

## C++ 8\_1

### 8.1 Classes Basics

Classes are an expanded concept of data structures: like data structures, they can contain data members, but they can also contain functions as members.

An object is an instantiation of a class. In terms of variables, a class would be the type, and an object would be the variable.

Classes are defined using either keyword `class` or keyword `struct`, with the following syntax:

```
class class_name {  
    access_specifier_1:  
    member1;  
    access_specifier_2:  
    member2;  
    ...  
} object_names;
```

Where `class_name` is a valid identifier for the class, `object_names` is an optional list of names for objects of this class. The body of the declaration can contain members, which can either be data or function declarations, and optionally access specifiers.

Classes have the same format as plain data structures, except that they can also include functions and have these new things called access specifiers. An access specifier is one of the following three keywords: `private`, `public` or `protected`. These specifiers modify the access rights for the members that follow them:

`private` members of a class are accessible only from within other members of the same class (or from their "friends").

`protected` members are accessible from other members of the same class (or from their "friends"), but also from members of their derived classes.

Finally, `public` members are accessible from anywhere where the object is visible.

By default, all members of a class declared with the `class` keyword have `private` access for all its members. Therefore, any member that is declared before any other access specifier has `private` access automatically.

### Defining C++ Objects

A class provides the blueprints for objects, so basically an object is created from a class. We declare objects of a class with exactly the same sort of declaration that we declare variables of basic types. Following statements declare two objects of class `Box`:

```
Box Box1;  
// Declare Box1 of type Box  
Box Box2;  
// Declare Box2 of type Box
```

Both of the objects `Box1` and `Box2` will have their own copy of data members.