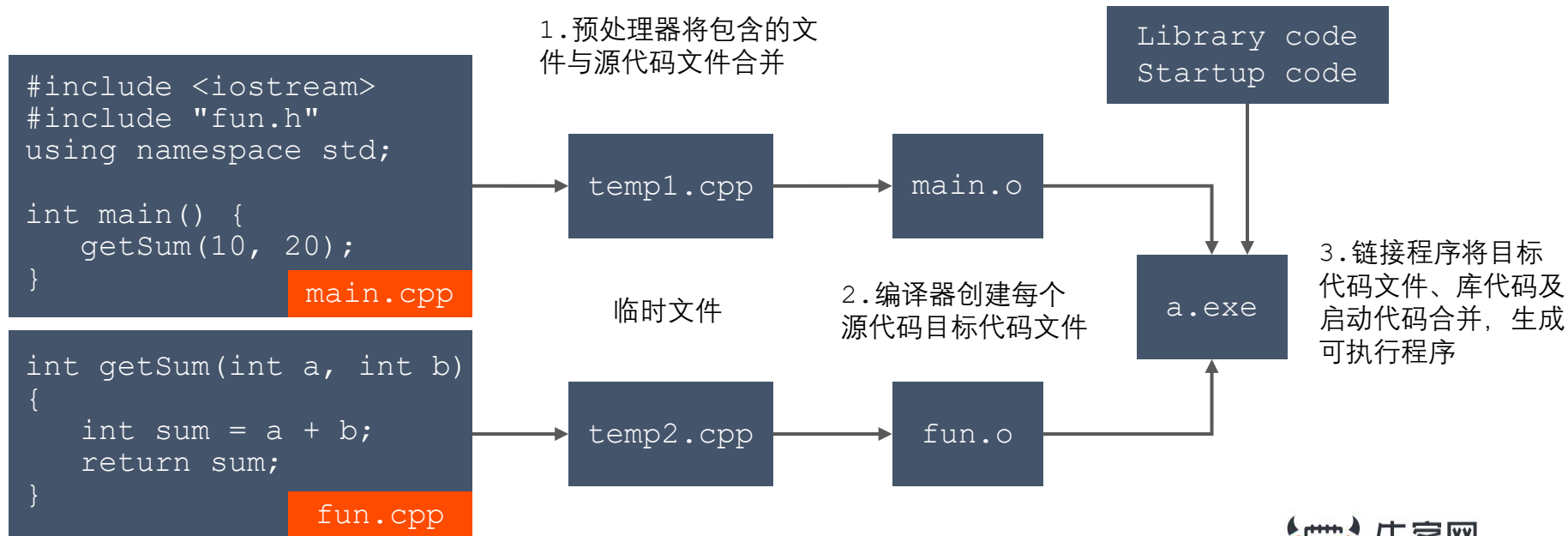


# 存储持续性、作用域、链接性

## ► 单独编译

- C++ 多文件编程中，可以单独编译这些文件，然后将它们链接成可执行程序。
- 编译多个源代码文件的命令：`g++ main.cpp fun.cpp`



## ► 存储持续性

■ C++使用三种不同的方案来存储数据，这些方案的区别就在于数据保留在内存中的时间。

1. 自动存储持续性：在函数定义中声明的变量（包括函数参数）的存储持续性为自动的。它们在程序开始执行其所属的函数或代码块时被创建，在执行完函数或代码块时，它们使用的内存被释放。
2. 静态存储持续性：在函数定义外定义的变量和使用关键字 `static` 定义的变量的存储持续性都为静态，它们在程序整个运行过程中都存在。
3. 动态存储持续性：用 `new` 运算符分配的内存将一直存在，直到使用 `delete` 运算符将其释放或程序结束为止。这种内存的存储持续性为动态，有时被称为自由存储（free store）或堆（heap）。

## ► 作用域和链接性

- 作用域 (scope) 描述了名称在文件 (翻译单元) 的多大范围内可见。例如, 函数中定义的变量可在该函数中使用, 但不能在其他函数中使用; 而在文件中的函数定义之前定义的变量则可在所有函数中使用。
- 链接性 (linkage) 描述了名称如何在不同单元间共享。链接性为外部的名称可在文件间共享, 链接性为内部的名称只能由一个文件中的函数共享。自动变量的名称没有链接性, 因为它们不能共享。

# Thanks

