6.2 Classes

Classes are similar to structs, where both are User defined data types containing data members and member functions.

```
class Date {
public:
        // member function(s)
        void output();

        // data members
        int month, day, year;
};
```

Creating a variable of type < *Class* > is creating an **object**.

    Example: Date d;   // d contains an object

The definition of a member function can be written either inside or outside of the class. If written outside of the class, member functions are written the same way as any normal function, except the name of the function is defined by the Class name followed by the scope resolution operator **::** to specify which class it is a member of.

```
void Date::output() {
        cout << month << "/" << day << "/" << year;
}
```

Encapsulation

- Also known as information hiding, data abstraction; encapsulation is one of the main principles of OOP
- Idea: set data members to **private** to prevent being accessed directly outside of the class and member functions.
- Instead of accessing data members directly, create **accessor** and **mutator** functions that will allow accessing data through a "middleman", and set the access to **public**

Note: Data members and member functions are set to **private** by default in a class.
In contrast, structs are set to **public** by default.