CSDN 学院

C++语言基础

迁者 - 贺利坚

http://blog.csdn.net/sxhelijian/

http://edu.csdn.net

本节主题：

派生类的构造函数和析构函数

派生类的构造函数和析构函数

- 构造函数的主要作用是对数据成员初始化
- 析构函数在释放对象前做一些相关的处理
- 背景
  - 派生类还继承了基类的数据成员
- 问题
  - 设计派生类的构造函数时，不仅要考虑派生类所增加的数据成员的初始化，还应当考虑基类的数据成员初始化
- 解决办法
  - 在执行派生类的构造函数时，调用基类的构造函数

一个简单派生类的定义

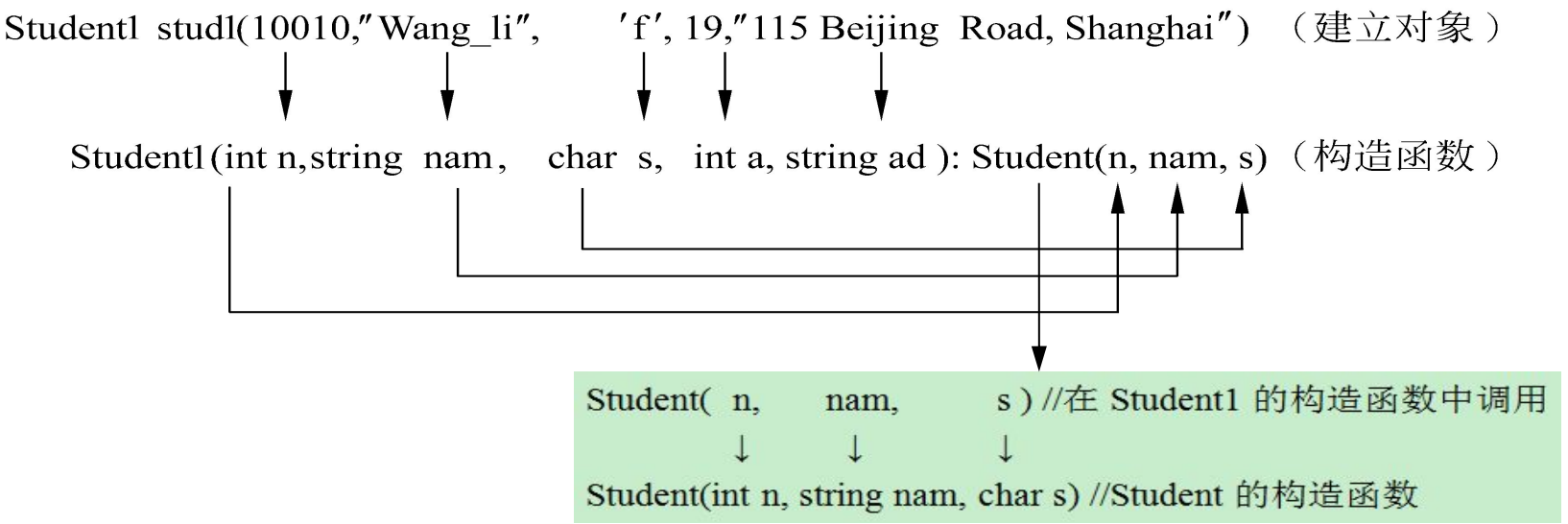
```
class Student {
public:
   Student(int n,string nam,char s):num(n),name(nam),sex(s) {}
   ~Student( ) { }
   void show( ) {
      cout<<"num: "<<num<<endl;
      cout<<"name: "<<name<<endl;
      cout<<"sex: "<<sex<<endl<<endl;
   }
protected:
   int num;
   string name;
   char sex ;
};
```

```
int main( )
{
   Student1 stud1(10010,"Wang",'f',19,"Beijing Road");
    ...
}
```

```
class Student1: public Student {
public:
   Student1(int n,string nam,char s,int a,string ad):Student(n,nam,s),age(a),addr(ad) {}
   void show1( ){
      show();
      cout<<"age: "<<age<<endl;
      cout<<"address: "<<addr<<endl<<endl;
   }
   ~Student1( ) { }
private:
   int age;
   string addr;
};
```

派生类构造函数一般形式

```
派生类构造函数名(总形式参数表列): 基类构造函数名(实际参数表列)
{
    派生类中新增数据成员初始化语句
}
```



构造函数的写法

- 在类内定义派生类构造函数

  Student1(int n,string nam,char s,int a,string ad):Student(n,nam,s),age(a),addr(ad) {}

- 在类的外面定义派生类构造函数
  - 类体中只写构造函数的声明:

  Student1(int n, string nam, char s, int a, string ad);

  - 在类的外面定义派生类构造函数

  Student1::Student1(int n, string nam, char s, int a, string ad)

  :Student(n, nam, s),age(a), addr(ad){}

- 不使用初始化表

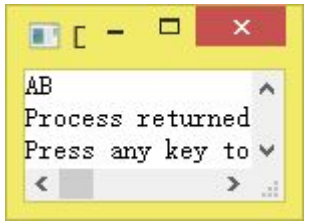
  Student1(int n, string nam, char s,int a, string ad): Student(n, nam, s) {age=a;addr=ad;}

