**CS 10A – Declarations and Definitions …C++ concepts to think about … (Information Sheet)**

**Why does C++ offer both declarations and definitions?** The declaration/definition distinction reflects the fundamental distinction between what we need to use something (an interface) and what we need for that something to do what it is supposed to (an implementation).

For a variable (introduced in Gaddis Chapter 2), a declaration supplies the type but only the definition supplies the object (the memory).

For a function (introduced later in Gaddis 6), a declaration again provides the type (argument types plus return type) but only the definition supplies the function body (the executable statements). Note that function bodies are stored in memory as part of the program, so it is fair to say that function and variable definitions consume memory, whereas declarations don’t.

The declaration/definition distinction allows us to separate a program into many parts that can be compiled separately. The declarations allow each part of a program to maintain a view of the rest of the program without bothering with the definitions in other parts. As all declarations (including the one definition) must be consistent, the use of names in the whole program will be consistent.

**Declaration Definition**

int value; value=10;

or int value=10;

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

//a function defined

double sqrt(double d) // a function prototype double sqrt(double d)

{

// pow() is a predefined function in math.h header file

return pow(d, 0.5);

}