

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 21 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

- 掌握类的基本设计方法(成员变量+成员函数)
- 熟练掌握构造函数与析构函数设计方法
- 掌握对象数组，对象指针，对象引用用法
- 掌握友元的概念及友元函数的定义和使用
- 熟练掌握 visual studio 代码调试方法

### 二、实验内容

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。

**【编程题 1】**请设计一个栈类 Stack，通过该类实现与栈相关的各种操作。常见栈类功能包括： 栈内数据初始化、数据进栈、数据出栈、判断栈是空的还是满的、计算当前栈的大小等。

**提示：**栈内数据存储时可以借助数组实现。

**【编程题 2】**请设计一个字符串类 Mystring，通过该类实现与字符串相关的各种操作。 字符串类的主要功能包括字符串连接、字符串比较、字符串增删改查操作、字符串反转操作、字符串长度计算等。

**【编程题 3】**请设计一个三角形类 Triangle，其中三角形信息包括：各边边长；三角形类对外能提供的接口功能包括：判断能否构成三角形、输出三角形周长、输出三角形面积、输出三角形类型（直角，锐角，还是钝角三角形）。

**【编程题 4】**请设计一个学生类 Student，其中学生的信息包括：姓名、性别、学号、出生日期、家庭住址、6 个科目的考试成绩。该类对外提供的函数接口功能包括：

构造函数：初始化学生类对象（无参构造，有参构造，拷贝构造，至少 3 种）

普通函数：获取学生任意信息，修改学生成绩信息，输出学生信息

析构函数：释放学生类对象内存空间

然后基于学生类回答下列 2 个问题：

1) 分析学生类对象及其成员变量的存储规律（可借助对象或成员变量的地址进行分析）。

2) 阐述普通构造函数（传值初始化新对象），拷贝构造函数（对象初始化对象）在初始化对象过程中的联系与区别。



【编程题 5】请设计一个点类 Point，类内包含点的二维坐标数值，通过该类实现空间点坐标的初始化，坐标修改，坐标位置输出功能；然后基于该 Point 类设计一个友元函数和一个友元类，实现计算两个点之间的直线距离。

### 三、 实验过程

仔细阅读课本给出范例，上网广泛查阅参考资料，内化于心，外化于形，动手设计各种类

### 四、 实验结果

```
*****栈类设计与测试*****
```

```
入栈，请输入数据（0终止）：2  
成功入栈
```

```
入栈，请输入数据（0终止）：4  
成功入栈
```

```
入栈，请输入数据（0终止）：6  
成功入栈
```

```
入栈，请输入数据（0终止）：8  
成功入栈
```

```
入栈，请输入数据（0终止）：10  
成功入栈
```

```
入栈，请输入数据（0终止）：12  
成功入栈
```

```
入栈，请输入数据（0终止）：0  
*****入栈完毕!*****
```

```
从栈顶依次输出当前栈内 6个元素并出栈：  
12 10 8 6 4 2
```

```
C:\Users\elephant\source\repos\Project1\Debug\Project1.exe (进程 17544) 已退出，返回代码为：0。  
按任意键关闭此窗口...
```



```
s1 != empty
s1.size = 25
after reverse, s1 =
.kcul ruoy wonk reven uoY
after reverse again and append s2, s1 =
You never know your luck.Sow nothing, reap nothing.
after append the first three characters of s1, s2 =
Sow nothing, reap nothing. You
after erase, s2 =
Sow nothing, reap nothing.
after erase, s1 =
You never know your luck.
s1 > s2
in s1, k's location of the first occurrence is
10
in s1, luck's location of the first occurrence is
20
C:\Users\elegant\source\repos\Project1\Debug\Project1.exe (进程 16224) 已退出, 返回代码为: 0。
按任意键关闭此窗口...
```



```
*****1. 录入信息*****
*****2. 查询信息*****
*****3. 浏览信息*****
*****4. 修改信息*****
*****0. 退出*****
please input you choice: 1
输入姓名: Alice
输入性别: F
输入学号: 201911010101
输入出生日期: 20000101
输入家庭住址: Utopia
输入语文成绩: 137
输入数学成绩: 145
输入英语成绩: 148
输入物理成绩: 98
输入化学成绩: 96
输入生物成绩: 98
继续输入吗? (Y or N)
Y
输入姓名: Bob
输入性别: M
输入学号: 201911010102
输入出生日期: 20000401
输入家庭住址: Utopia
输入语文成绩: 126
输入数学成绩: 150
输入英语成绩: 150
输入物理成绩: 97
输入化学成绩: 98
输入生物成绩: 100
继续输入吗? (Y or N)
N
*****1. 录入信息*****
*****2. 查询信息*****
*****3. 浏览信息*****
*****4. 修改信息*****
*****0. 退出*****
please input you choice: 2
Input the name searched
Bob
姓名: Bob
性别: M
学号: 201911010102
出生日期: 20000401
家庭住址: Utopia
语文成绩: 126
数学成绩: 150
英语成绩: 150
物理成绩: 97
化学成绩: 98
生物成绩: 100
*****1. 录入信息*****
*****2. 查询信息*****
*****3. 浏览信息*****
*****4. 修改信息*****
*****0. 退出*****
please input you choice: 4
Input the name, subject number, new grade:
Bob 4 100
*****1. 录入信息*****
*****2. 查询信息*****
*****3. 浏览信息*****
*****4. 修改信息*****
*****0. 退出*****
please input you choice: 3
学生总人数 = 2
姓名: Alice
性别: F
学号: 201911010101
出生日期: 20000101
家庭住址: Utopia
语文成绩: 137
数学成绩: 145
英语成绩: 148
物理成绩: 98
化学成绩: 96
生物成绩: 98
姓名: Bob
性别: M
学号: 201911010102
出生日期: 20000401
家庭住址: Utopia
语文成绩: 126
数学成绩: 150
英语成绩: 150
物理成绩: 100
化学成绩: 98
生物成绩: 100
*****1. 录入信息*****
*****2. 查询信息*****
*****3. 浏览信息*****
*****4. 修改信息*****
*****0. 退出*****
please input you choice: 0
```

