data Types

October 13\textsuperscript{th} 2005
Announcements

• Read the academic honesty policy.
• Re-read the academic honesty policy.
• See me if you have questions.
• Study for the midterm.
• Get working on HW2.
Outline

• Recall primitive types
• Explore complex data types
  - struct
  - enum
  - union
  - bit fields
Primitive Types

• Recall von Nuemann memory model

• 1 address holds a 32 bit value
  - fine for primitive types, enough space

• primitive types:
  - int, float, char, double
Multi-faceted Types

• what about types that
  - have multiple dimensions or properties
  - are aggregates of primitive types
• storage model doesn't work very well
  - how do you store a complex type in 1 32-bit memory cell?
Example: represent an MP3

• A simple int doesn't cut it
• Multiple properties about 1 logical entity

/* Some properties for an MP3 */
char file_name[256] = {0};
char song_name[256] = {0};
long length = 0;
int bit_rate = 144;
Repeating Properties

- run out of variable names
- parallel maintenance of data
- need a template for this logical collection of data
A Structure (struct keyword)

- collection of logically-related data

```c
struct song
{
    char filename[256];
    char songname[256];
    long length;
    int bitrate;
    char data[1000000];
};
```
Enumerations

• A way to declare a set of constants

• Has scope

```c
enum days {MON, TUES, WED, THUR, FRI};
enum month {Jan=1, Feb, Mar, Apr, May
            Jun, Jul, Aug, Sept, Oct
            Nov, Dec};
enum {FALSE=0, TRUE};
```
Unions

• Like a struct but has multiple personalities depending on context

```c
union pet
{
    char cat;
    int dog;
    float turtle;
}
```
Struct vs. Union

• Struct is all things at once; distinct memory cells are allocated for all members

• Union has memory allocated for the largest member
  - union instance is treated as only 1 member at a time
  - programmer must keep track