

编号: 1-1



山东师范大学  
SHANDONG NORMAL UNIVERSITY

## 信息科学与工程学院实验报告

### 《面向对象程序设计》

### Object-Oriented Programming

姓名: 熊淑贤

学号: 201911010517

班级: 计本 1901

时间: 2020 年 11 月 17 日



## 《面向对象程序设计》实验报告

**基本要求：**请围绕实验目的、实验内容、实验过程、实验结果（附图）、实验总结（重点阐述）五个部分进行撰写。若报告中若涉及源代码内容，请在附录部分提供完整源码及 GitHub 源码托管地址。报告撰写完毕后请提交 PDF 格式版本到云班课。

### 一、实验目的

- 熟练掌握复杂类的设计方法（三构一析+普函）
- 熟练掌握静态数据成员和函数的声明、定义及用法（static）
- 掌握常数据，常函数和常对象的声明、定义及用法（const）
- 掌握友元函数的声明、定义和使用方法（friend）
- 熟练掌握 visual studio 创建类的快捷方法（实战）
- 了解代码编辑器 VIM，以及 VIM 的有关基本用法（扩展）

### 二、实验内容

#### 【简答题】

- 什么是静态成员数据，其本质和主要作用是什么？
- 什么是静态成员函数，在类中为何要设计该函数？
- 静态数据成员与普通数据成员有何区别与联系？
- 类中可以有哪些类型的成员数据和成员函数，请详细列举并简要解释其含义。

**【程序设计题】**请采用 C++面向对象程序设计思想设计类，并实现类。 要求程序代码：清晰简洁易读，良好的开闭性， 编程风格规范统一。

设计某高校研究生 Graduate 类，该类包含研究生姓名（string name）、性别（bool gender）、学号（int id）、入学成绩（double score）、研究领域（string research）、指导教师（string tutor）和学校名称（const string university）与该研究生类有关的函数如下：

- 初始化与释放研究生对象数据（三构一析）；
- 采用普通成员函数形式输出研究生基本数据信息；
- 采用常成员函数形式输出研究生的基本数据信息；
- 输出多个研究生对象的平均成绩（要求在类内封装实现）；
- 以友元函数形式计算每个研究生对象的成绩等级（90 分以上输出 A，80-89 分输出 B，70-79 分输出 C，60-69 分输出 D，60 分以下输出 E）；

要求：设计实现该类，并在 main 函数内定义研究生对象数组，然后，再单独定义普通研究生对象和常研究生对象，最后，输出各对象的详细信息输出所有研究生的平均成绩与成绩等级。



### 三、实验过程

#### 【简答题】

- 什么是静态数据成员，其本质和主要作用是什么？

静态数据成员是为所有同类对象共享，必须在类外进行初始化；本质是一个全局对象，作用是便于类的对象共享类型访问。

- 什么是静态成员函数，在类中为何要设计该函数？

静态成员函数一般专门用来操作静态数据成员，没有 `this` 指针。

- 静态数据成员与普通数据成员有何区别与联系？

静态数据成员属于类，为所有同类对象所共享，而普通数据成员不能被共享。

- 类中可以有哪些类型的成员数据和成员函数，请详细列举并简要解释其含义。

普通数据成员和普通成员函数；

静态数据成员：`static` 修饰，存储在静态区，可被所有同类对象共享。

静态成员函数：一般专门用于访问静态数据成员。

常数据成员：用 `const` 修饰，提供一个不变化的量，只在类范围内使用。

常成员函数：对类中成员只访问不修改。

### 四、实验结果

```
-----各对象的详细信息如下-----
姓    名: Alice
性    别: F
学    号: 65
入学成绩: 92
研究领域: Computer System Architecture
指导教师: Edwin
学校名称: SDNU

姓    名: Alex
性    别: M
学    号: 66
入学成绩: 85
研究领域: Computer and Theory
指导教师: Apollo
学校名称: SDNU

姓    名: Ford
性    别: M
学    号: 67
入学成绩: 76
研究领域: Computer Application Technology
指导教师: Cole
学校名称: SDNU

3名研究生的平均成绩为 84.3333

研究生Alice的成绩等级为 : A
研究生Alex的成绩等级为 : B
研究生Ford的成绩等级为 : C
```



## 五、实验总结

程序实践是重要教学环节之一。我们需要通过程序实践，增强工程化意识，提高 C/C++ 语言实践能力。强化计算机应用技能，从而巩固和充实所学的理论知识，加深对相关内容的理解，拓宽知识面，培养的创新精神和实践能力。

通过这次实验，我掌握了复杂类的设计方法（三构一析+普函）；静态数据成员和函数的声明、定义及用法；常数据，常函数和常对象的声明、定义及用法（const）；友元函数的声明、定义和使用方法（friend）；visual studio 创建类的快捷方法（实战）；了解了代码编辑器 VIM，以及 VIM 的有关基本用法（扩展）。

