Lec 8 Functions Continue

Why do we need function?

type func_name(type_1 name1, …, type_n name_n)

return type

name of the function

input names and their types

Questions to think about

- Can we define two functions with the same name?
- How to define a function to handle different data type correctly?
- Can we define a function after the main function?

Can we define two functions with the same name?

- **Yes**, if the input types for the two functions are different; otherwise, **no**.

Can we define a function after the main function?

- **Yes**, if we **declare** the function **before** the main function.

- **Declaring** a function means specifying a type for what is named, no details needs to be provided for the function body.

- Example: double f(double);

- It simply tells the compiler that you can use the name, it doesn't allocate memory.

Definition vs Declaration

- Definition
  - the implementation
  - provide function body
  - consume memory

- Declaration
  - the interface
  - only supply argument types and return type
  - doesn't consume memory

Scope of variables

- Variables defined inside a function body are local variables, they cannot be accessed by other functions (including main function).

- When we call a function, we actually pass a copy of the given variable to the function.

Pass by reference

- Question: give two integer variables x, and y, write a function to swap their values.

Why pass by reference?

- avoid copying data;

- want to make some changes to the actual variable

Recursive function

- A function makes a call to itself inside the function body.

- To prevent infinite recursion, you need an if-else statement where one branch makes a recursive call, and the other branch does not.

- The branch without a recursive call is usually the base case

Example: Fibonacci sequence

- The Fibonacci sequence is the following:

  1, 1, 2, 3, 5, 8, …

- In general, we have F(n) = F(n-1) + F(n-2).

- Write a function to calculate the nth entry of Fibonacci sequence.