

# 实验：研究C++的对象模型

李雨轩

计算机2205

2204112913

## 一、实验要求

1. 定义一个类，其中有静态数据成员、各种类型非静态数据成员（含字符指针），甚至包括引用（可选），静态和非静态成员函数（含分配空间的构造函数、析构函数）。
2. 定义全局对象、main函数中局部对象、另一个被main调用的外部函数func中定义局部对象（可以是形参）、main函数中动态创建对象，每种对象至少2个。观察、分析各种对象地址。
3. 输出对象中各个静态与非静态数据成员的值、地址、对象的存储空间大小等信息。由此理解对象的本质、静态数据成员是本类对象共享一份拷贝等问题。此外，应观察对齐现象。
4. （可选）输出对象的每个字节，以揭示引用的实现方法。
5. 对于上述各种对象，输出静态、非静态成员函数地址，以及main、func等外部函数的地址，并分析。要求采用合理方法，避免编译器提出警告。

*注意：本题作为实验报告内容，要求有代码、注释、结果截图及分析。以班为单位统一收，电子版，发我的邮箱 [libaohong32@163.com](mailto:libaohong32@163.com)*

## 二、问题分析与实验过程

为了研究C++的对象模型，我设计了下面用于研究C++对象模型的程序，它展示了类的构造、拷贝构造、析构函数、静态成员函数、非静态成员函数以及对象的动态分配和释放等方面的特性。（源代码见附页）

### 1. 定义一个类：

定义了一个名为 MyClass 的类，该类包含以下内容：

- 静态数据成员 staticData
- 非静态数据成员：

- 整型数据 integerData
- 双精度浮点型数据 doubleData
- 字符指针 charPtrData
- 字符串类型数据 stringData
- 引用数据成员 refData
- 静态成员函数 staticFunction()
- 非静态成员函数 nonStaticFunction()
- 构造函数和析构函数，构造函数需要动态分配内存来初始化字符指针 charPtrData。

```
class MyClass {
public:
    static int staticData;
    int integerData;
    double doubleData;
    char* charPtrData;
    std::string stringData;
    int& refData;

public:
    MyClass(int intValue, double doubleValue, const char* charValue, const std::string&
stringValue, int& ref);

    MyClass(const MyClass& obj)
        : integerData(obj.integerData), doubleData(obj.doubleData), stringData(obj.stringData),
refData(obj.refData);

    ~MyClass();

    static void staticFunction();

    void nonStaticFunction();
};
```

## 2. 创建各种类型的对象:

- 全局对象 globalObj1 和 globalObj2
- main 函数中的局部对象 localObj1 和 localObj2
- 外部函数 func 中的局部对象 funcObj
- main 函数中动态创建的对象 dynamicObj1 和 dynamicObj2

## 3. 输出对象信息:

- 输出各个对象中静态和非静态数据成员的值、地址以及对象的存储空间大小。
- 观察静态数据成员的共享情况，即不同对象的静态数据成员是否共享同一份拷贝。
- 观察对象的存储空间布局，并分析可能出现的对齐现象。

下面是相应的代码和输出：

```
// 在类外初始化静态数据成员
int MyClass::staticData = -1010;

void func(MyClass obj) {
    // 外部函数中的局部对象
    obj.nonStaticFunction();
}

// 全局对象
MyClass globalObj1(1, 2.3, "Global 1", "Object", MyClass::staticData);
MyClass globalObj2(2, 3.7, "Global 2", "Object", MyClass::staticData);

int main() {

    // main 函数中的局部对象
    MyClass localObj1(3, 4.9, "Local 1", "Object", MyClass::staticData);
    MyClass localObj2(4, 5.1, "Local 2", "Object", MyClass::staticData);

    // 调用外部函数 func(), 传递局部对象
    funcObj(5, 6.3, "Func Object", "Object", MyClass::staticData);

    // 动态创建对象
    MyClass* dynamicObj1 = new MyClass(6, 7.2, "Dynamic1", "Object", MyClass::staticData);
    MyClass* dynamicObj2 = new MyClass(7, 8.5, "Dynamic2", "Object", MyClass::staticData);

    std::cout << "\nDetails of localObj1:" << std::endl;
    // 输出 localObj1 的信息...

    std::cout << "\nDetails of funcObj:" << std::endl;
    // 输出 funcObj 的信息...

    std::cout << "\nDetails of dynamicObj1:" << std::endl;
    // 输出 dynamicObj1 的信息...

    // 释放动态分配的对象
    delete dynamicObj1;
    delete dynamicObj2;

    return 0;
}
```

