* **仿函数（Functors）**：在STL中，仿函数是一种可以像函数一样调用的对象，通常通过重载函数调用运算符()来实现。它们在STL中广泛使用，特别是在需要自定义行为的情况下。
* **关联容器**：如std::set和std::map，它们使用仿函数（例如比较函数）来保持元素的有序。
* **算法**：STL中的许多算法，如find\_if、for\_each和transform，接受仿函数作为参数，以定制它们的行为。
* **适配器**：如not1和bind2nd，它们生成或修改仿函数以改变其行为。
* **ptr\_fun、mem\_fun和mem\_fun\_ref**：这些是STL提供的函数适配器，它们允许你将普通函数指针或成员函数指针转换为仿函数，这样它们就可以用于STL算法。
* **编写良好的仿函数**：为了确保仿函数与STL的其他部分兼容，你需要遵循一些指导原则，比如保持它们的状态无关性（除非设计为状态ful仿函数），确保它们是可复用的，以及正确实现复制构造函数和赋值运算符（如果需要）
* **普通函数**：任何普通函数都可以隐式地转换为一个仿函数。
* **函数对象**：任何重载了operator()的类都可以作为仿函数。
* **lambda表达式**：C++11引入了lambda表达式，它们是匿名仿函数。
* **成员函数指针**：通过std::mem\_fn或其他适配器，可以将成员函数指针转换为仿函数。