

# C++ Primer Plus学习笔记

## 第1章 预备知识

### 1.1 C++简介

C++融合了3种不同的编程方式：

- C语言代表的过程性
- 类代表的面向对象
- C++模板支持的泛型编程

### 1.2 C++简史

机器语言（汇编语言）->结构性语言（C、Pascal）->面向对象语言（C++）

#### 1.2.1 C语言

汇编语言依赖于计算机的内部机器语言。汇编语言是低级语言（low-level），即直接操作硬件，如直接访问CPU寄存器和内存单元。

因此汇编语言针对特定的计算机处理器，要将汇编程序移植到另一台计算机上，必须使用不同的汇编语言重新编写程序

高级语言（high-level）致力于解决问题，而不是针对特定硬件。

编译器将高级语言翻译成特定计算机的内部语言。

这样就可以通过对每个平台使用不同的编译器来在不同的平台上使用同一个高级语言程序了。

C语言将低级语言的效率、硬件访问能力和高级语言的通用性、可移植性融合在一起。

#### 1.2.2 C语言编程原理

计算机语言要处理两个概念：数据和算法。

数据是程序使用和处理的信息，而算法是程序使用的方法。

数据	算法
半杯黄油	奶油
1杯糖	逐渐加糖
2个鸡蛋	打鸡蛋
...	...

C语言是过程性语言，意味着它**强调的是编程的算法方面**。

### 1.2.3 面向对象编程

面向对象编程（OOP）**强调的是数据**。

在C++中，类是一种规范，它描述了这种新型数据格式，对象是根据这种规范构造的特定数据结构。

类是抽象的，它包含数据和类方法，将类实例化，即创建对象，对象保存了实际的数据值，我们可以用类方法操作对象实例的数据。

OOP程序设计方法首先设计类，它们表示了程序要处理的东西和对每个对象可执行的操作。然后我们可以设计一个使用这些类的对象的程序。

OOP编程不仅仅是将数据和方法合并为类定义，还可以**隐藏信息保护数据**。

C++的优点之一是：可以方便地**重用**和修改现有的，经过仔细测试的代码。

### 1.2.4 C++和泛型编程

泛型编程强调独立于特定数据类型。

要对不同类型的数据进行排序，通常必须为每一种类型创建一个排序函数。

泛型编程需要对语言进行扩展，以便可以只编写一个泛型（即不是特定类型的）函数，并将其用于各种实际类型。

### 1.2.5 C++的起源

C++是C语言的超集，这意味着任何有效的C程序都是有效的C++程序。

C++有C语言紧密联系硬件的能力（低级硬件访问），也有面向对象的特性（高级抽象）。

## 1.3 程序创建的技巧

假设编写一个C++程序。如何让他运行起来呢？

具体的步骤取决于计算机环境和使用的C++编译器，但大体如下：

1. 编写**源代码**
2. **编译**源代码。（将源代码翻译为机器语言）包含了翻译后的程序的文件就是程序的目标代码（object code）
3. 将目标代码与其他代码**链接**起来。（将库里的目标代码和使用函数的目标代码以及一些标准的启动代码组合起来，生成程序的运行阶段版本）包含该最终产品的文件被称为可执行文件

源代码->编译器->目标代码（obj）->链接程序（链接程序+库代码）->可执行代码（exe）