repercussions in the client’s behavior.

Our Story to Date

The first step in specifying a class design is providing a class declaration. The class
declaration is modeled after a structure declaration and can include data members and
function members. The declaration has a private section, and members declared in that
section can be accessed only through the member functions. The declaration also has a
public section, and members declared there can be accessed directly by a program using
class objects. Typically, data members go into the private section and member functions go
into the public section, so a typical class declaration has this form:

class className
{
private:
    data member declarations
public:
    member function prototypes
};

The contents of the public section constitute the abstract part of the design, the public
interface. Encapsulating data in the private section protects the integrity of the data and is
called data hiding. Thus, the class is the C++ way of making it easy to implement the OOP
goals of abstraction, data hiding, and encapsulation.

The second step in specifying the class design is to implement the class member functions.
You can use a complete function definition instead of a function prototype in the class
declaration, but the usual practice, except for very brief functions, is to provide the function
definitions separately. In that case, you need to use the scope operator to indicate to which
class a member function belongs. For example, suppose the Bozo class has a member
function called Retort() that returns a pointer to a char. Then, the function heading would
look like this:

    char * Bozo::Retort()

In other words, Retort() is not just a type char * function; it is a type char * function thatbelongs to the Bozo class. The full, or qualified, name of the function is Bozo::Retort().