滚动鼠标轴或单击, 开始截长图



```
It can form a triangle!  
The circumference of the triangle is 24  
The area of the triangle is 24  
The type is right-angled triangle!
```

```
the distance between a and b is: 21.8403
```

```
name1 = Alice, x = 3, y = 4  
name2 = Cindy, x = 1, y = 1  
distance is = 3.60555
```

```
C:\Users\elephant\source\repos\Project1\Debug\Project1.exe (进程 6160) 已退出，返回代码为: 0。  
按任意键关闭此窗口...
```

## 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，我理解了 c++对 C 的各项改进和扩展基本原理，掌握了 C++类的基本设计方法（成员变量+成员函数；掌握了构造函数与析构函数设计方法；掌握了对象数组，对象指针，对象引用用法；掌握了友元的概念及友元函数的定义和使用；能熟练掌握 visual studio 代码调试方法，熟练运用 C++特色函数解决实际问题， 总体达成了实验目标。



## ■ 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%E4%B8%89%E7%AB%A0%E7%AC%AC%E4%B8%80%E6%AC%A1%E4%BD%9C%E4%B8%9A](https://github.com/Shuxian-X/the-first/tree/master/%E7%AC%AC%E4%B8%89%E7%AB%A0%E7%AC%AC%E4%B8%80%E6%AC%A1%E4%BD%9C%E4%B8%9A)

**【编程题 1】**请设计一个栈类 Stack，通过该类实现与栈相关的各种操作。常见栈类功能包括：栈内数据初始化、数据进栈、数据出栈、判断栈是空的还是满的、计算当前栈的大小等。

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <iostream>
03 #include "student.h"
04 #include <cstring>
05 #include <cmath>
06 #include <algorithm>
07 using namespace std;
08
09 const int N = 106;
10
11 class Stack
12 {
13 private:
14     int top;
15     double data[N];
16
17 public:
18     Stack(); //构造函数
19     ~Stack(); //析构函数
20
21     void init(); //初始化
22     bool full(); //判满
23     bool empty(); //判空
24     void push(double &val); //入栈
25     double pop(); //出栈
26     int size(); //求大小
27     double get_top(); //求栈顶元素
28 };
29
30 Stack::Stack() //构造函数
31 {
32     top = 0;
33     memset(data, 0, sizeof data);
34 }
35
```



