Copyright ©2021-2099 Aiting Chen. All rights reserved

**第二章课后作业**

2.1 一是全面兼容C，并对C的功能作了不少扩充；二是增加了面向对象的机制，具体表现为抽象数据类型、封装与信息隐蔽、以继承方式实现程序的重用、以函数重载、运算符重载和虚函数来实现多态性等等。

2.4 没有进行函数原型声明。程序有一个自定义函数sum，该程序首先编写了主函数其次才是自定义函数，但是在主函数中有自定义函数的调用，而此时在主函数前面并没有sum的函数原型声明，导致编译错误。改正的方法有两种，一是将sum整个函数提到主函数前，二是在主函数前加上sum的函数原型声明。

2.5 （1）等价，函数原型可省略形参而只保留类型

（2）不等价，函数定义时不能省略形参

2.10 函数的重载需要函数参数的类型不同、函数参数个数的不同或者二者兼有才能进行重载。重载函数名字相同，只有返回类型不同不能重载。

2.11 内联函数适用于函数代码小且频繁调用的函数

2.12 内联函数不能含有复杂的语句，例如for语句、switch语句等。

2.13 C++允许设置默认值，设置了默认值的参数必须放在未设置默认值参数的右边。在函数的声明和定义都可以设置默认值，当已经在函数的声明中设置了默认值时，函数的定义可以不表明默认值。默认值不必全都设置。

2.15 同2.10重载函数的知识点。

2.18 程序输出的是缀有作用域运算符的i，即全局变量i，全局变量i在主函数中被赋值为局部变量i+1，局部变量为100，所以输出结果为101.

2.19 f函数的形参中第一个形参为引用，可以修改对应实参的值；第二个形参为普通变量，其的改变不能改变对应实参的值，所以输出结果为：10 10.

2.21 例子：

#include<iostream>

using namespace std;

int main()

{ int a=1;

#define T1 a+a

#define T2 T1-T1

cout<<”T2 is “<<T2<<endl;

return 0; }

在这个程序中，输出结果为：T2 is 2，但这显然不符合我们的预期，原因是C++将T1-T1解释为a+a-a+a，而实际上应该是（a+a）-(a+a),而用const改写程序则不会出现这种错误，原因是const修饰符定义常量，常量一旦被定义，程序中任何地方都不能更改。

#include<iostream>

using namespace std;

int main()

{ int a=1;

int const T1 a+a

int const T2 T1-T1

cout<<”T2 is “<<T2<<endl;

return 0; }

此时的输出结果：T2 is 0，符合期望。