

COMP1917: 12 Structs Part 2

Sim Mautner

s.mautner@unsw.edu.au

September 5, 2016

Purpose

- Anything we could put in a spreadsheet.
- Previously we could have put each column in an array.
- The problem is if we swap two items, or remove an item from one array, we have to update all the other arrays so that the elements stay lined up.
- Alternative: Each row goes into a single struct. And we have an array of structs.
- Examples:
 - ▶ http://dogeit.weebly.com/uploads/3/0/1/5/30151923/416766_orig.gif
 - ▶ http://www.imagebon.com/postpic/2009/09/sales-spreadsheet-example_7285.png

Accessing Members of a Struct

- When the variable is a struct, use the `.` notation. (Eg `date.day`)
- When the variable is a pointer to a struct, use the `->` notation. (Eg `date->day`)
- The `->` notation is short-hand for dereferencing the pointer, and then using the dot notation. The following two lines are equivalent:
 - ▶ `date->day`
 - ▶ `(*date).day`
- This is often used when the struct has been malloced, and so we are working with the pointer to the struct.
- Ex 1: Create a variation of `Date.c` and `Date.h` which uses `malloc` to declare the structs on the heap, and passes around a pointer to the struct.