### ■ 附录：程序源码（建议基于 Highlight 软件导入）

GitHub 源码托管地址：

<https://github.com/Shuxian-X/the-first/tree/master/%E7%AC%AC%E5%9B%9B%E7%AB%A0%E7%AC%AC%E4%B8%80%E6%AC%A1%E4%BD%9C%E4%B8%9A>

```
01 #include <iostream>
02 #include <string>
03
04 using namespace std;
05
06 class Student
07 {
08 private:
09     string name; //姓名
10     bool gender; //性别, 1 男 0 女
11     int id; //学号
12     double score; //入学成绩
13     string research; //研究领域
14     string tutor; //指导教师
15     const string university; //学校名称
16
17 public:
18     Student(); //构造函数
19     Student(string na, bool g, int ID, double sco, string re, string tu);
20     Student(const Student &per); //拷贝构造函数
21     ~Student(); //析构函数
22
23     string Get_name(); // 获取研究生对应数据信息
```



```
24  bool Get_gender();
25  int Get_id();
26  double Get_score();
27  string Get_research();
28  string Get_tutor();
29  string Get_university();
30
31  void Display(); //普通成员函数 输出研究生基本数据信息
32  void Display()const; //常成员函数 输出研究生基本数据信息
33  double Output_GPA(const Student *stu, int n); //输出多个研究生对象的平均成绩
34  friend void Output_level(const Student &stu); //友元函数 输出每个研究生对象的成绩等级
35  };
36
37  Student::Student() : university("SDNU") //构造函数,在初始化列表里对const数据成员初始化
38  {
39  }
40
41  Student::Student(string na, bool g, int ID, double sco, string re, string tu) :
university("SDNU") //构造函数,在初始化列表里对const数据成员初始化
42  {
43      name = na;
44      gender = g;
45      id = ID;
46      score = sco;
47      research = re;
48      tutor = tu;
49  }
50
51  Student::Student(const Student & per) : university("SDNU") //构造函数,在初始化列表里对const数
据成员初始化
52  {
53      name = per.name;
54      gender = per.gender;
55      id = per.id;
56      score = per.score;
57      research = per.research;
58      tutor = per.tutor;
59  }
60
61  Student::~~Student()
62  {
63  }
64
65  string Student::Get_name()
66  {
```



```
67     return name;
68 }
69
70 bool Student::Get_gender()
71 {
72     return gender;
73 }
74
75 int Student::Get_id()
76 {
77     return id;
78 }
79
80 double Student::Get_score()
81 {
82     return score;
83 }
84
85 string Student::Get_research()
86 {
87     return research;
88 }
89
90 string Student::Get_tutor()
91 {
92     return tutor;
93 }
94
95 string Student::Get_university()
96 {
97     return university;
98 }
99
100 void Student::Display() //普通成员函数 输出研究生基本数据信息
101 {
102     cout << "姓    名: " << name << endl;
103     cout << "性    别: " << (gender ? "M" : "F") << endl;
104     cout << "学    号: " << id << endl;
105     cout << "入学成绩: " << score << endl;
106     cout << "研究领域: " << research << endl;
107     cout << "指导教师: " << tutor << endl;
108     cout << "学校名称: " << university << endl << endl;
109 }
110
111 void Student::Display() const //常成员函数 输出研究生基本数据信息
```



```
112 {
113     cout << "姓    名: " << name << endl;
114     cout << "性    别: " << (gender ? "M" : "F") << endl;
115     cout << "学    号: " << id << endl;
116     cout << "入学成绩: " << score << endl;
117     cout << "研究领域: " << research << endl;
118     cout << "指导教师: " << tutor << endl;
119     cout << "学校名称: " << university << endl << endl;
120 }
121
122 double Student::Output_GPA(const Student * stu, int n) //输出多个研究生对象的平均成绩
123 {
124     double sum = 0;
125     for (int i = 0; i < n; ++i)
126         sum += stu[i].score;
127     double res = sum / n;
128     cout << n << "名研究生的平均成绩为 " << res << endl << endl;
129     return res;
130 }
131
132 void Output_level(const Student &stu) //友元函数 输出研究生对象的成绩等级
133 {
134     int tp = (int)(stu.score / 10);
135     switch (tp)
136     {
137     case 10:
138     case 9:
139         cout << "研究生" << stu.name << "的成绩等级为 :  A" << endl;
140         break;
141     case 8:
142         cout << "研究生" << stu.name << "的成绩等级为 :  B" << endl;
143         break;
144     case 7:
145         cout << "研究生" << stu.name << "的成绩等级为 :  C" << endl;
146         break;
147     case 6:
148         cout << "研究生" << stu.name << "的成绩等级为 :  D" << endl;
149         break;
150     default:
151         cout << "研究生" << stu.name << "的成绩等级为 :  E" << endl;
152         break;
153     }
154 }
155
156 int main()
```



```
157 {
158     //研究生对象数组
159     Student stu[3] =
160     {
161         Student("Alice", 0, 0101, 92, "Computer System Architecture", "Edwin"),
162         Student("Alex", 1, 0102, 85, "Computer and Theory", "Apollo"),
163         Student("Ford", 1, 0103, 76, "Computer Application Technology", "Cole"),
164     };
165
166     //普通研究生对象
167     //Student a("Ada", 0, 0104, 446, "Information Security and Engineering", "Barbara");
168     Student a(stu[0]);
169
170     //常研究生对象
171     //const Student b("Cathy", 0, 0105, 401, "Computer Science and Technology", "Diana");
172     const Student b(stu[1]);
173
174     cout << "-----各对象的详细信息如下-----" << endl;
175     for (int i = 0; i < 3; ++i)
176         stu[i].Display();
177
178     stu[0].Output_GPA(stu, 3);
179
180     for(int i = 0; i < 3; ++i)
181         Output_level(stu[i]);
182
183     return 0;
184 }
```