

C - FUNCTIONS

FUNCTIONS

Example:

```
double sum(double a, double b) {  
    double res;  
    res = a+b;  
    return res;  
}
```

FUNCTIONS (CONT.)

- Can go before or after main
- If after, must put function prototype before
 - `double sum(double a, double b);` or
 - `double sum(double, double);`
- functions that don't return anything have `void` return type
- can only return 1 value
- best to get in habit of using function prototype and defining later
 - will come up again when we look at bigger programs

FUNCTIONS

Pass-by-reference or Pass-by-value?

- C is always pass-by-value
 - sometimes that value is a value
 - sometimes it is a pointer

SWAP FUNCTION IN C

- Suppose you want function to swap to integers (returns void)
- Take 10 mins and try to get this function working
- Pass-by-value, think carefully about arguments

FUNCTIONS AND POINTERS

- If you want to be able to change the variable:
 - pass a pointer
 - function declaration must specify argument is pointer
- Passing arrays
 - Always passes a pointer
 - Typically pass size as separate argument
- Returning arrays
 - Either return pointer (or modify in place)