CS102 – Review for exam over Chapter 5

Vocabulary to know

Parameter	Function prototypes	Reference parameters
Argument	Function stubs	Constant references
Function header	Shadowing	Stepwise refinement
Function body	Call-by-value	Top-down design
Unreachable code	Call-by-reference	

Be able to code a function. This means

1. Be able to design a function header
   This header should include the function's type, its name, and its input parameters.
2. Be able to write the body too.
3. Be able to write a statement that calls your function.
   This could be a cout, or calculation.

Be able to write code that calls a prewritten function, given its type and inputs..

Benefits of Using Functions–Know these!

They simplify the coding process and speed up the development process
They are reusable
    If you have to use the same code in different places, you don't have to rewrite it
They make testing easier
They make it easier for a team to design a large project

Special comments that come immediately before functions (JavaDoc)

Brief description
@param – one per paramter
@return – what gets returned
Know what they are and be able to use them

The return Statement

Code after a return statement is not executed
A function of type void should not have a return
A function of any other type must have a return that matches the function type

General function ideas

Why are there void functions?
Functions should always be short
A function should do only one task

Parameters in functions
The default in C++ is call by value
For the most part, you should use return values instead of references

Recursion
What is it?
Why do we use it?
Recursion always needs a stopping point
Recursion is much slower than solving the problem directly
For the factorial function, it's easy to code directly and more efficient

Just to be explicit: You *will* have to write C++ code, and determine what is calculated or printed by a program fragment.