```
36 Stack::~~Stack()//析构函数
37 {
38 }
39
40 void Stack::init() //初始化
41 {
42     top = 0;
43     memset(data, 0, sizeof data);
44 }
45
46 bool Stack::full() //判满
47 {
48     if (top == N)
49         return 1;
50     return 0;
51 }
52
53 bool Stack::empty() //判空
54 {
55     if (top == 0)
56         return 1;
57     return 0;
58 }
59
60 void Stack::push(double & val) //入栈
61 {
62     data[top++] = val;
63 }
64
65 double Stack::pop() //出栈
66 {
67     return data[--top];
68 }
69
70 int Stack::size() //求大小
71 {
72     return top;
73 }
74
75 double Stack::get_top() //求栈顶元素
76 {
77     return data[top-1];
78 }
79
80
```





```
81 int main()
82 {
83     cout << "*****栈类设计与测试*****" << endl;
84
85     Stack sta;
86     double tp = 0;
87     while (!sta.full())
88     {
89         cout << "入栈, 请输入数据 (0终止) : ";
90         cin >> tp;
91         if (!tp)
92         {
93             cout << "*****入栈完毕!*****" <<
94             endl << endl;
95             break;
96         }
97         sta.push(tp);
98         cout << "成功入栈\n\n";
99     }
100     cout << "从栈顶依次输出当前栈内 " << sta.size(
101     ) << "个元素并出栈: \n";
102     while (!sta.empty())
103     {
104         cout << sta.pop() << " ";
105     }
106     cout << endl;
107     return 0;
108 }
```

【编程题 2】请设计一个字符串类 Mystring，通过该类实现与字符串相关的各种操作。字符串类的主要功能包括字符串连接、字符串比较、字符串增删改查操作、字符串反转操作、字符串长度计算等。

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <iostream>
03 #include <cstring>
04 #include <cmath>
05 #include <algorithm>
06 using namespace std;
07
08 const int npos = 0x3f3f3f3f;
09
10 class Mystring
11 {
12 private:
```



```
13  char *p_str;
14  int length;
15
16 public:
17  Mystring();
18  Mystring(const char *str);
19  ~Mystring();
20  bool empty(); //判断字符串是否为空
21  int size(); //计算当前字符串长度
22  char *display(); //输出当前字符串
23  void reverse(); //把当前字符串反转
24  int append(Mystring &s); //把字符串s连接到当前字符串结尾
25  int append(Mystring &s, int n); //把字符串s的前n个字符连接到当前字符串结尾
26  int compare(const Mystring &s);
27  //比较当前字符串和s的大小, 返回值为 1 表示当前字符串大于给定字符串 s, 为 0 相等, 为 -1 小于
28  int insert(Mystring &s, int pos = 0); //在当前字符串第 pos 位置前插入字符串 s
29  int erase(int pos = 0, int n = 0); //删掉当前字符串 pos 开始, 长度为 n 的字符
30  int find(const char c, int pos = 0); //从pos开始查找字符c在当前字符串中第一次出现的位置
31  int find(const Mystring &s, int pos = 0); //从pos开始查找字符串s在当前串中第一次出现的位置
32 };
33
34 Mystring::Mystring() :p_str(NULL), length(0) {}
35
36 Mystring::Mystring(const char * str)
37 {
38     length = strlen(str);
39     p_str = new char[length + 1];
40     strcpy(p_str, str);
41     p_str[length] = '\0';
42 }
43
44 Mystring::~Mystring()
45 {
46     delete[] p_str;
47 }
48
49 bool Mystring::empty() //判断字符串是否为空
50 {
51
52     return length ==0 ? true : false;
53 }
54
55 int Mystring::size() //计算当前字符串长度
56 {
57     return length;
```



```
58 }
59
60 char * Mystring::display() //输出当前字符串
61 {
62     return p_str;
63 }
64
65 void Mystring::reverse() //把当前字符串反转
66 {
67     Mystring res;
68     res.length = length;
69     res.p_str = new char[length + 1];
70     for (int i = 0; i < length; ++i)
71         res.p_str[i] = p_str[length - 1 - i];
72     res.p_str[length] = '\0';
73     strcpy(p_str, res.p_str);
74 }
75
76 int Mystring::append(Mystring & s) //把字符串s连接到当前字符串结尾
77 {
