**面向过程的程序设计**

面向过程其实是最为实际的一种思考方式。可以说面向过程是一种基础的方法，它考虑的是实际的实现。一般的面向过程是从上往下逐步求精，所以面向过程最重要的是模块化的思想方法。与面向过程相比，面向对象的方法主要是把事物对象化，对象包括属性与行为。

当程序规模不是很大时，面向过程的方法还会体现出一种优势，因为程序的流程很清楚，用模块与函数的方法很好组织。

面向过程的程序设计方法主要由以下三种逻辑结构组成：

❑顺序结构。顺序结构是一种线性、有序的结构，它依次执行各语句模块。

❑循环结构。循环结构是重复执行一个或几个模块，直到满足某一条件为止。

❑选择结构。选择结构是根据条件成立与否选择程序执行的通路。

采用面向过程的程序设计方法，程序结构清晰，易于阅读、测试、排错和修改。由于每个模块执行单一功能，模块间联系较少，使程序编制比过去更简单，程序可靠性更强，而且增加了可维护性，每个模块可以独立编制、测试。

**C语言概述**

C语言是一种计算机程序设计语言，它既具有高级语言的特点，又具有汇编语言的特点。它由美国贝尔研究所的D.M.Ritchie于1972年推出，1978年后，C语言已先后被移植到大、中、小及微型机上，它可以作为工作系统设计语言，编写系统应用程序，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。它的应用范围广泛，具备很强的数据处理能力，不仅仅是在软件开发上，而且各类科研都需要用到C语言，适于编写系统软件，三维，二维图形和动画，具体应用比如单片机以及嵌入式系统开发。

**面向对象的程序设计**

面向对象编程（Object Oriented Programming, OOP）是一种计算机编程架构。OOP的一条基本原则是计算机程序是由单个能够起到子程序作用的单元或对象组合而成。OOP达到了软件工程的三个主要目标，即重用性、灵活性和扩展性。

面向对象程序的主要结构特点有两个：一个是程序一般由类的定义和类的使用两部分组成，在主程序中定义各对象并规定它们之间传递消息的规律；另一个是程序中的一切操作都是通过面向对象发送消息来实现的，对象接收到消息后，启动有关方法完成相应的操作。

面向对象设计的最大优点就是软件具有可重用性。当人们对软件系统的要求有所改变时，不需要程序员做大量的工作就能使系统做相应的变化。

面向对象程序设计中的概念主要包括对象和类，分别介绍如下：

1）对象。从概念上讲，对象代表着正在创建的系统中的一个实体。在日常生活中，对象就是我们认识世界的基本单元。对象是现实世界中的一个实体，整个世界就是由各种各样的对象构成的。例如，一个人、一辆汽车、一个足球等。

2）类。类是对象的模板，是对一组具有共同的属性特征和行为特征的对象的抽象。例如，由一个个大学生构成的“大学生”类，其中的每一个大学生是“大学生”类的一个对象。一个类的所有对象都有相同的数据结构，并且共享相同的实现代码。类和对象之间的关系是抽象和具体的关系。

面向对象系统中最主要的特性是封装性、继承性和多态性，分别介绍如下：

1）封装性。在面向对象程序设计中，数据的抽象是在确定类时强调对象的共同点而忽略了它们的不同点的结果。数据的封装则是隐藏了数据的内部实现细节的结果，将数据抽象的外部接口与内部的实现细节清楚地分开。

2）继承性。以面向对象程序设计的观点来看，继承所表达的是对象与类之间的关系，这种关系使得某类对象可以继承另外一类的特征和能力。继承机制为程序提供了一种组织、构造和重用类的手段。继承使一个类（基类）的数据结构和操作被另一个类即派生类重用，在派生类中只需描述其基类中没有的数据和操作。这样一来，就避免了公用代码的重复开发，减少了代码和数据冗余。

3）多态性。面向对象程序设计中的多态性，是指不同的对象收到相同的消息时所产生的多种不同的行为方式。C++语言支持两种多态性，即编译时的多态性和

运行时的多态性。编译时的多态性通过重载来实现；运行时的多态性通过虚函数来实现。程序运行的到底是函数的哪个版本，需要在运行时通过对象发送的消息来确定。

**C++概述**

程序员通常将C++读做“C plus plus”，“CPP”。C++是一种应用较广的计算机编程语言，是一种静态数据类型检查的、支持多重编程范式的通用程序设计语言。它支持过程化程序设计、数据抽象、面向对象程序设计、泛型程序设计等多种程序设计风格。

**C++的发展历程**

早期的计算机程序语言是计算机控制指令，每条指令就是一组二进制数，不同的计算都有不同的计算机指令集。使用二进制指令集开发程序是件很头痛的事，需要记住大量的二进制数。为了便于记忆，人们将二进制数用字母组合代替，以字符串关键字代替二进制机器码的编程语言称为汇编语言，汇编语言是低级语言，虽然汇编语言比机器码容易记忆，但仍然具有可读性差的缺点，大量的跳转指令和地址值很难让程序员在很短的时间内理解程序的意思，于是编程语言进入了高级语言时代。

