

《C++ 程序设计》OJ1 (参考答案)

VCG

2023 年 5 月 27 日



1 圆

【描述】

编写一个圆类 Circle，实现半径的输入、面积的计算和输出

【输入描述】

输入一行，输入圆的半径 (double 类型)。

【输出描述】

输出一行，输出圆的面积 (保留小数点后两位)

【输入样例】

3

【输出样例】

28.27

【提示】

- 1、在输出面积前使用如下语句：`cout<<setiosflags(ios::fixed)<<setprecision(2);` 来设置输出格式，因此需要包含 `iomanip` 头文件
- 2、使用 `acos(-1.0)` 来得到圆周率的值，可将其设置为常量，因此：`const double PI=acos (-1.0);` 需要包含 `cmath` 头文件

main.cpp

```
1 #include<iostream>
2 #include<cmath>
3 #include<iomanip>
4
5 const double PI = acos(-1);
6
7 class Circle
8 {
9 private:
10     double r;
11 public:
12     Circle() { r = 0; }
13     Circle(double r) { this->r = r; }
14     double Area() { return PI*r*r; }
15 };
16
17 int main()
18 {
19     double r;
20     std::cin >> r;
21     Circle circle(r);
22
23     std::cout << std::setiosflags(std::ios::fixed) << std::setprecision(2);
24     std::cout << circle.Area() << std::endl;
25
26     return 0;
27 }
```

2 图书类

【描述】

构造函数，析构函数练习。

以下是图书类 Book 的声明，请完成 Book 类，并根据输入 (name,author, sale) 使用不同的构造函数创建对象，并显示信息。【注意，下述代码默认存在，提交代码时不能包括以下代码】

```
1 class Book
2 {
3 private:
4     char name[256];    //书名
5     char author[256]; //作者
6     int sale;          //销售量
7 public:
8     Book(); //无参构造函数
9     Book(const char * _name, const char * _author, int _sale); //有参构造函数
10    Book(const Book &x); //拷贝构造函数
11    void Print(); //显示数据
12    ~Book(); //析构函数
13 };
14
15 int main()
16 {
17     char name[256], author[256];
18     std::cin.getline(name, sizeof(name));
19     std::cin.getline(author, sizeof(author));
20
21     int sale;
22     std::cin >> sale;
23     if (strcmp(name, "-1") == 0 && strcmp(author, "-1") == 0 && sale == -1)
24     {
25         Book bk;
26         bk.Print();
27     }
28     else if (strcmp(name, "0") == 0 && strcmp(author, "0") == 0 && sale == 0)
29     {
30         Book bk = Book();
31         bk.Print();
32     }
33     else
34     {
35         Book bk(name, author, sale);
36         bk.Print();
37     }
38
39     return 0;
40 }
```

【输入描述】

输入三行数据。对应成员变量 (name, author, sale)

第一行书的名称 (长度不超过 100, 可能含有空格)

第二行作者名字 (长度不超过 100, 可能含有空格)

第三行销量 (整数类型)

根据输入 (name, author, sale) 使用不同的构造函数创建一个 Book 对象, 并显示信息:

(1) (" -1", "-1", -1) 时, 使用无参构造函数创建对象, name 默认值 "No name", author 默认值 "No author", sal 的默认值 0.

(2) ("0", "0", 0) 时, 使用拷贝构造函数创建对象, 大致为:

```
Book bk = Book();
```

```
bk.print();
```

(3) 其他, 对应带参数构造函数创建对象。

【输出描述】

使用类中的 void print() 方法来输出一定格式的字符串, 详见样例。

【输入样例】

```
The Art of Computer Programming
```

```
Donald Ervin Knuth
```

```
1000
```

【输出样例】

```
Name: The Art of Computer Programming, Author: Donald Ervin Knuth, Sale: 1000
```

【提示】

- 1、输出每个图书的信息占一行, 项目之间", " 分隔, 冒号后面都有一个空格, 最后换行。
- 2、字符串输入空格, 类似 cin.getline(name, sizeof(name))
- 3、比较两个字符串是否相等, 请用 strcmp(s1, s2) 函数。例如 strcmp(name, "-1"), 如果 name 是 "-1", 则返回值为 0.
- 4、必须要用类 (class) 来实现代码, 否则不得分

