嵌入式C++开发

常见知识点

整理by Yang，Gavin

2016-12-12

# 1、哪种情况必须用到初始化成员列表

必须用到初始化成员列表的四种情况：

1）初始化一个reference成员；  
2）初始化一个const成员；  
3）调用一个基类的[构造函数](https://www.baidu.com/s?wd=%E6%9E%84%E9%80%A0%E5%87%BD%E6%95%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkmyRsnyNWnHfsuhcznW7h0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHbdrHbdrHbsrj0LPHf3rH0zn0" \t "_blank)，而该函数有一组参数；  
4）调用一个数据成员对象的[构造函数](https://www.baidu.com/s?wd=%E6%9E%84%E9%80%A0%E5%87%BD%E6%95%B0&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YkmyRsnyNWnHfsuhcznW7h0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnHbdrHbdrHbsrj0LPHf3rH0zn0" \t "_blank)，而该函数有一组参数。

举例说明：

|  |  |
| --- | --- |
|  | #include <iostream>  using namespace std;    class Base  {  public:      Base(const string &str = "", int i = 0) : Bstr(str), \_i(i) // 使用const引用避免复制,      // 如果只使用const则无法使用字面常量"DerivedStr"为str赋值      {          cout << "Base Constructor" << " Bstr = " << Bstr << ", \_i = " << \_i << endl;      }        string Bstr;      int \_i;  };  class Derived : public Base  {  public:        // 调用基类构造函数，而它拥有一组参数时，要使用成员初始化列表      Derived() : Base("DerivedStr", 200)  // 这个是正确的      {          //Base::Bstr = "DerivedStr"; // 基类构造函数再次之前调用，这里赋值没有用。          //Base::\_i = 200;          cout << "Derived Constructor" << endl;      }        string Dstr;  };  int main()  {      Derived d;      return 0;  } |

# 2、说一下C++的虚函数

通过虚函数实现了C++程序在运行时的多态。每个带有虚函数的类对象都有一个指向虚函数表的指针。这个指针位于类对象内存空间的顶部。父类的虚函数被子类重写后，在虚函数表中，子类的虚函数地址会覆盖父类的这个虚函数地址。当在使用父类指针去执行这个虚函数时，就会执行子类的虚函数。这就实现了多态的性质。**虚函数表是存放在静态存储区里面**。

# 3、什么时候应必须使用dynamic\_cast？什么时候dynamic\_cast可以使用[static\_cast](http://baike.baidu.com/view/1745207.htm)代替？

1、对应单继承，如果父类的指针指向子类对象，用dynamic\_cast和static\_cast效果一样；

2、如果父类指针不是真的执行子类对象，用dynamic\_cast则返回NULL，能够更早的禁止error的发生。如果用static\_cast虽然返回的不是NULL，但在运行时可能抛出异常。

3、对应多重继承，如果父类Base指针真的指向子类对象，使用dynamic\_cast和static\_cast都可以转换为子类对象指针。但如果要转化为父类的兄弟类Base1，必须使用dynamic\_cast,使用static\_cast不能编译。

4、对应不是真的指向子类对象，要想转化为子类指针或兄弟类对象指针，用dynamic\_cast则返回NULL。如果用static\_cast虽然返回的不是NULL，但在运行时可能抛出异常。

# 4、qobject\_cast与dynamic区别

T qobject\_cast(QObject \*obj)

类T必须继承自QObject类，并且使用Q\_OBJECT对象。qobject\_cast类似于dynamic\_cast，但它不需要RTTI的支持。qobject\_cast依赖QMetaObject信息来判断。

# 5、static\_cast, dynamic\_cast, const\_cast, reinterpret\_cast

dynamic\_cast: 通常在基类和派生类之间转换时使用,run-time cast  
const\_cast: 主要针对const和volatile的转换.  
static\_cast: 一般的转换，no run-time check.通常，如果你不知道该用哪个，就用这个。  
reinterpret\_cast: 用于进行没有任何关联之间的转换，比如一个字符指针转换为一个整形数。