

COMP1917: 08 Pointers and Functions

Sim Mautner

s.mautner@unsw.edu.au

August 15, 2016

Pointers

- All variables are stored at a specific address in the computer's memory.
- We can store and inspect the address of any variable.
- & is used to say “address of” .
- Ex 1: Write an application which declares some variables and displays the variables as well as their addresses.

Pointers

- We can store the “address of” a variable into another variable. The type of this other variable is a pointer, and depends on the type of the variable it’s pointing to. (Eg a pointer to an `int` would be of type “int pointer” and is written as `int *`).
- Ex 2: Write an application which declares some variables, declares some variables to hold the addresses of those variables, and then prints them out.
- If we only have a pointer, we can access the “contents of” that pointer, by *dereferencing* it.
- Ex 3: Adapt Ex 2 to use an approach of dereferencing the pointers.

Functions: Pass by Value, Pass by Reference

- When passing parameters into a function, we cannot change the value of the original variable.
- If we want to change the value of the original variable, we can do this by passing in the address of the variable (or a pointer to the variable).
- Ex 4: Write a `void` function (and an application to use it) which takes in a pointer to an `int`, and multiplies the value by 2.
- This approach is useful if we want a function to effectively have more than one return value. (Note: it still has only a maximum of 1 return value, but by changing the value of parameters, more information can be passed back to the function which called.)