消除 Error C4996 'strcpy': 错误, 请在程序第一行加上:

```
#define _CRT_SECURE_NO_WARNINGS
```

main.cpp

```

1  #define _CRT_SECURE_NO_WARNINGS
2  #include <iostream>
3  #include <cstring>
4
5  class Book
6  {
7  private:
8      char name[256];    //书名
9      char author[256];  //作者
10     int sale;           //销售量
11 public:
12     Book(); //无参构造函数
13     Book(const char *_name, const char *_author, int _sale); //有参构造函数
14     Book(const Book& x); //拷贝构造函数
15     void Print(); //显示数据
16     ~Book(); //析构函数
17 };
18
19 Book::Book()
20 {
21     strcpy(name, "No name");
22     strcpy(author, "No author");
23     sale = 0;
24 }
25
26 Book::Book(const char *_name, const char *_author, int _sale)
27 {
28     strcpy(name, _name);
29     strcpy(author, _author);
30     sale = _sale;
31 }
32
33 Book::Book(const Book& x)
34 {
35     strcpy(name, x.name);
36     strcpy(author, x.author);
37     sale = x.sale;
38 }
39
40 Book::~Book()
41 {
42 }
43
44 void Book::Print()
45 {
46     std::cout << "Name: " << name << ", " << "Author: " << author << ", " << "Sale: "
47         << sale << std::endl;

```

```
48
49 int main()
50 {
51     char name[256], author[256];
52     std::cin.getline(name, sizeof(name));
53     std::cin.getline(author, sizeof(author));
54
55     int sale;
56     std::cin >> sale;
57     if (strcmp(name, "-1") == 0 && strcmp(author, "-1") == 0 && sale == -1)
58     {
59         Book bk;
60         bk.Print();
61     }
62     else if (strcmp(name, "0") == 0 && strcmp(author, "0") == 0 && sale == 0)
63     {
64         Book bk = Book();
65         bk.Print();
66     }
67     else
68     {
69         Book bk(name, author, sale);
70         bk.Print();
71     }
72
73     return 0;
74 }
```

3 过道栅栏

【描述】

现在需在一矩形体育场周围建一矩形过道，并在四周围安上栅栏。栅栏价格为 50 元/米，过道造价为 240 元/平方米。过道宽为 1.5 米，体育场的长宽由键盘输入。

体育场和矩形过道都是如下 Rectangle 的对象，请实现以下 Rectagnle 类，编写 main 函数计算并输出过道和栅栏的造价。

```
1 class Rectangle
2 {
3     private:
4         double length;
5         double width;
6     public:
7         Rectangle(double Length = 10., double Width = 5.);
8         double Area();
9         double Perimeter();
10 };
```

【输入描述】

体育场的长和宽

【输出描述】

输出 2 行。

第一行是栅栏的造价。

第二行是过道的造价

【输入样例】

2.4 1.2

【输出样例】

960

4752

【提示】

体育场用对象 rect1(x, y) 来表示的话，过道外的矩形就用对象 rect2(x + 3, y + 3) 表示

main.cpp

```
1 #include<iostream>
2
3 class Rectangle
4 {
5 private:
6     double length;
7     double width;
8 public:
9     Rectangle(double Length = 10., double Width = 5.);
10    double Area();
11    double Perimeter();
12 };
13
14 Rectangle::Rectangle(double Length, double Width)
15 {
16     length = Length;
17     width = Width;
18 }
19
20 double Rectangle::Area()
21 {
22     return length * width;
23 }
24
25 double Rectangle::Perimeter()
26 {
27     return 2 * (length + width);
28 }
29
30 int main()
31 {
32     double price = 50.0;
33     double x, y;
34     std::cin >> x >> y;
35     Rectangle rect1(x, y), rect2(x + 3, y + 3);
36     std::cout << rect2.Perimeter()*price << std::endl;
37     double area12;
38     area12 = rect2.Area() - rect1.Area();
39     price = 240.0;
40     std::cout << area12 * price << std::endl;
41     return 0;
42 }
```

4 评分程序

【描述】

为一门课写一个评分程序，评分原则如下：

- (1) 有两次随堂考试，每次满分 50 分；
 - (2) 有一次期中考试和一次期末考试，每次满分 100 分；
 - (3) 期末考试占总评成绩的 50%，期中考试占总评成绩的 25%，两次随堂考试总共占 25%；
 - (4) 总评成绩 90 100 分为 A，80 89 分为 B，70 79 分为 C，60 69 分为 D，低于 60 分为 E；
- 设计一个 Score 类，数据成员如下：

```
1 string name; //记录学生姓名
2 double s[4]; //存储4次成绩，s[0]和s[1]存储2次随堂考试，s[2]存储期中考试，s[3]存储期
   末考试
3 double total; //记录总评成绩
4 char grade; //记录对应的等级
```

学生信息由键盘录入，默认总评成绩的等级为 B，其他数据项无默认值。计算总评成绩并给出等级，输出某个同学的全部信息。

主函数如下：

```
1 int main()
2 {
3     Score *s1=new Score;
4     s1->Input();
5     s1->Evaluate();
6     s1->Output();
7     delete s1;
8     return 0;
9 }
```