78     int len = length + s.length;
79     Mystring res;
80     res.length = len;
81     res.p_str = new char[len + 1];
82     for (int i = 0; i < length; ++i)
83         res.p_str[i] = p_str[i];
84     for (int i = length; i < len; ++i)
85         res.p_str[i] = s.p_str[i - length];
86     res.p_str[len] = '\0';
87
88     if(p_str)
89         delete []p_str;
90     length = len;
91     p_str = new char[len + 1];
92     strcpy(p_str, res.p_str);
93     return 1;
94 }
95
96 int Mystring::append(Mystring & s, int n) //把字符串s的前n个字符连接到当前字符串结尾
97 {
98     int len = length + n;
99     Mystring res;
100     res.length = len;
101     res.p_str = new char[len + 1];
102     for (int i = 0; i < length; ++i)
```



```
103     res.p_str[i] = p_str[i];
104     for (int i = length; i < len; ++i)
105         res.p_str[i] = s.p_str[i - length];
106     res.p_str[len] = '\0';
107
108     if(p_str)
109         delete []p_str;
110     length = len;
111     p_str = new char[len + 1];
112     strcpy(p_str, res.p_str);
113     return 1;
114 }
115
116 //比较当前字符串和s的大小，返回值为 1 表示当前字符串大于给定字符串 s，为 0 相等，为 -1 小于
117 int Mystring::compare(const Mystring & s)
118 {
119     int len = min(length, s.length);
120     for (int i = 0; i < len; ++i)
121     {
122         if (p_str[i] != s.p_str[i])
123             return p_str[i] > s.p_str[i];
124     }
125     return length > s.length;
126 }
127 }
128
129 //在当前字符串第 pos 位置前插入字符串 s
130 int Mystring::insert(Mystring & s, int pos)
131 {
132     if (pos < 0 || pos > length)
133     {
134         cout << "out of range\n";
135         return 0;
136     }
137     int len = length + s.length;
138     Mystring res;
139     res.length = len;
140     res.p_str = new char[len + 1];
141     for (int i = 0; i < pos; ++i)
142         res.p_str[i] = p_str[i];
143     for(int i = pos; i < pos + s.length; ++i)
144         res.p_str[i] = s.p_str[i-pos];
145     for(int i = pos+s.length; i < len; ++i)
146         res.p_str[i] = p_str[i - s.length];
147     res.p_str[len] = '\0';
```



```
148
149     if(p_str)
150         delete []p_str;
151     length = len;
152     p_str = new char[len + 1];
153     strcpy(p_str, res.p_str);
154     return 1;
155 }
156
157 int Mystring::erase(int pos, int n) //删掉当前字符串pos开始, 长度为 n 的字符
158 {
159     if(pos < 0 || pos+n > length)
160     {
161         cout << "out of range\n";
162         return 0;
163     }
164     int len = length - n;
165     Mystring res;
166     res.length = len;
167     res.p_str = new char[len + 1];
168     for (int i = 0; i < pos; ++i)
169         res.p_str[i] = p_str[i];
170     for (int i = pos + n; i < length; ++i)
171         res.p_str[i - n] = p_str[i];
172     res.p_str[len] = '\0';
173
174     if(p_str)
175         delete []p_str;
176     length = len;
177     p_str = new char[len + 1];
178     strcpy(p_str, res.p_str);
179     return 1;
180 }
181
182 int Mystring::find(const char c, int pos) //从pos开始查找字符c在当前字符串中第一次出现的位置
183 {
184     if (c == NULL || pos >= length)
185         return npos;
186     for (int i = pos; i < length; ++i)
187         if (p_str[i] == c)
188             return i;
189     return npos;
190 }
191
192 int Mystring::find(const Mystring & s, int pos) //从pos开始查找字符串s在当前串中第一次出现的位
```



