COMP1511 19T1
Week 7, Tuesday: Structure and Composition

Jashank Jeremy
jashank.jeremy@unsw.edu.au

references and indirection
structured data
Administrivia
Don’t panic!

Assignment 1: Coco
out now ... due 7 April 23:59:59

Weekly Test #4
due tomorrow, 3 April 23:59:59

No Marc!
on week06tue, week06thu, week07tue
lectures by Jashank, instead.
Memory is a linear array of lots of boxes: **bytes**

variables are a group of bytes:

```
int i = 42;
```

variables have size, location

variables can store locations of other variables

```
int *ip = &i;
```
Memory is a linear array of lots of boxes: bytes

arrays are contiguous sequences of variables:

```c
char str[] = "Hello!";
```

pointers and arrays are mostly interchangeable;

‘*’ is mostly equivalent to ‘[]’

arithmetic on pointers is well-defined but horrific
New Operations
Reference and Dereference

`&`
reference, address-of;
‘where is this variable in memory?’

`*`
dereference, indirection
‘what’s at this location in memory?’

`sizeof`
‘how big is this variable or type?’
main is a rather peculiar function.

```c
int main(void);
```
main is a rather peculiar function.

```c
int main(void);
```

Except... that’s not the only way to do it.

```c
int main(int argc, char *argv[]);
```

**argc**: the argument count;

**argv**: the argument vector
No Argument From Me

What's That ‘argv’ Thing?

```
/ my prog \0
- o \0
d cc \0
d cc . c \0
```

NULL
Structured types let us compose our own complex expressions.

```
struct tag {
    member-type member-name,
    ...
};
```