C++ Primer Plus第十四章总结：

第十四章主要讲了C++中的代码重用。本章介绍了除公有继承外，进行代码重用的其他方式。比如说包含：如果一个类包含了另一个类的对象，则称为包含。包含可以用来模拟has-a关系，其中一个类包含另一个类的对象。也可以使用私有继承和保护继承来模拟这种关系。本章还讲解了类模板，它让程序员能够使用泛型定义类，然后使用模板根据具体类型创建特定的类。例如：栈模板能够使程序员创建整数栈或字符串栈。此外，本章还介绍了多重公有继承，使用着这种方式能够将一个类从多个类派生出来。

C++编程思想第九章总结：

虚基类是在C++中解决菱形继承问题的一种机制。当一个类从两个或多个不相关的类中继承时，会出现多重继承，导致菱形形状的类继承图。这种情况下，子类可能具有重复的基类部分，可能导致一些继承成员函数具有歧义。

虚基类背后的实现机制主要涉及以下几个方面：

1.虚继承：通过使用关键字virtual来声明基类的继承方式为虚继承。虚继承允许派生类共享基类的部分而不是整个基类，从而消除多重继承中的歧义。

2.虚函数表（vtable）：每个包含虚函数的类都会有一个虚函数表，该表包含了类中所有虚函数的地址。每个对象的虚函数表指针（vptr）都会指向其所属类的虚函数表。通过虚函数表，程序能够实现多态，即调用子类的函数而不是父类的函数。

3.初始化列表：在派生类的构造函数中，可以使用初始化列表来调用虚基类的构造函数，以确保正确的成员变量被初始化。在多继承中，必须使用初始化列表显式地调用每个基类的构造函数。

通过虚基类，我们可以实现更灵活和清晰的类层次结构，避免多重继承带来的歧义和复杂性。虚基类背后的实现机制涉及到虚继承、虚函数表和初始化列表等关键概念，这些机制共同协作，使得C++能够支持多态和灵活的类层次结构。