置

```
193 {
194     int len = s.length;
195     if (s.p_str == NULL || pos < 0 || pos+len >= length)
196         return npos;
197     for (int i = pos; i < length-len; ++i)
198     {
199         if (p_str[i] == s.p_str[0])
200         {
201             int j = 0, t = i;
202             while (j < len && p_str[t] == s.p_str[j])
203                 ++t, ++j;
204             if (j == len)
205                 return i;
206         }
207     }
208     return npos;
209 }
210
211
212
213 int main()
214 {
215     Mystring s1("You never know your luck.");
216     Mystring s2("Sow nothing, reap nothing.");
217
218     if (!s1.empty())
219         cout << "s1 != empty\n\n";
220     cout << "s1.size = " << s1.size() << endl << endl;
221
222     s1.reverse();
223     cout << "after reverse, s1 = " << endl << s1.display() << endl << endl;
224
225     s1.reverse();
226     s1.append(s2);
227     cout << "after reverse again and append s2, s1 = " << endl << s1.display() << endl <<
endl;
228
229     s2.append(s1, 3);
230     cout << "after append the first three characters of s1, s2 = " << endl << s2.display()
<< endl << endl;
231
232     if(s2.erase(25, 3))
233         cout << "after erase, s2 = " << endl << s2.display() << endl << endl;
234     if (s1.erase(24, s2.size()))
```



```
235     cout << "after erase, s1 = " << endl << s1.display() << endl << endl;
236
237     if (s1.compare(s2) == 1)
238         cout << "s1 > s2\n" << endl;
239     else if(s1.compare(s2) == -1)
240         cout << "s1 < s2\n" << endl;
241     else
242         cout << "s1 = s2\n" << endl;
243
244     int p1 = s1.find('k');
245     cout << "in s1, k's location of the first occurrence is " << endl << p1 << endl <<
endl;
246
247     int p2 = s1.find("luck");
248     cout << "in s1, luck's location of the first occurrence is " << endl << p2 << endl <<
endl;
249
250     return 0;
251 }
```

【编程题 3】请设计一个三角形类 Triangle，其中三角形信息包括：各边边长；三角形类对外能提供的接口功能包括：判断能否构成三角形、输出三角形周长、输出三角形面积、输出三角形类型（直角，锐角，还是钝角三角形）。

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <iostream>
03 #include <cstring>
04 #include <cmath>
05 #include <algorithm>
06 using namespace std;
07
08 const double eps = 1e-6;
09
10 class Triangle
11 {
12 private:
13     double a, b, c, s; //各边边长，面积
14
15 public:
16     Triangle(double x = 3, double y = 4, double z = 5);
17     ~Triangle();
18     bool judge_tri(); //判断能否构成三角形
19     double C(); //输出三角形周长
20     double S(); //输出三角形面积
21     void judge_type(); //输出三角形类型（直角，锐角，还是钝角三角形）
```



```
22 };
23
24 Triangle::Triangle(double x, double y, double z)//判断能否构成三角形
25 {
26     a = x;
27     b = y;
28     c = z;
29 }
30
31 Triangle::~~Triangle()
32 {
33 }
34
35 bool Triangle::judge_tri() //输出三角形周长
36 {
37     if (a + b > c && a + c > b && b + c > a)
38         return 1;
39     return 0;
40 }
41
42 double Triangle::C() //输出三角形面积
43 {
44     if (!judge_tri())
45     {
46         cout << "No triangle!\n\n";
47         return 0;
48     }
49     cout << "The circumference of the triangle is " << a + b + c << endl << endl;
50
51     return a + b + c;
52 }
53
54 double Triangle::S() //输出三角形面积
55 {
56     if (!judge_tri())
57     {
58         cout << "No triangle!\n\n";
59         return 0;
60     }
61     double p = (a + b + c) / 2.0;
62     s = sqrt(p*(p - a)*(p - b)*(p - c));
63     cout << "The area of the triangle is " << s << endl << endl;
64
65     return s;
66 }
```





