在C++中，外部链接（external linkage）是指标识符（变量、函数、类等）可以在不同的编译单元（源文件）中进行共享和访问。具有外部链接的标识符可以在一个源文件中定义，并在其他源文件中进行声明和使用。

默认情况下，全局变量、非静态函数和非内联函数具有外部链接。这意味着它们可以在不同的源文件中进行共享和访问。

需要注意的是，如果要在多个源文件中使用同名的全局变量或函数，它们的类型和定义必须一致，否则会导致链接错误。另外，静态变量和函数具有内部链接（internal linkage），它们只能在定义它们的源文件中访问。

可以使用关键字extern在一个源文件中声明在其他源文件中定义的标识符，以便在当前源文件中使用它们。在声明中使用extern告诉编译器该标识符具有外部链接，并且可以在其他源文件中找到其定义。

总结起来，外部链接允许在不同的源文件中共享和访问标识符。具有外部链接的标识符可以在一个源文件中定义，并在其他源文件中进行声明和使用。

static 关键字可用于声明变量、函数、类数据成员和类函数。

默认情况下，在所有块的外部定义的对象或变量具有静态持续时间和外部链接。静态持续时间意味着，在程序启动时分配对象或变量，并在程序结束时释放对象或变量。外部链接意味着，变量的名称在用于声明变量的文件的外部是可见的。相反，内部链接意味着，名称在用于声明变量的文件的外部是不可见的。

在以下情况下，可使用static关键字。

·在文件范围（全局和/或命名空间范围）内声明变量或函数时，static关键字将指定变量或函数具有内部链接。在声明变量时，变量具有静态持续时间，并且除非您指定另一个值，否则编译器会将变量初始化为 0。

·在函数中声明变量时，static关键字将指定变量将在对该函数的调用中保持其状态。

·在类声明中声明数据成员时，static 关键字将指定类的所有实例共享该成员的一个副本。必须在文件范围内定义静态数据成员。声明为const static的整型数据成员可以有初始值设定项。

·在类声明中声明成员函数时，static关键字将指定类的所有实例共享该函数。 由于函数没有隐式this指针，因此静态成员函数不能访问实例成员。 若要访问实例成员，请使用作为实例指针或引用的参数来声明函数。

·不能将联合成员声明为静态的。但是，全局声明的匿名联合必须是显式声明的static。

编写程序过程中，名称（name）可以是符号常量、变量、函数、结构、枚举、类和对象等等。工程越大，名称互相冲突性的可能性越大。另外使用多个厂商的类库时，也可能导致名称冲突。为了避免，在大规模程序的设计中，以及在程序员使用各种各样的C++库时，这些标识符的命名发生冲突，标准C++引入关键字 namespace（命名空间/名字空间/名称空间），可以更好地控制标识符的作用域。

例如，我们在C语言中，通过static可以限制名字只在当前编译单元内可见，在C++中我们通过namespace来控制对名字的访问。

namespace的语法：

·namespace中可定义常量、变量、函数、结构体、枚举、类等

·namespace只能在全局定义。

·namespace支持嵌套定义。

·namespace是开放的，可随时添加新的成员。

·namespace关键字可以为已有空间名字增加别名

·无名命名空间意味着命名空间中的符号只能在本文件中访问，相当于给符号增加了static修饰。