第二章知识总结：

C++与 C 语言的区别，对其的改进和扩展：

一、函数中基本控制的区别

1. I/O 流实现输入输出

（1）cin 实现输入，cin>>变量 1>>变量 2>>

（2）cout 实现输出，cout<<表达式 1<<表达式 2<<endl(换行符，相当于”\n”的作用)

（3）cin 和 cout 都包含在头文件#include <stdlib.h>中

2. 新增的单行注释

/**/和//

3. const 定义常量

Const 与 define 的区别在于 const 定义常量使同时确定了常量的类型

Const[常量类型]符号常量名=表达式；//有分号

Const 结合指针 离谁近保护谁的值不被改变

4. 新增强制类型转换

对原表达式进行圆括号，不对所需转换的类型圆括号

5. Bool

True-真-1

False-假-0

cout<<f<<boolalpha<<f<<noboolalpha<<f<<endl;

1 true 1;

6. 名字空间

Namespace 名字空间名称{....;}

（1）先用 using namespace 名字空间名称声明整个名字空间

（2）名字空间名称:: 局部内容名“::”叫域解析符，可以解决命名冲突问题

二、Using namespace 名字空间名称:: 函数的区别

1. 局部变量的用法、定义方法：随用随定义

（3）域解析符:: 作用：扩大全局变量的可见范围局部内容名和全局内容名冲突时可以通过这个引用全局变量。

2. 形式参数使用的不同：形参在声明时可以带有默认值。

3. 内联函数使用：inline 代替宏定义可以增加安全性

内联函数不可以使用循环语句和开关语句。

4. 函数重载

三、新增：引用及其运用

数据类型 &引用名=一个已知类型的变量名;

（1）引用名要合法

（2）对引用名必须指定是那个变量的，即对其初始化

（3）引用名只是变量的别名，不存在各自的内存空间

（4）不能建立：a:void 类型的引用名

b:引用的引用名

c:指向引用的指针

d;引用数组

1. 引用作为形式参数：扩大了实参的作用域

2. 引用与指针的区别

3. 引用作为返回值使用：类型名&函数名(形式参数表)

四、动态内存空间管理

1.用 new 函数申请动态内存空间

New 函数也存在于头文件#include<stdlib.h>中

new 数据类型名;

new 数据类型名[整型常量];

2.用 delete 释放动态内存空间

Delete 函数也存在于头文件#include<stdlib.h>中

delete[ ]指针变量名;

delete 指针变量名;

3.Void 类型的指针指向所有类型变量

五、C++中的异常处理

1.异常及其处理

可以减轻底层函数负担

2.异常处理的实现

（1）检查异常：try

（2）抛出异常：throw

（3）捕捉异常：catch