



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY

# C++程序设计

## Programming in C++



1011018

主讲：魏英，计算机学院

# 对象的定义和使用

- ◆ 1、对象的定义.....
- ◆ 2、对象的动态建立和释放.....

- ▶ 定义一个类时，也就是定义了一个具体的数据类型。若要使用类，需要将类实例化，即定义该类的对象。
- ▶ 需要注意，我们之前也使用了“对象”一词，那里主要是指数据对象。
- ▶ 从现在起，“对象”一词专门表示类的实体。

## 26.1 对象的定义

- ▶ 1. 先定义类类型再定义对象
- ▶ 有两种定义对象的形式：
- ▶ ①将类的名字直接用作类型名：

**类名 对象名列表;**

- ▶ ②指定关键字class或struct，后面跟着类的名字：

**class 类名 对象名列表;**

**或**

**struct 类名 对象名列表;**

- ▶ 对象名列表是一个或多个对象的序列，各对象之间用逗号（，）分隔，最后必须用分号（；）结束，对象取名必须遵循标识符的命名规则。例如：

```
Point a,b; //C++特色定义对象  
class Point x,y; //兼容C语言特色定义对象
```

## 26.1 对象的定义

- ▶ 2. 定义类类型的同时定义对象
- ▶ 一般形式为：

```
class 类名 { //类体  
    成员列表  
} 对象名列表;
```

- ▶ 例如：

```
class Point { //类体  
public:    ...//公有的数据成员和成员函数  
private: ...//私有的数据成员和成员函数  
} one , two; //对象列表
```

## 26.1 对象的定义

- ▶ 3. 直接定义对象
- ▶ 一般形式为：

```
class { //类体  
    成员列表  
} 对象名列表;
```

- ▶ 例如：

```
class { //无类名类体  
public:    ...//公有的数据成员和成员函数  
private:  ...//私有的数据成员和成员函数  
} p1 , p2; //对象列表
```

- ▶ 一般而言，定义类型时不进行存储空间的分配。
- ▶ 当定义一个对象时，将为其分配存储空间。



- ▶ 有时人们希望在需要用到对象时才创建（create）对象，在不需要用该对象时就撤销（destroy）它，释放其所占的存储空间，从而提高存储空间的利用率。
- ▶ 利用new运算符可以动态地分配对象空间，delete运算符释放对象空间。

- ▶ 动态分配对象的一般形式为：

```
类名 * 对象指针变量;  
对象指针变量 = new 类名;
```

- ▶ 例如：

```
Point *p; //定义指向Point对象的指针变量  
p = new Point; //动态分配Point对象
```

- ▶ 用new运算动态分配得到的对象是无名的，它返回一个指向新对象的指针的值，即分配得到是对象的内存单元的起始地址。程序通过这个地址可以间接访问这个对象，因此需要定义一个指向类的对象的指针变量来存放该地址。**显然，用new建立的动态对象是通过指针来引用的。**
- ▶ 在执行new运算时，如果内存不足，无法开辟所需的内存空间，C++编译器会返回一个0值指针。因此，只要检测返回值是否为0，就可以判断动态分配对象是否成功，只有指针有效时才能使用对象指针。

- ▶ 当不再需要使用由new建立的动态对象时，必须用delete运算符予以撤销。例如：

```
delete p; //撤销p所指向的Point对象
```

- ▶ 释放了p所指向的对象。此后程序不能再使用该对象。
- ▶ 请注意，new建立的动态对象不会自动被撤销，即使程序运行结束也是如此，必须人为使用delete撤销。

## 26.2 对象的动态建立和释放

### 【例26.1】

```
1  #include<iostream>
2  using namespace std;
3  class Box
4  { public:
5      int width, length, height;
6  };
7  int main()
8  {
9      Box * p=new Box;
10     p->width=10; p->length=20; p->height=30;
11     cout<<p->width<<"\t"<<p->length<<"\t"<<p->height<<endl;
12     delete p;
13     return 0;
14 }
```

# CP 程序设计