```
67
68 void Triangle::judge_type() //输出三角形类型 (直角, 锐角, 还是钝角三角形)
69 {
70     if (!judge_tri())
71     {
72         cout << "No triangle!\n\n";
73         return;
74     }
75     double mn = min(a, min(b, c)), mx = max(a, max(b, c));
76     double md = a + b + c - mn - mx;
77     if (fabs(mn * mn + md * md) - mx * mx < eps)
78     {
79         cout << "The type is right-angled triangle!\n\n";
80     }
81     else if((mn * mn + md * md) < mx * mx)
82     {
83         cout << "The type is acute triangle!\n\n";
84     }
85     else
86     {
87         cout << "The type is obtuse triangle!\n\n";
88     }
89 }
90
91 int main()
92 {
93     Triangle t(6, 8, 10);
94     if (t.judge_tri())
95         cout << "It can form a triangle!\n\n";
96     else
97         cout << "It can't form a triangle!\n\n";
98     t.C();
99     t.S();
100     t.judge_type();
101
102     return 0;
103 }
```

【编程题 4】请设计一个学生类 Student，其中学生的信息包括：姓名、性别、学号、出生日期、家庭住址、6 个科目的考试成绩。该类对外提供的函数接口功能包括：

构造函数：初始化学生类对象（无参构造，有参构造，拷贝构造，至少 3 种）

普通函数：获取学生任意信息，修改学生成绩信息，输出学生信息

析构函数：释放学生类对象内存空间



然后基于学生类回答下列 2 个问题：

1) 分析学生类对象及其成员变量的存储规律（可借助对象或成员变量的地址进行分析）。

2) 阐述普通构造函数（传值初始化新对象），拷贝构造函数（对象初始化对象）在初始化对象过程中的联系与区别。

Student.h

```
01 #ifndef _STUDENT
02 #define _STUDENT
03 #include <iostream>
04 #include <cstring>
05 #include <cmath>
06 #include <algorithm>
07 #define SIZE 80
08 using namespace std;
09
10
11 class Student
12 {
13 private:
14     char *name; //姓名
15     char sex; //性别
16     char ID[19], birth[10], addr[56]; //学号, 出生日期, 家庭住址
17     double Chinese, Math, English, Physics, Chemistry, Biology; //6个科目的考试成绩
18 public:
19     Student(); //无参构造
20     Student(char *na, char sx, char *id, char *bir, char*addr, double Chin, double M, double
E, double P, double Chem, double B); // 有参构造
21     Student(const Student &per); //拷贝构造
22     ~Student(); //析构函数
23     char *Get_name(); //提取姓名
24     char Get_sex(); //提取性别
25     char *Get_ID(); //提取学号
26     char *Get_birth(); //提取出生日期
27     char *Get_addr(); //提取家庭住址
28     double Get_Chinese(); //提取语文成绩
29     double Get_Math(); //提取数学成绩
30     double Get_English(); //提取英语成绩
31     double Get_Physics(); //提取物理成绩
32     double Get_Chemistry(); //提取化学成绩
33     double Get_Biology(); //提取生物成绩
34     void Change(int num, double det); //修改学生成绩信息, 各科编号分别对应数字 1,2,3,4,5,6
35     void Display(); //显示学生信息
36     void Input(); //输入学生信息
37 };
```