输出结果会显示每个对象的静态和非静态数据成员的值、地址以及对象的存储空间大小。通过观察这些输出，我们可以理解C++对象模型中对象的本质、静态数据成员的共享情况，以及对齐现象。

### 三、输出结果说明与分析

(输出结果见附页)

## 1. 全局对象的构造和析构:

```
++++++ Construct Global 1 is created ++++++  
++++++ Construct Global 2 is created ++++++
```

这两行显示了全局对象 `globalObj1` 和 `globalObj2` 的构造。同时显示在 `main` 函数之前，说明全局对象在 `main` 函数执行前就已经创建。

## 2. `main` 函数的开始和结束标记:

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ MAIN FUNCTION BEGIN:  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  
  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ MAIN FUNCTION END  
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

这两行表示了 `main` 函数的开始和结束。

## 3. 对象的构造和析构:

```
++++++ Construct Local 1 is created ++++++  
++++++ Construct Local 2 is created ++++++
```

这两行显示了局部对象 `localObj1` 和 `localObj2` 的构造。

```
++++++ Destruct Local 2 is deleted ++++++  
++++++ Destruct Local 1 is deleted ++++++
```

这两行显示了局部对象 `localObj2` 和 `localObj1` 的析构。构造和析构顺序符合对象的创建和销毁顺序。

```
++++++ Construct Func Object is created ++++++
```

这行显示了外部函数 `func` 中的局部对象 `funcObj` 的构造。

```
++++++ Destruct Func Object is deleted ++++++
```

这行显示了外部函数 `func` 中的局部对象 `funcObj` 的析构。

```
++++++ Construct Dynamic1 is created ++++++
++++++ Construct Dynamic2 is created ++++++
```

这两行显示了动态创建的对象 dynamicObj1 和 dynamicObj2 的构造。

```
++++++ Destruct Dynamic1 is deleted ++++++
++++++ Destruct Dynamic2 is deleted ++++++
```

这两行显示了动态创建的对象 dynamicObj1 和 dynamicObj2 的析构。

## 4. 对象的静态和非静态数据成员的值、地址和存储空间:

### 1. 静态数据成员地址:

- 静态数据成员 staticData 在内存中只有一份拷贝，无论有多少个 MyClass 对象被创建，它们都共享同一份静态数据成员。
- 静态数据成员被定义为 static int staticData。因此，无论是通过类名还是对象名来访问该静态成员，都指向同一个地址。

### 2. 非静态数据成员地址:

- 每个对象的非静态数据成员都有自己的地址，即使两个对象的数据成员值相同，它们的地址也不同。
- 输出中显示了全局对象 globalObj1 和 globalObj2 的地址不同，即使它们具有相同的数据成员值，但它们的地址不同。对于局部对象和动态创建的对象，同样也可以观察到它们的地址不同，这表明它们的非静态数据成员在内存中占据不同的位置。

### 3. 全局对象 globalObj1 和 globalObj2:

- 这些对象的静态数据成员 staticData 在内存中只有一份拷贝，由于是全局对象，它们的地址位于静态存储区。
- 其非静态数据成员（如 integerData、doubleData、charPtrData、stringData 等）分别存储在这些对象所在的内存空间中。

### 4. 局部对象 localObj1 和 localObj2:

- 这些对象的静态数据成员 staticData 在内存中仍然只有一份拷贝，但由于是局部对象，它们的地址位于栈上。
- 同样，其非静态数据成员也存储在对象所在的内存空间中，这些对象的地址位于栈上。（0x7FFFFFFF...）

### 5. 动态创建的对象 dynamicObj1 和 dynamicObj2:

- 与局部对象类似，这些对象的静态数据成员 staticData 仍然共享一份拷贝，但由于是动态创建的，它们的地址位于堆上。
- 非静态数据成员也存储在对象所在的堆上的内存空间中。

## 四、结论总述

- globalObj1 和 globalObj2 是在全局作用域下创建的对象，它们在程序启动时被创建，构造函数被调用，输出了相关的构造信息。在程序结束时，析构函数被调用，释放了相应的资源。

