1. 复习与导入：
2. 用户自定义函数
3. 系统函数
4. 定义函数与调用函数
5. 函数的形参与实参
6. 函数的返回值
7. 有返回值的函数
8. 无返回值的函数
9. 函数的避暑胜地明
10. 局部变量和全局变量
11. 教学内容：
12. 类的声明和组成

1、类的概念：

类比基本数据类型功能强大，一般来讲，类是数据和函数的封装体

2、类的声明：

class 类名

{

public:

公用的数据成员和成员函数;

private:

私有的数据成员和成员函数;

protected:

受保护的数据成员和成员函数;

};

被public声明为公用的数据成员和成员函数可以被本类中的成员函数所引用，也可以被类的作用域内的其他函数引用。

被private声明为私有的成员，只能被本类中的成员函数引用，类外不能调用（友元除外）

被protected声明为受保护的成员，不能被类外访问，但可以被派生类的成员函数访问，保护成员会在继承中使用。

例如：定义一个关于长方形的类rect，类中除了包括长x和宽y两个数据外，还包括对长和宽的设置、求周长及面积的操作。

class rect

{

private:

float x,y;

public:

void set(float a,float b);

float perimeter();

float area();

};

3、成员函数的声明和实现

在类体外定义成员函数的结构如下：

类型 类名：：成员 函数名（参数列表）

｛

成员函数体

｝

在类型和成员函数名之间加了“类名：：”。“：：”

class Time

{

public:

void set\_time(int h,int m,int s);

void show\_time()

{

cout<<hour<<":"<<minute<<":"<<sec<<endl;

}

private:

int hour;

int minute;

int sec;

};

void Time::set\_time(int h,int m,int s)

{

hour=h;

minute=m;

sec=s;

}

1. 对象的定义和对象成员引用
2. 定义对象

定义对象的一般形式：

类名 对象名;

如定义一个Time类对象，对象名为mytime

Time mytime;

class Student

{

public:

void display()

{

cout<<"num:"<<num<<endl;

cout<<"name:"<<name<<endl;

cout<<"sex:"<<sex<<endl;

}

private:

int num;

char name[20];

char sex;

}stud1,stud2;

Student stu3,stu4;

1. 对象成员的引用

对象名.成员名;

例如:stud1.num=1001;

1. 构造函数和析构函数

要创建一个对象，一般要将对象中的数据成员进行初始化和为对象申请必要的存储空间。但不能在类定义中对数据成员初始化，因为类并不是一个实体，而是一个抽象类型，并不占存储空间，也就无法存储数据。

例如，下面的写法是错误的

class Time

{

int x=1;

int y=2;

}

1. 构造函数

构造函数的作用就是在创建对象时利用特定的值构造对象，将对象初始化为一个特定的状态，使此对象具有区别于其他对象的特征。构造函数在对象创建的时候由系统自动调用。

构造函数也是类的一个成员函数，除了具有一般成员函数的特征之外，还有自己特有的性质。构造函数的函数名与类名相同，而且不能有任何返回类型，也不能标识为void类型。构造函数一般声明为公有函数。构造函数是在定义对象时由C++自动调用。

例：

#include<iostream>

using namespace std;

class Time

{

private:

int hour;

int minute;

int sec;

public:

Time()

{

hour=0;

minute=0;

sec=0;

}

void set\_time();

void show\_time();

};

void Time::set\_time()

{

cin>>hour;

cin>>minute;

cin>>sec;

}

void Time::show\_time()

{

cout<<hour<<":"<<minute<<":"<<sec<<endl;

}

void main()

{

Time t1;

t1.set\_time();

t1.show\_time();

Time t2;

t2.set\_time();

t2.show\_time();

}

使用构造函数时应注意如下问题上：

1. 构造函数在对象被创建的时候自动调用
2. 构造函数没有返回值，因此也不需要在定义构造函数时声明类型，它的作用主要是进行初始化。
3. 构造函数不需要用户调用，也不能被用户调用。在建立对象时自动执行一次，而且只执行一次
4. 如果用户没定义构造函数，系统会自己生成一个，但函数是空的。
5. 带参数的构造方法

应用构造函数定义对象的一般形式为：

类名 对象名(构造函数的参数表);

#include<iostream>

using namespace std;

class Box

{

public:

Box(int,int,int);

int volume();

private:

int height;

int width;

int length;

};

Box::Box(int h,int w,int len)

{

height=h;

width=w;

length=len;

}

int Box::volume()

{

return height\*width\*length;

}

void main()

{

Box box1(12,25,30);

cout<<"The volume of box1 is"<<box1.volume()<<endl;

Box box2(15,30,21);

cout<<"The volume of box2 is"<<box2.volume()<<endl;

}