1. 一般来说，如果非成员函数是类接口的组成部分，则这些函数的声明应该与类在同一个头文件内。
2. Ordinarily, functions that do output should do minimal formatting. That way user code can decide the format.
3. Classes that have members of built-in or compound type(such as pointers and arrays) usually should rely on the **synthesized default constructor** **only** if all such members have in-class initializers
4. The **=default** can appear with the declaration inside the class body or on the definition outside the class body. Like any other function. if the =default appears inside the class body, the default constructor will be inlined
5. Constructors should not **override** **in-class initializers** except to use a different initial value. If you can’t use in-class initializers, each constructor should explicitly initialize every member of built-in type
6. Members that do not appear in the constructor initializer list are initialized by the corresponding in-class initializer (if there is one) or are default initialized.
7. When we provide an in-class initializer, we must do so **following an = sign** or **inside braces**.
8. A const member function that return \*this as a reference should have a return type that is a **reference to const.**
9. **class string str**;(C-style) equals to **string str**;
10. A friend function can be defined inside the class body. Such functions are implicitly **inline**.
11. Member function **definition** are processed **after** the compiler processes **all of the declarations** in the class.
12. Type Names Are Special
    1. Ordinarily, an inner scope can redefine a name from an outer scope even if that name has already been used in the inner scope. However, in a class, **if** a member uses a name from an outer scope and that name **is a type**, then **the class may not subsequently redefine that name**.
13. **explicit**
    1. When a ctor is declared **explicit**, it can be used only with the direct form of initialization. Moreover, the compiler will not use this ctor in an automatic conversion.
    2. Although the compiler will not use an explicit ctor for an implicit conversion, we can use such ctors explicitly to force a conversion
14. static data members
    1. Even if a **const static** data member is initialized in the class body, that member ordinarily should be defined outside the class definition.
    2. **A static data member can have incomplete type** (p278). In particular, a static data member can have the same type as the class type of which it is a member.
    3. We can use a static member as a default argument. A nonstatic data member may not be used as a default argument because its value is part of the object of which it is a member. Using a nonstatic data member as a default argument provides no object from which to obtain the member’s value and so is an error.
    4. Ordinarily, class static member may not be initialized in the class body. However, we can provide in-class initializer for static member that have const integral type and **must** do so for static members that are constexprs of literal type.