- `localObj1` 和 `localObj2` 是在 `main()` 函数中定义的局部对象，它们在 `main()` 函数执行时创建，在 `main()` 函数结束时销毁。在创建时，输出了构造函数的信息，在销毁时，输出了析构函数的信息。
- `dynamicObj1` 和 `dynamicObj2` 是通过 `new` 运算符在堆上分配内存创建的对象。它们在堆上分配了内存，需要手动释放。在创建时，输出了构造函数的信息，在释放内存时，输出了析构函数的信息。
- 输出了非静态成员函数和静态成员函数的地址。可以观察到这些函数在内存中的位置。非静态成员函数的地址和静态成员函数的地址是固定的，不依赖于对象的实例。
- 通过输出对象的大小，观察到了对象在内存中的布局情况。可以发现对象的大小并不等于所有数据成员的大小之和，这是因为编译器对对象进行了对齐操作。这种对齐操作是为了提高访问速度和内存访问的效率。
- 通过观察对象的构造和析构函数的调用，可以了解对象的生命周期。全局对象在程序启动时被创建，在程序结束时被销毁；局部对象在函数执行时创建，在函数结束时销毁；动态对象则由程序员手动创建和销毁。

```

1  /* 3.2 实验： 研究C++的对象模型 */
2
3
4  #include <iostream>
5  #include <cstring>
6
7  class MyClass
8  {
9  public:
10     static int staticData;
11     int integerData;
12     double doubleData;
13     char *charPtrData;
14     std::string stringData;
15     int &refData;
16
17 public:
18     // Constructor with dynamic memory allocation
19     MyClass(int intValue, double doubleValue, const char *charValue, const std::string
&stringValue, int &ref)
20         : integerData(intValue), doubleData(doubleValue), stringData(stringValue), refData(ref)
21     {
22         charPtrData = new char[strlen(charValue) + 1];
23         strcpy(charPtrData, charValue);
24         std::cout << "      "
25                 << "      "
26                 << "+++++ Construct " << charPtrData << " is created +++++" << std::endl;
27     }
28
29     MyClass(const MyClass &obj)
30         : integerData(obj.integerData), doubleData(obj.doubleData), stringData(obj.stringData),
refData(obj.refData)
31     {
32         charPtrData = new char[strlen(obj.charPtrData) + 1];
33         strcpy(charPtrData, obj.charPtrData);
34         std::cout << "      "
35                 << "      "
36                 << "+++++ Copy Construct " << charPtrData << " is created +++++" << std::endl;
37     }
38
39     // Destructor
40     ~MyClass()
41     {
42         std::cout << "      "
43                 << "      "
44                 << "+++++ Destruct " << charPtrData << " is deleted +++++" << std::endl;
45         delete[] charPtrData;
46     }
47
48     // Static member function
49     static void staticFunction()
50     {
51         std::cout << "Static function called" << std::endl;
52     }
53
54     // Non-static member function
55     void nonStaticFunction()
56     {
57         std::cout << "Non-static function called" << std::endl;
58     }
59
60     void printData()

```

```

61 {
62
63     std::cout << " "
64         << "----- Sec#1: Data of object -----" << std::endl;
65
66     std::cout << " "
67         << "in Object: " << this << std::endl;
68     std::cout << " "
69         << "Integer data: " << integerData << std::endl;
70     std::cout << " "
71         << "Double data: " << doubleData << std::endl;
72     std::cout << " "
73         << "Char pointer data: " << charPtrData << std::endl;
74     std::cout << " "
75         << "String data: " << stringData << std::endl;
76 }
77
78 void printDataAddress()
79 {
80     std::cout << " "
81         << "----- Sec#2: Address of data -----" << std::endl;
82
83     std::cout << " "
84         << "in Object: " << this << std::endl;
85     std::cout << " "
86         << "Integer data: " << &integerData << std::endl;
87     std::cout << " "
88         << "Double data: " << &doubleData << std::endl;
89     std::cout << " "
90         << "Char pointer data: " << reinterpret_cast<void *>(charPtrData) << std::endl;
91     std::cout << " "
92         << "String data: " << &stringData << std::endl;
93 }
94 };
95
96 // Initializing static data member
97 int MyClass::staticData = -1010;
98
99 // Global objects
100 MyClass globalObj1(1, 2.3, "Global 1", "Object", MyClass::staticData);
101 MyClass globalObj2(2, 3.7, "Global 2", "Object", MyClass::staticData);
102
103 // External function
104 void func(MyClass obj)
105 {
106     obj.printData();
107     obj.printDataAddress();
108 }
109
110 int main()
111 {
112     std::cout << "\n$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ MAIN FUNCTION BEGIN:
113     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n"
114         << std::endl;
115
116     std::cout << "\n----- Para#0: Initialization -----" <<
std::endl;
117     // Local objects in main()
118     MyClass localObj1(3, 4.9, "Local 1", "Object", MyClass::staticData);
119     MyClass localObj2(4, 5.1, "Local 2", "Object", MyClass::staticData);
120     // Dynamic objects in main()
121     MyClass *dynamicObj1 = new MyClass(5, 6.3, "Dynamic1", "Object", MyClass::staticData);
122     MyClass *dynamicObj2 = new MyClass(6, 7.2, "Dynamic2", "Object", MyClass::staticData);

