总结：

1，预备知识：

讲解了c语言的历史和怎样发展到C++，；

还讲解了程序创建的技巧，比如在Microsoft Visual C++中，源代码文件的扩展名可以是cpp、cxx、cc；

除此之外，还介绍了Linux的编译和Windows的编译。

2，开始学习C++：

C++也能在包含头文件stdio.h的情况下使用printf（）等函数；

cout相当于c中的printf，用于显示消息，

插入运算符＜＜表示该语句将这个字符串发送给cout，如 cout＜＜“Come on!”

与printf（）相比，cout无需使用特殊代码%s，%d等；

iostream中的i和o分别表示输入和输出；

在c中，stdio.h中的扩展名h代表头文件，而在c++中则无需扩展名（如iostream）；

using namespace std 表示名称空间，有助于将多个代码组合起来，std就是用来存放类、函数、变量等标准组件；

控制符endl表示重起一行，endl相当于“\n”；

cin＞＞ carrost 表示信息从cin流向carrot；

3，处理数据：

变量命名：名称的第一个字符不能是数字，不能使用C++关键字（int，void，return，double）；

#define和#include一样是预处理器编译指令，作用是定义一个变量的值，与子处理器的全局搜索并替换类似；

int表示整数类型，char表示字符和小整数；

\n：换行符，\t：水平制表符，\v：垂直制表符，\b：退格，\r：回车，\a：振铃；

signed char和unsigned char的区别：前者表示-128到127，后者表示0到255；

const用于修改变量声明和初始化，如

const int Months=12，此时Months的值被固定，不再允许修改该变量的值，

在C++中，应该使用const来定义符号常量而不是#define；

C++中的算术运算符包括+，-，\*，/，%，

%为求模，即取余数；

4，复合类型：

数组：数组声明应该指出储存在每个元素中值的类型，数组名，数组中的元素数；

定义数组时可以使用初始化，但之后就不能使用了，也不能将一个数组赋给另一个数组；

带有双引号的表示字符串，在C++中经常用指针来处理字符串而不是数组；

用cin（）时读取“New York”这类单词时会丢失后半部分，此时可以用cin.getline（）这类面向行输入的函数；

string类变量是一个表示字符串的实体，要使用string类，必须在头文件中包含string：#include＜string＞；

不能将一个数组赋给另一个数组，但可以将一个string对象赋给另一个string对象；

可以使用+将两个string对象合并，还可以使用+=将字符串附加到string对象的末尾；

strcpy（）可以将字符串复制到字符数组中，strcat（）可以将字符串附加到字符数组末尾；

结构是一种比数组更灵活的数据格式，因为同一个结构可以储存多种类型的数据，

structure inflatable（结构声明）；

共同体是一种数据格式，能够储存不同的数据类型，但只能同时储存其中的一种类型，只能储存int，long或double，共用体常用于节省内存；

共用体的最大长度是其最大成员的长度；

面向对象编程与传统的过程性编程的区别在于OPP强调的是在运行阶段（而不是编译阶段）进行决策；

在编译时给数组分配内存称为静态联编，如使用数组声明，在程序运行时选择数组的长度称为动态联编，如使用new（）运算符创建数组时；

整数可以执行加减乘除，而指针描述的是位置，不能将两个地址相乘，指针可以与整数相加；

值为0的指针被称为空指针，常用于表示运算符或函数失败；

char \*pc 中，pc是指针，char \*是指向pc的指针；

vector是new创建动态数组的替代品，提供了自动内存管理功能，必须包含头文件vector；

总结：数组，结构，指针是C++的3种复合类型。

16，string类和标准模板库：

string类为将字符作为对象来处理提供了一种方便的方法。string类提供了自动内存管理功能以及众多处理字符串的方法和函数。例如，这些方法和函数能够合并字符串、将一个字符串插入到另一个字符串中、反转字符串、在字符串中搜索字符或子字符串以及执行输入和输出操作。

诸如auto\_ptr 以及C++11新增的shared\_ptr和unique\_ptr等智能指针模板使得管理由new分配的内存更容易。如果使用这些智能指针来保存new回的地址，则不必在以后使用删除运符。