```
38
39 #endif
```

Student.cpp

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include "student.h"
03
04 Student::Student()
05 {
06     name = NULL;
07     sex = 'M';
08     Chinese = 0;
09     Math = 0;
10     English = 0;
11     Physics = 0;
12     Chemistry = 0;
13     Biology = 0;
14 }
15
16 Student::Student(char * na, char sx, char * id, char * bir, char * ad, double Chin, double
M, double E, double P, double Chem, double B)// 有参构造
17 {
18     if (na)
19     {
20         name = new char[strlen(na) + 1];
21         strcpy(name, na);
22     }
23     sex = sx;
24     strcpy(ID, id);
25     strcpy(birth, bir);
26     strcpy(addr, ad);
27     Chinese = Chin;
28     Math = M;
29     English = E;
30     Physics = P;
31     Chemistry = Chem;
32     Biology = B;
33 }
34
35 Student::Student(const Student & per) //拷贝构造
36 {
37     if (per.name)
38     {
39         name = new char[strlen(per.name) + 1];
40         strcpy(name, per.name);
```



```
41     }
42     sex = per.sex;
43     strcpy(ID, per.ID);
44     strcpy(birth, per.birth);
45     strcpy(addr, per.addr);
46     Chinese = per.Chinese;
47     Math = per.Math;
48     English = per.English;
49     Physics = per.Physics;
50     Chemistry = per.Chemistry;
51     Biology = per.Biology;
52 }
53
54 Student::~Student() //析构函数
55 {
56     if (name)
57         delete[] name;
58 }
59
60 char * Student::Get_name()//提取姓名
61 {
62     return name;
63 }
64
65 char Student::Get_sex() //提取性别
66 {
67     return sex;
68 }
69
70 char * Student::Get_ID() //提取学号
71 {
72     return ID;
73 }
74
75 char * Student::Get_birth() //提取出生日期
76 {
77     return birth;
78 }
79
80 char * Student::Get_addr() //提取家庭住址
81 {
82     return addr;
83 }
84
85 double Student::Get_Chinese() //提取语文成绩
```



```
86 {
87     return Chinese;
88 }
89
90 double Student::Get_Math() //提取数学成绩
91 {
92     return Math;
93 }
94
95 double Student::Get_English() //提取英语成绩
96 {
97     return English;
98 }
99
100 double Student::Get_Physics() //提取物理成绩
101 {
102     return Physics;
103 }
104
105 double Student::Get_Chemistry() //提取化学成绩
106 {
107     return Chemistry;
108 }
109
110 double Student::Get_Biology() //提取生物成绩
111 {
112     return Biology;
113 }
114
115 void Student::Change(int num, double det) //修改学生成绩信息,各科编号分别对应数字 1,2,3,4,5,6
116 {
117     switch (num)
118     {
119     case 1:
120         Chinese = det;
121         break;
122     case 2:
123         Math = det;
124         break;
125     case 3:
126         English = det;
127         break;
128     case 4:
129         Physics = det;
130         break;
```



```
131     case 5:
132         Chemistry = det;
133         break;
134     case 6:
135         Biology = det;
136         break;
137     }
138 }
139
140 void Student::Display() //显示学生信息
141 {
142     cout << " 姓 名 : " << name << endl;
143     cout << " 性 别 : " << sex << endl;
144     cout << " 学 号 : " << ID << endl;
145     cout << "出生日期: " << birth << endl;
146     cout << "家庭住址: " << addr << endl;
147     cout << "语文成绩: " << Chinese << endl;
148     cout << "数学成绩: " << Math << endl;
149     cout << "英语成绩: " << English << endl;
150     cout << "物理成绩: " << Physics << endl;
151     cout << "化学成绩: " << Chemistry << endl;
152     cout << "生物成绩: " << Biology << endl << endl;
153 }
154
155 void Student::Input() //输入学生信息
156 {
157     char na[10];
158     cout << "输入 姓 名 : ";
159     cin >> na;
160     if (name)
161         delete[]name;
162     name = new char[strlen(na) + 1];
163     strcpy(name, na);
164     cout << "输入 性 别 : ";
165     cin >> sex;
166     cout << "输入 学 号 : ";
167     cin >> ID;
168     cout << "输入出生日期: ";
169     cin >> birth;
170     cout << "输入家庭住址: ";
171     cin >> addr;
172     cout << "输入语文成绩: ";
173     cin >> Chinese;
174     cout << "输入数学成绩: ";
175     cin >> Math;
```



```
176     cout << "输入英语成绩: ";
177     cin >> English;
178     cout << "输入物理成绩: ";
179     cin >> Physics;
180     cout << "输入化学成绩: ";
181     cin >> Chemistry;
182     cout << "输入生物成绩: ";
183     cin >> Biology;
184
```

Main.cpp

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <iostream>
03 #include "student.h"
04 #include <cstring>
05 #include <cmath>
06 #include <algorithm>
07 using namespace std;
08
09 const int N = 16;
10
11 int cnt = 0;
12
13 void menu() //定义菜单函数
14 {
15     cout << "*****1.录入信息*****" << endl;
16     cout << "*****2.查询信息*****" << endl;
17     cout << "*****3.浏览信息*****" << endl;
18     cout << "*****4.修改信息*****" << endl;
19     cout << "*****0.退出*****" << endl;
20 }
21
22 void Output_stu(Student *array) //输出对象数组元素
23 {
24     cout << "学生总人数 = " << cnt << endl << endl;
25     for (int i = 0; i < cnt; ++i)
26         array[i].Display();
27 }
28
29 void Input_stu(Student *array) //输入对象数组元素
30 {
31     char ch;
32     do
33     {
34         array[cnt].Input();
```