```



```

122     std::cout << "-----END-----\n"
123         << std::endl;
124
125     // Calling external function func()
126     std::cout << "\n----- Para#1: Inside External func() -----" <<
std::endl;
127     std::cout << "    "
128         << "Call external func(): " << &func << " with object: " << &localObj1 << std::endl;
129     func(localObj1);
130     std::cout << "-----END-----\n"
131         << std::endl;
132
133     std::cout << "\n----- Para#2: Inside main() -----" <<
std::endl;
134     std::cout << "    "
135         << "Global object 1 address: " << &globalObj1 << std::endl;
136     std::cout << "    "
137         << "Global object 2 address: " << &globalObj2 << std::endl;
138     std::cout << "    "
139         << "Local object 1 address: " << &localObj1 << std::endl;
140     std::cout << "    "
141         << "Local object 2 address: " << &localObj2 << std::endl;
142     std::cout << "    "
143         << "Dynamic object 1 address: " << &dynamicObj1 << std::endl;
144     std::cout << "    "
145         << "Dynamic object 2 address: " << &dynamicObj2 << std::endl;
146
147     std::cout << "    "
148         << "Size of MyClass: " << sizeof(MyClass) << " bytes" << std::endl;
149     std::cout << "-----END-----\n"
150         << std::endl;
151
152     std::cout << "\n----- Para#3: Details of globalObj1 -----" <<
std::endl;
153     globalObj1.printData();
154     globalObj1.printDataAddress();
155     // print function address
156     void (MyClass::*pmf)() = &MyClass::nonStaticFunction;
157     unsigned *p0 = (unsigned *)&pmf;
158     std::cout << "    "
159         << "nonStaticFunction address: " << std::hex << std::noshowbase << *p0 << std::endl;
160     // print static function address
161     void (*pf)() = &MyClass::staticFunction;
162     unsigned *p1 = (unsigned *)&pf;
163     std::cout << "    "
164         << "staticFunction address: " << std::hex << std::noshowbase << *p1 << std::endl;
165     std::cout << "-----END-----\n"
166         << std::endl;
167
168     std::cout << "\n----- Para#4: Details of localObj1 -----" <<
std::endl;
169     localObj1.printData();
170     localObj1.printDataAddress();
171     // print function address
172     void (MyClass::*pmf1)() = &MyClass::nonStaticFunction;
173     unsigned *p2 = (unsigned *)&pmf1;
174     std::cout << "    "
175         << "nonStaticFunction address: " << std::hex << std::noshowbase << *p2 << std::endl;
176     // print static function address
177     void (*pf1)() = &MyClass::staticFunction;
178     unsigned *p3 = (unsigned *)&pf1;
179     std::cout << "    "
180         << "staticFunction address: " << std::hex << std::noshowbase << *p3 << std::endl;

```

```

181     std::cout << "-----END-----\n"
182         << std::endl;
183
184     std::cout << "\n----- Para#5: Details of dynamicObj1 -----" <<
std::endl;
185     dynamicObj1->printData();
186     dynamicObj1->printDataAddress();
187     // print function address
188     void (MyClass::*pmf2)() = &MyClass::nonStaticFunction;
189     unsigned *p4 = (unsigned *)&pmf2;
190     std::cout << "      "
191         << "nonStaticFunction address: " << std::hex << std::noshowbase << *p4 << std::endl;
192     // print static function address
193     void (*pf2)() = &MyClass::staticFunction;
194     unsigned *p5 = (unsigned *)&pf2;
195     std::cout << "      "
196         << "staticFunction address: " << std::hex << std::noshowbase << *p5 << std::endl;
197     std::cout << "-----END-----\n"
198         << std::endl;
199
200     // Freeing dynamic objects
201     delete dynamicObj1;
202     delete dynamicObj2;
203
204     std::cout << "\n$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ MAIN FUNCTION END
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$\n"
205         << std::endl;
206
207     return 0;
208 }

