第五章：

1. C++允许使用多维数组，如int a[2][3][4]
2. 如果对数组中全部元素都在定义时赋值，则定义数组时对第一维的长度可以不指定，但第二维长度不能省，如 int a[2][2]={1,2,3,4}等价于int a[][2]={1,2,3,4}
3. 字符串数组是有长度的，输入字符串长度过多时系统不会报错，但是会将多出字符存在数组所在内存的后面，可能会破坏其它数据造成严重后果
4. C++提供新的数据类型，字符串类型（string类型）。字符串类与其它数据类型相同，需要先定义后使用，而且字符串类的数据长度会根据需要自行改变，无需顾虑会超长。如 注 注：string不是c++本身的基本类型，而是在c++标准库中声明的一个类

string a=”super”;

cin>>a;

1. 字符串变量的运算：

字符串复制直接用赋值号 string1=string2

字符串连接用加号 string1=string1+string2

字符串比较用>,>=,<,<=,!等

6.string类型可以有数组称为字符串数组

第六章：

1. 一个变量的地址称为该变量的指针，有一个变量是用来存放地址的，则称为指针变量
2. C++编译系统一律将形参数组名作为指针变量处理
3. C++有函数指针，定义方式为 函数类型 （\*变量名）（形参表） 1

Int max(int x,in y);

Int (\*p)(int ,int);p=max;

1. 返回指针值的函数称为指针函数，类型名 \*函数名（参数列表）
2. 指针数组int \*p[4]为四个值，但不可写为int (\*p)[4],此为指向一位数组的指针
3. const int \*p=&a，不能通过P改变a的值，但可以对a直接操作改变值，也可以改变p的指向，此时p成为指向常变量的指针变量
4. int \*const p=a,此时指针不能改变指向，但可以通过指针指改变变量的值，称为常指针
5. 将以上两种叠加在一起为指向常量的常指针变量，即指针的指向不能改变，也不能通过指针改变其值const 基本类型名 \*const指针变量名
6. 可以把非void型指针赋给void型指针变量，但不能把void指针直接赋给非void型指针变量，必须先进行强制转换
7. 对一个数据可以建立一个引用，它的作用是为一个变量起一个别名

Int a; int&b=a;

注意：1.引用不是一种独立的数据类型，对引用只有声明，没有定义

1. 声明一个引用时，必须同时初始化，即声明它代表哪一个变量
2. 在声明一个引用后，不能再使之作为另一变量的引用
3. 不能建立引用数组，也不能引用数组
4. 不能建立引用的引用
5. 不能建立指向引用的指针
6. 可以取引用的地址

第七章：

Struct 结构体名

{成员表

}变量名

1. 声明结构体类型的一般方式为
2. 结构体类型作为函数参数时，如果使用结构体变量名做参 数，将会在时间和空间上开销很大，使用指针则会不便阅读，使用结构体变量的引用可以避免以上两种劣势
3. C++一般用new和delete进行内存分配和撤销储存空间，new和delete是运算符，不是函数，因为执行效率高
4. C++在声明枚举时保留了c的用法，但是在定义枚举类型名的时候允许不写enum