**条款01:视C++为一个语言联邦**

我们该如何理解C++呢？

最简单的方法是将C++视为一个由相关语言组成的联邦而非单一语言。在其某个次语言中，各种守则与通例都倾向简单、直观易懂、并且容易。为了理解C++, 你必须认识其主要的次语言。幸运的是总共只有四个：

1. C语言
2. Object-Oriented C++：classes，封装、继承、多态等等。
3. Template C++：C++的泛型编程部分，由于templates威力强大，它们带来崭新的编程范型，也就是所谓的模板元编程。
4. STL：STL是个template程序库，它对容器、迭代器、算法以及函数对象的规约有极佳的紧密配合与协调。STL有自己特殊的办事方式，当你伙同STL一起工作，你必须遵守它的规约。

记住这四个次语言，当你从某个次语言切换到另一个，导致高效编程守则要求你改变策略时，不要感到惊讶。请记住C++高效编程守则视状况而变化，取决于你使用C++的哪一部分。