对象在几种情况下都会被复制，比如当我们初始化一个变量时，或者当我们通过值传递或返回一个对象时（第6.2.1节和第6.3.2节）。当我们使用赋值运算符时，对象被赋值（第4.4节）。当对象不再存在时，它们就被销毁了，比如当一个局部对象从它创建的块中退出时被销毁了（第6.1.1节）。当向量（或数组）被销毁时，存储在向量（或数组）中的对象也被销毁。

文本, 信件

描述已自动生成

许多需要动态内存的类可以（通常应该）使用向量或字符串来管理必要的存储。使用向量和字符串的类避免了分配和释放内存的复杂性。

在你知道如何定义第13章中介绍的操作之前，类分配的资源应该直接存储为类的数据成员

作为编程风格的一个问题，当我们定义一个类，希望它的所有成员都是公共的，我们使用结构。如果我们希望有私有成员，那么我们使用类。

一个类可以允许另一个类或函数访问它的非公共成员，方法是将该类或函数设为友元。一个类使一个函数成为它的朋友，方法是在该函数的前面包含一个关键字friend的声明：

通常，最好在类定义的开始或结束时将友元声明分组在一起。

用户代码不会无意中损坏封装对象的状态。

封装类的实现可以随着时间的推移而改变，而不需要改变用户级代码。

虽然用户代码不需要在类定义更改时更改，但使用类的源文件必须在类更改时重新编译。

许多编译器不强制执行友元函数必须在类外声明才能使用的规则。

有时（但不经常）会发生这样的情况：类具有我们希望能够修改的数据成员，即使是在const成员函数内部也是如此。

**如果一个 const 成员函数返回 \*this 作为引用，那么它的返回类型应该是 const 引用**。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

当我们在一个对象上调用display时，该对象是否为const决定了调用哪个版本的display：

即使两个类具有完全相同的成员列表，它们也是不同的类型。每个类的成员都不同于任何其他类（或任何其他作用域）的成员。

重要的是要明白友谊是不可传递的。也就是说，如果类Window\_mgr有自己的友元，这些友元就没有对Screen的特殊访问权。

每个类控制哪些类或函数是它的朋友。

首先，定义Window\_mgr类，它声明但不能定义clear。

在clear可以使用Screen的成员之前，必须声明Screen。

接下来，定义类Screen，包括一个用于clear的friend声明。·最后，定义clear，现在可以引用Screen中的成员。

友元声明影响访问，但不是普通意义上的声明，理解这一点很重要

成员函数定义在编译器处理完类中的所有声明之后才被处理。

* 类成员函数中的名字查找分为两个阶段：
  + 第一阶段处理成员声明。
  + 第二阶段处理函数体。
* 由于这种分阶段的处理方式，类成员函数可以访问类的所有成员，即使这些成员在函数体之后声明。

即使外部对象是隐藏的，仍然可以通过使用范围操作符访问该对象

我们必须使用构造函数初始化列表来为常量、引用或没有默认构造函数的类类型的成员提供值。

* 初始化和赋值的区别在于初始化的效率更高，且某些成员必须通过初始化列表进行初始化。
* 使用构造函数初始化列表可以避免编译错误，提高效率，并使代码更清晰。
* 对于常量成员、引用成员和没有默认构造函数的类成员，必须使用初始化列表。

最好按照成员声明的顺序编写构造函数初始值设定项。此外，如果可能的话，避免使用成员初始化其他成员。

**在实践中，如果定义了其他构造函数，通常也应该提供一个默认构造函数**。

图形用户界面, 文本

描述已自动生成

我们可以通过将构造函数声明为显式来防止在需要隐式转换的上下文中使用构造函数

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

类的静态成员存在于任何对象之外。对象不包含与静态数据成员关联的数据。因此，每个Account对象将包含两个数据成员--所有者和金额。只有一个interestRate对象将由所有Account对象共享

* **静态数据成员：**
  + **属于类本身，而不是类的某个对象。**
  + **所有对象共享同一个静态数据成员。**
  + **必须在类外定义和初始化。**
* **静态成员函数：**
  + **属于类本身，而不是类的某个对象。**
  + **没有 this 指针，不能访问非静态成员。**
  + **不能声明为 const。**
  + **不能直接调用非静态成员函数。**

与任何类成员一样，当我们在类体外部引用类静态成员时，我们必须指定定义该成员的类。但是，static关键字仅用于类体内的声明

确保对象只定义一次的最佳方法是将静态数据成员的定义放在包含类非内联成员函数定义的同一个文件中。