构造函数和析构函数执行的顺序

- 建立派生类对象时，执行构造函数的顺序:
  - 派生类构造函数先调用基类构造函数；
  - 再执行派生类构造函数本身(即派生类构造函数的函数体)。

- 在派生类对象释放时，
  - 先执行派生类析构函数~Derived( )
  - 再执行其基类析构函数~Base( )

```
class Base
{
protected:
   Base(){cout<<'A';}
};
class Derived: public Base
{
public:
   Derived(char c){cout<<c;}
};
```




```
int main()
{
   Derived d1('B');
   return 0;
}
```

有子对象的派生类的构造函数

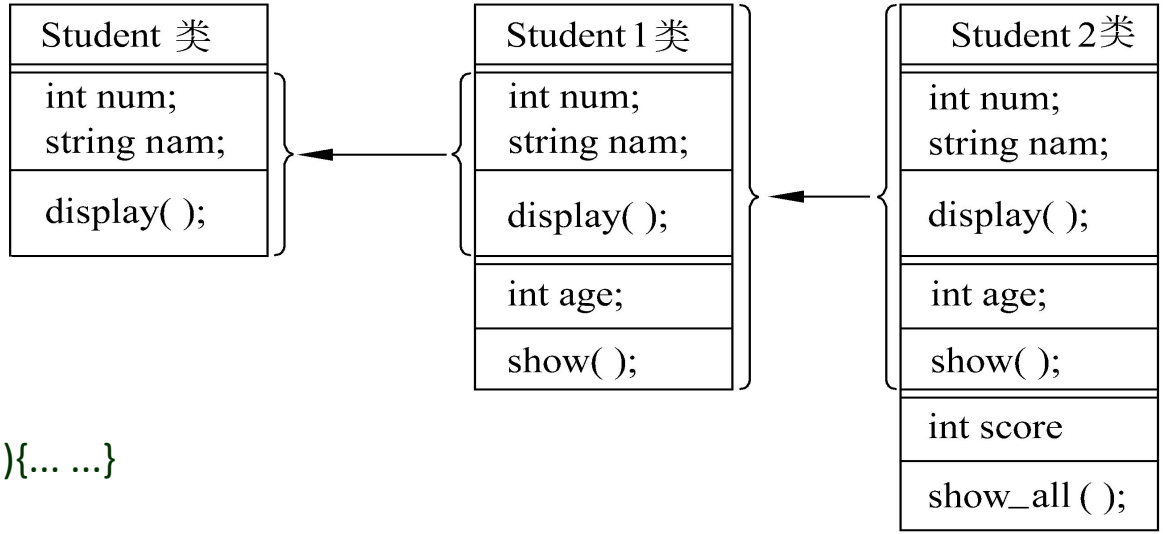
- 子对象(subobject)，即对象中的对象，类的数据成员是另一个类的对象

```
class Student1: public Student{
public:
    Student1(int n, string nam,int n1, string nam1,int a, string ad):
        Student(n,nam), monitor(n1,nam1),age(a),addr(ad) {} ……
private:
  Student monitor;
  int age;
  string addr;
};
```

执行构造函数的顺序:
① 基类的构造函数；
② 子对象构造函数；
③ 其他数据成员初始化。

```
派生类构造函数名(总形式参数表列)
    : 基类构造函数名(实际参数表列),子对象名(实际参数表列)
{
    派生类中新增数据成员初始化语句
}
```

多层派生时的构造函数



Student(int n, string nam){... ...}

Student1(int n, string nam,int a): Student(n,nam){... ...}

Student2(int n, string nam, int a, int s): Student1(n,nam,a){... ...}

派生类构造函数的特殊形式

- 当不需要对派生类新增的成员进行任何初始化操作时，派生类构造函数的函数体可以为空。
  - 这种派生类构造函数的作用只是为了将参数传递给基类构造函数和子对象，并在执行派生类构造函数时调用基类构造函数和子对象构造函数。
- 基类中没有定义构造函数，或定义了没有参数的构造函数
  - 在派生类构造函数中可不写调用基类构造函数的语句，调用派生类构造函数时系统会自动调用基类的默认构造函数。
- 基类或子对象类型的声明中定义了带参数的构造函数
  - 必须显式地定义派生类构造函数，并在派生类构造函数中写出基类或子对象类型的构造函数及其参数表。
- 基类中既定义无参数的构造函数，又重载了有参数的构造函数
  - 派生类构造函数中可以调用带参数的基类构造函数，也可以不调用基类的构造函数。

CSDN 学院

THANKS

本课程由 迁者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：http://ems.csdn.net/
人才服务：http://job.csdn.net/
CTO俱乐部：http://cto.csdn.net/
高校俱乐部：http://student.csdn.net/
程序员杂志：http://programmer.csdn.net/

CODE平台：https://code.csdn.net/
项目外包：http://www.csto.com/
CSDN博客：http://blog.csdn.net/
CSDN论坛：http://bbs.csdn.net/
CSDN下载：http://download.csdn.net/