应对这些声明进行注释。这样，任何阅读代码的人都可以看到该语句是故意省略的。

无关的null语句并不总是无害的。

许多编辑器和开发环境都有工具来自动修改源代码以匹配其结构。这是一个好主意，使用这些工具，如果他们可用。

重要的是要理解执行流程是跨案例标签的。在一个case标签匹配后，执行从该标签开始，并继续执行所有剩余的case，或者直到程序显式地中断它。为了避免为后续的case执行代码，我们必须显式地告诉编译器停止执行。在大多数情况下，下一个case标签之前的最后一个语句是break。.

虽然没有必要在开关的最后一个标签之后包含中断，但最安全的做法是提供一个中断。这样，如果稍后添加了其他案例，则中断已经存在

最佳实践即使默认情况下没有工作，定义默认标签也很有用。定义一个空的默认节向后续读者表明该案例已被考虑。

在while条件或while主体中定义的变量在每次迭代时创建和销毁。

图形用户界面, 文本, 电子邮件

中度可信度描述已自动生成

值得记住的是，在for头中定义的任何对象的可见性都限于for循环的主体。因此，在本例中，在for完成后，index不可访问。

C++ offers four jumps: break,

continue, and goto, which we cover in this chapter, and the return statement,

continue语句终止最近的封闭循环的当前迭代，并立即开始下一次迭代。

程序不应该使用goto。goto使程序难以理解和修改。.

异常是运行时异常，如丢失数据库连接或遇到意外输入，这些异常存在于程序的正常功能之外。处理异常行为可能是设计任何系统最困难的部分之一。

· throw表达式，检测部分使用它来指示它遇到了它无法处理的东西。我们说一个throw引发一个异常。

try块，处理部分使用它来处理异常。try块以关键字try开始，以一个或多个catch子句结束。在try块中执行的代码抛出的错误通常由catch子句处理。因为它们“处理”异常，catch子句也被称为异常处理程序。

一组异常类，用于传递有关在抛出和关联的捕获之间发生了什么的信息。

与任何块一样，在try块内声明的变量在块外是不可访问的-特别是catch子句不能访问它们。

重要的是要认识到异常会中断程序的正常流程。在异常发生时，调用者请求的一些计算可能已经完成，而其他计算仍然未完成。一般来说，绕过程序的一部分可能意味着一个对象处于无效或不完整的状态，或者一个资源没有被释放，等等。在异常处理过程中正确“清理”的程序被称为异常安全。编写异常安全的代码是非常困难的，并且（很大程度上）超出了本语言入门的范围。

有些程序使用异常只是为了在异常情况发生时终止程序。这样的程序通常不担心异常安全。

处理异常并继续处理的程序通常必须不断地知道是否可能发生异常，以及程序必须做些什么来确保对象有效，资源不泄漏，以及程序恢复到适当的状态

我们偶尔会指出用于促进异常安全的特别常见的技术。然而，程序需要健壮异常处理的读者应该意识到，我们所涉及的技术本身不足以实现异常安全。