STL 是一个容器类模板、迭代器类模板、函数对象模板和算法函数模板的集合，它们的设计是一致的，都是基于泛型编程原则的。算法通过使用模板，从而独立于所存储的对象的类型;通过使用迭代器接口，从而独立于容器的类型。迭代器是广义指针。

STL 使用术语“概念”来描述一组要求。例如，正向迭代器的概念包含这样的要求，即正向迭代器能够被解除引用，以便读写，同时能够被递增。概念真正的实现方式被称为概念的“模型”。例如，正向选代器概念可以是常规指针或导航链表的对象。基于其他概念的概念叫作“改进”。

有些算法被表示为容器类方法，但大量算法都被表示为通用的、非成员函数，这是通过将迭代器作为容器和算法之间的接口得以实现的。这种方法的一个优点是:只需一个诸如for each（）或 copy（）这样的函数，而不必为每种容器提供一个版本;另一个优点是:STL 算法可用于非 STL 容器，如常规数组、string 对象容器和算法都是由其提供或需要的迭代器类。For each（）算法使用一个输入迭代器，所有的STL容器类类型都满足其最低要求;而sort（）则要求随机访问迭代器，并非所有的容器类都支持这种迭代器。如果容器类不能满足特定算法的要求，则可能提供一个专用的方法。

17：输入输出和文件：

流是进出程序的字节流。缓冲区是内存中的临时存储区域，是程序与文件或其他I/ O设备之间的桥梁。信息在缓冲区和文件之间传输时，将使用设备(如磁盘驱动器)处理效率最高的尺寸以大块数据的方式进行传输。信息在缓冲区和程序之间传输时，是逐字节传输的，这种方式对于程序中的处理操作更为方便。 C++通过将一个被缓冲流同程序及其输人源相连来处理输入。同样，C++也通过将一个被缓冲流与程序及其输出目标相连来处理输出。iostream 和 fstream文件构成了I/O类库，该类库定义了大量用于管理流的类。包含了iostream文件的C++程序将自动打开8个流，并使用8个对象管理它们。cin对象管理标准输人流，后者默认与标准输人设备(通常为键盘)相连;cout对象管理标准输出流，后者默认与标准输出设备(通常为显示器)相连;cerr 和clog 对象管理与标准错误设备(通常为显示器)相连的未被缓冲的流和被缓冲的流。这4个对象有都有用于宽字符的副本，它们是wein、wcout、wcerr 和 wclog。

I/O类库提供了大量有用的方法。istream 类定义了多个版本的抽取运算符(>>)，用于识别所有基本的 C++类型，并将字符输人转换为这些类型。get()方法族和 getline()方法为单字符输入和字符串输入提供了进一步的支持。同样，ostream 类定义了多个版本的插人运算符(<<)，用于识别所有的 C++基本类型，并将它们转换为相应的字符输出。put()方法对单字符输出提供了进一步的支持。wistream 和 wostream 类对宽字符提供了类似的支持。

使用 ios\_base 类方法以及文件iostream 和iomanip中定义的控制符(可与插人运算符拼接的函数)，可以控制程序如何格式化输出。这些方法和控制符使得能够控制计数系统、字段宽度、小数位数、显示浮点变量时采用的计数系统以及其他元素。

fstream文件提供了将 iostream方法扩展到文件I/O的类定义。ifstream 类是从 istream 类派生而来的。通过将ifstream 对象与文件关联起来，可以使用所有的istream方法来读取文件。同样，通过将 ofstream 对象与文件关联起来，可以使用ostream 方法来写文件;通过将fstream 对象与文件关联起来，可以将输人和输出方法用于文件。

要将文件与流关联起来，可以在初始化文件流对象时提供文件名，也可以先创建一个文件流对象，然后用open()方法将这个流与文件关联起来。close()方法终止流与文件之间的连接。类构造函数和open()方法接受可选的第二个参数，该参数提供文件模式。文件模式决定文件是否被读和/或写、打开文件以便写人时是否截短文件、试图打开不存在的文件时是否会导致错误、是使用二进制模式还是文本模式等。