```

```

1  /*****\
2  * 以下是原始输出 *
3  \*****/
4
5
6      +++++ Construct Global 1 is created +++++
7      +++++ Construct Global 2 is created +++++
8
9  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ MAIN FUNCTION BEGIN:
10 $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
11
12 ----- Para#0: Initialization -----
13      +++++ Construct Local 1 is created +++++
14      +++++ Construct Local 2 is created +++++
15      +++++ Construct Dynamic1 is created +++++
16      +++++ Construct Dynamic2 is created +++++
17 -----END-----
18
19
20 ----- Para#1: Inside External func() -----
21      Call external func(): 1 with object: 0x7ffc0032c5b0
22      +++++ Copy Construct Local 1 is created +++++
23      ----- Sec#1: Data of object -----
24      in Object: 0x7ffc0032c6b0
25      Integer data: 3
26      Double data: 4.9
27      Char pointer data: Local 1
28      String data: Object
29      ----- Sec#2: Address of data -----
30      in Object: 0x7ffc0032c6b0
31      Integer data: 0x7ffc0032c6b0
32      Double data: 0x7ffc0032c6b8
33      Char pointer data: 0x563bdcc92020
34      String data: 0x7ffc0032c6c8
35      +++++ Destruct Local 1 is deleted +++++
36 -----END-----
37
38
39 ----- Para#2: Inside main() -----
40      Global object 1 address: 0x563bdbb38260
41      Global object 2 address: 0x563bdbb382a0
42      Local object 1 address: 0x7ffc0032c5b0
43      Local object 2 address: 0x7ffc0032c570
44      Dynamic object 1 address: 0x563bdcc91f40
45      Dynamic object 2 address: 0x563bdcc91fb0
46      Size of MyClass: 64 bytes
47 -----END-----
48
49
50 ----- Para#3: Details of globalObj1 -----
51      ----- Sec#1: Data of object -----
52      in Object: 0x563bdbb38260
53      Integer data: 1
54      Double data: 2.3
55      Char pointer data: Global 1
56      String data: Object
57      ----- Sec#2: Address of data -----
58      in Object: 0x563bdbb38260
59      Integer data: 0x563bdbb38260
60      Double data: 0x563bdbb38268
61      Char pointer data: 0x563bdcc90eb0

```

```

62     String data: 0x563bdbb38278
63     nonStaticFunction address: dbb357a4
64     staticFunction address: dbb35772
65     -----END-----
66
67
68     ----- Para#4: Details of localObj1 -----
69     ----- Sec#1: Data of object -----
70     in Object: 0x7ffc0032c5b0
71     Integer data: 3
72     Double data: 4.9
73     Char pointer data: Local 1
74     String data: Object
75     ----- Sec#2: Address of data -----
76     in Object: 0x7ffc0032c5b0
77     Integer data: 0x7ffc0032c5b0
78     Double data: 0x7ffc0032c5b8
79     Char pointer data: 0x563bdcc91f00
80     String data: 0x7ffc0032c5c8
81     nonStaticFunction address: dbb357a4
82     staticFunction address: dbb35772
83     -----END-----
84
85
86     ----- Para#5: Details of dynamicObj1 -----
87     ----- Sec#1: Data of object -----
88     in Object: 0x563bdcc91f40
89     Integer data: 5
90     Double data: 6.3
91     Char pointer data: Dynamic1
92     String data: Object
93     ----- Sec#2: Address of data -----
94     in Object: 0x563bdcc91f40
95     Integer data: 0x563bdcc91f40
96     Double data: 0x563bdcc91f48
97     Char pointer data: 0x563bdcc91f90
98     String data: 0x563bdcc91f58
99     nonStaticFunction address: dbb357a4
100    staticFunction address: dbb35772
101    -----END-----
102
103    ++++++ Destruct Dynamic1 is deleted ++++++
104    ++++++ Destruct Dynamic2 is deleted ++++++
105
106    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ MAIN FUNCTION END
107    $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
108
109    ++++++ Destruct Local 2 is deleted ++++++
110    ++++++ Destruct Local 1 is deleted ++++++
111    ++++++ Destruct Global 2 is deleted ++++++
112    ++++++ Destruct Global 1 is deleted ++++++

```