第一个高级语言是美国尤尼法克公司在1952年研制成功的Short Code，但被广泛使用的高级语言是FORTRAN，它是由美国科学家巴克斯设计并在IBM公司的计算机上实现的，但FORTRAN语言和Algol60主要应用于科学和工程计算，随后出现了Pascal和C语言。C语言是在其他语言基础上发展起来的。首先是Richard Martin开发一种高级语言BCPL，随后Ken Thompson使用BCPL语言对其进行了简化，形成一门新的语言—B语言。但B语言没有类型的概念，Dennis Ritchie对B语言进行了研究和改进，在B语言基础上添加了结构和类型，并将这个改进后的语言命名为C语言，寓意很简单，因为在英文字母表中，字母C是字母B的下一个字母，这预示着语言的发展。

随着软件规模的增大，用C语言编写程序渐渐显得有些吃力了。于是，AT&T Bell(贝尔)实验室的Bjarne Stroustrup博士及其同事于20世纪80年代初在C语言的基础上成功开发了C++｡ Stroustrup经过钻研在C语言中加入类的概念，因此，C++最初的名字是C with Class。到1983年12月由Rick Mascitti建议改名为C Plus Plus，即C++。

首先提出类的概念的语言是Simula，它具有很高的灵活性，但无法胜任比较大型的程序。此后在Simula语言基础上发展的Smalltalk语言才是真正的面向对象语言，但Smalltalk-80不支持多继承。

C++从Simula继承了类的概念，从Algol68继承了运算符重载、引用，以及在任何地方声明变量的功能，从BCPL获得了用“//”注释，从Ada得到了模板、命名空间，从Ada、Clu和ML得到了异常。

C++是由C发展而来的，保留了C语言原有的所有优点，与C兼容。用C语言写的程序基本上可以不加修改地用于C++。从C++的名字可以看出它是C的超越和集中。C++既可用于面向过程的结构化程序设计，又可用于面向对象的程序设计，是一种功能强大的混合型的程序设计语言。

**C++的特点**

C++语言的运算符十分丰富，共有30多个，有算术、关系、逻辑、位、赋值、指针、条件、逗号、下标、类型转换等多种类型。

C++语言的数据结构多样，有整型、实型、字符型、枚举类型等基本类型，有数组、结构体、共用体等构造类型以及指针类型；还为用户提供了自定义数据类型，能够实现复杂的数据结构；还可以定义类实现面向对象编程，类和指针结合可以实现高效的程序。

C++语言的控制语句形式多样、使用方便。有两路分支、多路分支和虚幻结构几种控制语句，便于结构化模块的实现和控制，结合面向对象编程便于程序的编制和维护。

C++语言是一种面向对象的程序设计语言，采用抽象和实际相结合，各对象间使用消息进行通信，对象通过继承方法增加了代码的复用。

C++语言继承了C语言的特性，可以直接访问地址，进行位运算，从而能对硬件进行操作。C++语句具有编写简单方便、便于理解的优点，还具有低级语言与硬件结合紧密的优点。

C++语句具有很强的移植性，用C++编写的程序基本不用太多的修改就可用于不同型号的计算机上，C++标准可在多种操作系统下使用。

**C与C++**

C++语言是由C语言发展而来的，C语言是一种结构化语言，它的重点在于算法和数据结构。C程序的设计首要考虑的是如何通过一个过程，对输入或环境条件进行运算处理得到输出或实现过程（事务）控制。而对于C++，首先要考虑的是如何构造一个对象模型，让这个模型能够契合与之对应的问题域，这样就可以通过获取对象的状态信息得到输出或实现过程（事务）控制。

C与C++的最大区别在于它们解决问题的思想方法不一样，即它们在程序设计思想上不同。之所以说C++比C更先进，是因为“设计这个概念已经被融入到C++之中”，而就语言本身而言，在C中更多的是算法的概念，但是C是C++的子集。从上文可以看出，C实现了C++中过程化控制及其他相关功能，而在C++中的C，相对于原来的C还有所加强，引入了重载、内联函数、异常处理等概念，C++更是拓展了面向对象设计的内容，如类、继承、虚函数、模板和包容器类等。

再深入一点，在C++中，除了考虑数据封装、类型问题以外，还需要考虑诸如对象粒度的选择、对象接口的设计和继承、组合与继承的使用等问题。所以相对于C, C++包含了更丰富的“设计”的概念，但C是C++的一个自洽子集，也具有强大的功能，同样值得学习。

C++对C的“增强”，表现在六个方面：

（1）类型检查更为严格。

（2）增加了面向对象的机制。

（3）增加了泛型编程的机制（Template）。

（4）增加了异常处理。

（5）增加了运算符重载。

（6）增加了标准模板库（STL）。 面向对象程序设计，是针对开发较大规模的程序而提出来的，目的是提高软件开发的效率。不要把面向对象和面向过程对立起来，面向对象和面向过程不是矛盾的，而是各有用途、互为补充的。

