Functions, Multi-file Programs, Variable Scope & Lifetime

Functions

- Functions
  - Every C program has at least one function (main) and can define additional functions
  - A function takes in an input (**argument**) and returns an output (**return**)
  - Functions allow code reuse
    - If we define a function once, we can call it many times
    - Makes the program much easier to change!
  - Functions allow code modularisation
    - Interaction with the rest of program is explicit and limited
    - We can consider the code in isolation
    - Makes the program easier to read, test and debug
  - Functions reduce complexity of program
- Function definition
  - A function **definition** provides the body of the function
    - return_type function_name (parameter list) {
      // body of function
      }
  - Function header
    - **Return type**: the data type of the value the function returns
      - int: returns int values
      - double: returns double values
      - void: returns no values
    - **Function name**: name of the function, e.g. "max", "power"
    - **Parameter list**: list of type, order and number of parameters used in a function
  - Function body
    - **Statements:** define the function's role
    - **Variables:** each function has its own variables which are created when the function is called and destroyed when the function returns
      - A function's variables are not accessible outside the function
    - **Return statement**: stops execution of a function, specifying value to return to (unless function is of type void)
      - A run-time error occurs if end of non-void function is reached without return
- Function declaration
  - A function **declaration** provides key info about a function's name, return type, number/type of parameters, and how to call the function (the **function prototype**)
  - The declaration takes the form return_type function_name (parameter list)
    - e.g. double power(double x, int n); int max (int, int)
  - Declarations allow top-down order of functions in file, making it more readable