```
35     ++cnt;
36     cout << "继续输入吗? (Y or N) " << endl << endl;
37     cin >> ch;
38 }
39 while (ch == 'Y');
40 }
41
42 int Search_stu(Student *array, char *na) //按姓名查询
43 {
44     for (int i = 0; i < cnt; ++i)
45         if (strcmp(array[i].Get_name(), na) == 0)
46             return i;
47 }
48
49 int main()
50 {
51     Student array[N];
52     int choice, i;
53     do
54     {
55         menu();
56         cout << "please input you choice: ";
57         cin >> choice;
58         if (choice >= 0 && choice <= 4)
59         {
60             switch (choice)
61             {
62             case 1:
63                 Input_stu(array);
64                 break;
65             case 2:
66                 cout << "Input the name searched" << endl;
67                 char na[20];
68                 cin >> na;
69                 i = Search_stu(array, na);
70                 if (i == N)
71                     cout << "查无此人" << endl << endl;
72                 else
73                     array[i].Display();
74                 break;
75             case 3:
76                 Output_stu(array);
77                 break;
78             case 4:
79                 cout << "Input the name, subject number, new grade: " << endl;
```





```
80         int num;
81         double grade;
82         cin >> na >> num >> grade;
83         i = Search_stu(array, na);
84         if (i == N)
85             cout << "查无此人" << endl << endl;
86         else
87             array[i].Change(num, grade);
88         cout << endl;
89         break;
90     }
91 }
92 }
93 while (choice);
94
95 return 0;
96 }
97
```

【编程题 5】请设计一个点类 Point，类内包含点的二维坐标数值，通过该类实现空间点坐标的初始化，坐标修改，坐标位置输出功能；然后基于该 Point 类设计一个友元函数和一个友元类，实现计算两个点之间的直线距离。

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <iostream>
03 #include "student.h"
04 #include <cstring>
05 #include <cmath>
06 #include <algorithm>
07 using namespace std;
08
09 class Another;
10
11 class Point
12 {
13 private:
14     double x, y; //横、纵坐标
15     char nm[6] = "Alice"; //这个点的名字
16
17 public:
18     Point(double a = 0, double b = 0); //构造函数
19     ~Point(); //析构函数
20     void Modify(double a = 0, double b = 0); //修改坐标
21     void Output(); //输出坐标
22     friend double distance1(Point&, Point&); //函数 distance1 作为类 Point 的友元函数，求a、
```



b之间的距离

```
23     friend Another; //类 Another 作为类 Point 的友元类
24 };
25
26 class Another
27 {
28 private:
29     char *name;
30     double u, v;
31
32 public:
33     Another(); //构造函数
34     Another(const char *na, double a, double b); //构造函数
35     ~Another(); //析构函数
36     double distance2(Point &); //求距离
37 };
38
39 Point::Point(double a, double b)
40 {
41     x = a;
42     y = b;
43 }
44
45 Point::~~Point()
46 {
47 }
48
49 void Point::Modify(double a, double b) //修改坐标
50 {
51     x = a;
52     y = b;
53 }
54
55 void Point::Output() //输出坐标
56 {
57     cout << "坐标为 : (" << x << ', ' << y << ')' << endl << endl;
58 }
59
60 double distance1(Point &a, Point &b) //求a、b之间的距离
61 {
62     return sqrt((a.x-b.x)*(a.x-b.x)+(a.y-b.y)*(a.y-b.y));
63 }
64
65 Another::Another()
66 {
```



```
67     name = NULL;
68 }
69
70 Another::Another(const char * na, double a, double b)
71 {
72     if (na)
73     {
74         name = new char[strlen(na) + 1];
75         strcpy(name, na);
76     }
77     u = a;
78     v = b;
79 }
80
81 Another::~~Another()
82 {
83     if (name)
84         delete[] name;
85 }
86
87 double Another::distance2(Point &obj)//求当前点与 obj 之间的距离
88 {
89     cout << "name1 = " << obj.nm << ", x = " << obj.x << ", y = " << obj.y << endl;
90     cout << "name2 = " << name << ", x = " << u << ", y = " << v << endl;
91     double dis = sqrt((obj.x - u)*(obj.x - u) + (obj.y - v)*(obj.y - v));
92     cout << "distance is = " << dis << endl << endl;
93     return dis;
94 }
95
96
97 int main()
98 {
99     Point a(3, 4);
100     Point b(9, 25);
101     Another c("Cindy", 1.0, 1.0);
102     double dis = distance1(a, b);
103     cout << "the distance between a and b is: " << dis << endl << endl;
104     c.distance2(a);
105     return 0;
106 }
```