学习C++，既要会利用C++进行面向过程的结构化程序设计，也要会利用C++进行面向对象的程序设计，更要会利用模板进行泛型编程。

C++和C的设计哲学并不一样，两者取舍不同，所以不同的程序员和软件项目会有不同选择，难以一概而论。与C++相比，C具备编译速度快、容易学习、显式描述程序细节、较少更新标准等优点，后两者也可同时视为缺点。在语言层面上，C++包含绝大部分C语言的功能(例外之一，C++没有C99的变长数组VLA)，且提供OOP和GP的特性。但其实用C也可实现OOP思想，亦可利用宏去实现某程度的GP，只不过C++的语法能较简洁、自动地实现OOP/GP。

C++的RAII(resource acquisition is initialization，资源获取就是初始化)特性比较独特，C/C#/Java没有相应功能。回顾历史，Stroustrup开发的早期C++编译器Cpre/Cfront是把C++源代码翻译为C，再用C编译器编译的。由此可知，C++编写的程序，都能用等效的C程序代替，但C++在语言层面上提供了OOP/GP语法、更严格的类型检查系统、大量额外的语言特性(如异常、RTTI等)，并且C++标准库也较丰富。有时候C++的语法可使程序更简洁，如运算符重载、隐式转换。但另一方面，C语言的API通常比C++简洁，能较容易供其他语言程序调用。因此，一些C++库会提供C的API封装，同时也可供C程序调用。相反，有时候也会把C的API封装成C++形式，以支持RAII和其他C++库整合等。

性能方面，相对运行于虚拟机语言(如C#/Java)，C/C++直接以静态形式把源程序编译为目标平台的机器码。一般而言，C/C++程序在编译及链接时可进行的优化最丰富，启动时的速度最快，运行时的额外内存开销最少。而C/C++相对动态语言(如Python/Lua)也减少了运行时的动态类型检测。此外，C/C++的运行行为是确定的，且不会有额外行为(例如C#/Java必然会初始化变量)，也不会有如垃圾收集(GC)而造成的不确定性延迟，而且C/C++的数据结构在内存中的布局也是确定的。有时C++的一些功能会使程序性能优于C，当中以内联和模版最为突出，这两项功能使C++标准库的sort()通常比C标准库的qsort()快多倍(C可用宏或人手编码去解决此问题)。另一方面，C/C++能直接映射机器码，之间没有另一层中间语言，因此可以做底层优化，例如使用内部(intrinsic)函数和嵌入汇编语言。然而，许多C++的性能优点并非免费午餐，代价包括较长的编译链接时间和较易出错，因而增加开发时间和成本。

**C++与Visual C++**

C++是一门编程语言，其本身是独立的，不依赖于任何平台、操作系统、工具等。类似的还有C语言、Java语言、Basic语言、Pascal语言等，它们都被称为高级语言。

但是计算机不可能像人一样思考和讲话，它只能处理0、1代码，也就是所谓的机器语言，而作为程序员显然不可能天天用机器语言来编写程序，此时就需要机器语言，而作为程序员显然不可能天天用机器语言来编写程序，此时就需要一个工具来将高级语言所写的代码翻译成机器语言供计算机执行。Visual C++就是这样一个工具，类似的工具还有很多，像C++Builder、Turbo C等。这些工具又不仅仅是一个翻译的工具，而是一个集成的开发环境，提供了漂亮的编程界面、很多支持和帮助，以及丰富的类库供程序员调用。

因此可以说，C++是一门编程语言，而Visual C++则是基于编程语言（C/C++）的开发工具。

**C++程序**

C++程序的开发通常要经过编辑、编译、链接、运行调试这几个步骤。编辑是将源程序输入到计算机中，产生扩展名为.cpp的物理文件。编译是将程序的源代码转换为机器语言代码。但是编译后的程序还不能由计算机执行，还需要进行链接操作。链接是将多个目标文件以及库中的某些文件连在一起，生成一个扩展名为.exe的可执行文件。最后还要对可执行文件进行运行、调试。

C++程序的结构

每个C++程序都由一个或多个文件组成。根据约定，用于存储源代码的文件有两类：头文件和源文件。按照约定的规则，一组对应关联的文件的名字一般是一样的。

❑头文件：它可以包含描述程序所需的数据类型的代码，以及其他类型的声明。这些文件之所以称为头文件，是因为通常在其他源文件的开头包含它们。头文件通常用文件扩展名.h来区分，但这不是强制的，在一些系统中，也使用其他扩展名来标识头文件，例如.hpp。

❑源文件：它的扩展名是.cpp，包含了函数声明，即程序的可执行代码。这些代码通常引用在自己的头文件中定义的数据类型的声明或定义。编译器在编译代码时，需要知道这些声明或定义，因此应在文件的开头通过#include指令指定.cpp文件中需要的.h文件。#include指令是编译器的一个预编译指令，它可以把指定头文件的内容插入代码。还要为代码需要的标准库头文件添加#include指令。

一个完整的程序还可能包含一些资源文件，例如字符串资源、控件资源等。