

Most of these notes are based on **Chapter 3** of D & D.

“Composite types” contain multiple elements. In C, a struct can contain multiple variables, or pieces of data, all grouped together

Classes, however, can contain *data members* and *member functions*

Classes are useful because they hide details, just as functions in C hide some of the details. The difference is classes also can handle their own variables by design (more later)

The data members are like the attributes of an object and are similar to the data stored in a C struct

Member functions perform the actions, and are associated with specific data members

Note that you must create an object before you can access any member functions
This is part of the reason why C++ is called an object-oriented language

In general, the programs you develop will have a handful of classes and one main program that uses public member functions.

Public is an “access-specifier” label that states that these are “available to the public”, in this case functions outside the class such as the main function

As a rule of thumb, data members are usually private and member functions are public. That said, any function not needed by the main function should also be made private

Discussion of Fig 3.1 and associated example code

As per D & D convention, words in a class will be capitalized, e.g. GradeBook. This is also called “camelcase” because it is similar to the humps of a camel.

Like a C struct, a elements of a class are enclosed by { } and ended with a ;

Also by convention, the first word of a member function starts with a lower case character and all subsequent words are uppercase

Discussion of Fig 3.3 and associated example code

String is an object of the C++ standard template library (STL). We will discuss the STL in greater depth later in the semester.

As per the example given in class, using “cin << strname” will only read up until the first “ “ character. For this reason the member function getline is more useful

Discussion of Fig 3.5 and associated example code

More about access specifiers: note by default all members of a class are private unless specifically denoted by “public:”

Although you may interchange public and private designations, it is considered good practice to group all of the public elements first followed by private elements.

Data members are typically private because we want to “hide” these from the program accessing them. We sometimes will refer to such a main program as the “client”

Note that this often time will save time debugging and making modifications: data management, processing, and display can differ based on what the programmer decides. For example, although the private variable of the current time may be represented as seconds since 1980, the client may only see the date in “hh:mm:ss”. This is what “data hiding” actually is.

Any function that “sets” data members via the client are sometimes called “mutators”

Similarly, any function that “reads” information stored in private data members are sometimes called “accessors”

Also note that the mutators and accessors may be called within the class they are a part of for displaying and managing data. As such, it is possible for the programmer to hide data from the member functions as well (see time example above)