第一章

关键点：一前两行改为 #include<iostream>

using namespace std;

(预处理命令不加 .h ，并在下面加一句using namespace std

其意为使用命名空间 std)

二cout为输出，cin为输入，使用方式为

cout<<”\*\*\*”(<<\*)<<endl //endl意为换行//

cin>>\*(>>\*......)

三.包含类

先声明类，再分为私有与公有两部分，公用的部分可以被类与非本类语句调用，私有则仅能被本类语句引用，表现了封装特点以保证隐密性，公有一般为函数，私有一般为语句

class name

{private //私用//

\*\*\*

Public //公用//

\*\*\*

}

第二章

关键点：一：数据类型多出了共用体类型（union）和类类型（class）

二：int或char前面可以加signed（表示有符号）或unsigned（表示无符号）

三：用不到long double类型（double与long double分配字节相同）。

四：十进制整数后加L表示long int型常量，八进制整数前加0表示八进制数，前加0X表示此为16进制数

五：在十进制小数后加F或f表示单精度浮点型，加L表示双精度浮点型

六：cout<<’a’可以表示输出a

七：/\*可以表示各种转义字符，详见p22，’/数字’表示对应ASCII的字符

八：符号常量是在全局声明中给一个符号赋予一个字符，例如

#define price 30。（一个数据用到多次时，只在全局声明处修改即可，达到便利的效果）

九：定义变量只要在使用前定义就行

十：定义变量时，加上const，变量的值在运行时不能改变，称为常变量（define定义的符号没有类型，而const只是不可改变仍具有变量的特征）

十一：不同类型数据运算时系统自行进行类型转换

十二：强制转换表达方式为：（类型名）（表达式）

十三：赋值过程中两侧类型不一致时，系统自动进行转换（必须都是数值型或字符型）

十四：当负值赋给其他变量时，会因为按储存单元中的储存形式直接传送的原因导致出现意外的结果

十五：逗号表达式从左到右求解，值为最右侧表达式的值（多数情况下使用逗号表达式是为了分别得到各个表达式的值）

第三章

关键点：一：数据结构即对数据的描述（在程序中要指定数据的类型和数据的组织形式），操作步骤叫做算法，

程序=算法+数据结构

二：算法的表示可以用自然语言、流程图、计算机语言或伪代码

三：对变量或其他对象的定义也是一条语句

四：“流”指的是数据流，<<是流插入运算符，>>是流提取运算符

五：不能用一个<<插入多个输出项如<<a,b提取运算符一样

六：标准输入流与输出流有控制符，控制符在p50

七：cout可以输出getchar函数得到字符的ASCII的值

八：bool用来定义逻辑型变量，只能为false（0）或者ture（1）

第四章

关键点：一：函数调用注意事项：函数调用的实参值许多c++系统是按从右至左的顺序求值的（如i=3，func（i,++i）括号内并非3，4而是4，4）

二：函数定义注意事项：c++不允许嵌套定义函数（即定义的函数里有另一个函数）

三：内置函数：在函数首行左端加关键字inline即为内置函数（即在编译时把所调用函数的代码嵌入主函数中，可以加快程序运行效率，详细原因p98）

注意：一般只将规模很小的而使用频繁的函数声明为内置函数，内置函数中不可有复杂控制语句如循环与switch

四：函数的重载：c++允许同一定义多个函数但这些函数参数个数与参数类型不能相同（将功能相似的函数用同一个函数名命名但参数个数与类型不同，可以达到方便效果）

五：函数模版：当函数类型与参数类型不确定时可使用函数模版，在全局区template<typename（或class） T>后在声明函数时将函数类型与参数类型换位T即可如 T func(T a,T b)

六：有默认参数的函数：在函数声明时可以直接给参数赋值，若不想使用可以在传值时可以进行覆盖（注意事项：当声明与定义时都给参数赋值，那么默认是函数声明处，部分系统会报错）

七：在函数内定义的叫局部变量，函数外定义的叫全局变量，也叫外部变量

八：变量储存方式：内存中供用户储存的空间可分为三部分，程序区，静态储存区和动态储存区

程序中的数据存在静态储存区和动态储存区里

全局变量储存在静态储存区中，程序开始时为其分配储存空间，程序执行完毕就会将这些空间释放

动态储存区存放数据有，函数形式参数（调用函数时给形参分配储存空间），函数定义中的变量，函数调用时的现场保护与返回地址

九：c++中变量有储存类别的属性，有四种，为自动的，静态的，寄存器的，外部的

十：？？自动变量：函数中的局部变量，如果不用关键字static加以声明，编译系统是对他们动态的分配空间的，这类局部变量称为自动变量

十一：用static声明静态局部变量：

定义变量时用static关键字后可以在函数结束调用后该变量不消失原值，如static int c

十二：用register声明寄存器变量，是在改变量使用频繁的情况下，可以加大执行效率，如register int a

十三：用extern声明外部变量，如果不在全局区声明某变量，声明变量时加上关键字extern可以将作用域拓展到文件尾处。

当有多文件时，加extern可以引用别的文件的变量，如

File1.cpp file2.cpp

extern int a; int a=3;

则文件1就引用了文件2中的a

十四：用static声明静态外部变量，若在程序设计时想要某些外部变量仅仅被本文本使用，在定义外部变量时加关键字static即可（主要是为了防extern）

这些关键字很复杂，p116上有汇总