【输入描述】

输入 5 行。

第 1 行是学生姓名

第 2 和 3 行是两次随堂考试成绩

第 4 行是期中考试成绩

第 5 行是期末考试成绩

【输出描述】

如果输入成绩超出了范围，输出：error

否则显示：姓名，总分和等级

【输入样例】

```
Mary
50
50
```

90

100

【输出样例】

name: Mary, total: 97.5, grade: A

【输入样例 2】

Jack

90

90

100

100

【输出样例 2】

error

【提示】

冒号和逗号后各有一个空格



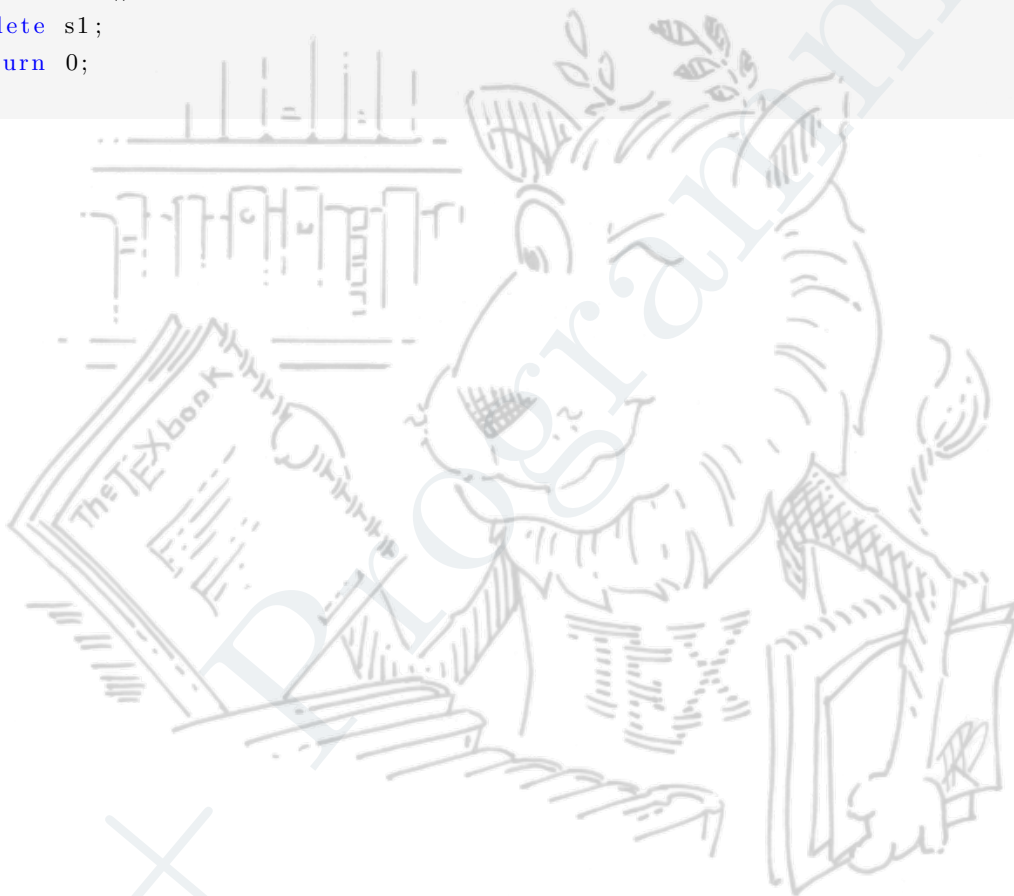
main.cpp

```

1  #include<iostream>
2  #include<string>
3
4  class Score
5  {
6  private:
7      std::string name;//学生姓名
8      double s[4]; //s[0],s[1] 随堂考试, s[2] 期中考试, s[3] 期末考试
9      double total;//总评成绩
10     char grade;//等级
11 public:
12     Score()
13     {
14         grade = 'B';
15     }
16     void Input()
17     {
18         std::cin >> name;
19         for (int i = 0; i < 4; i++)
20             std::cin >> s[i];
21         if (s[0] < 0 || s[0]>50 ||
22             s[1] < 0 || s[1]>50 ||
23             s[2] < 0 || s[2]>100 ||
24             s[3] < 0 || s[3]>100)
25         {
26             std::cout << "error" << std::endl;
27             grade = 'W';
28         }
29     }
30     void Evaluate()
31     {
32         if (grade == 'W')
33             return;
34         total = s[3] * 0.5 + s[2] * 0.25 + (s[1] + s[0])*0.25;
35         if (total >= 90 && total <= 100)
36             grade = 'A';
37         else if (total >= 80)
38             grade = 'B';
39         else if (total <= 79 && total >= 70)
40             grade = 'C';
41         else if (total <= 69 && total >= 60)
42             grade = 'D';
43         else if (total < 60)
44             grade = 'E';
45         else
46             grade = 'W';
47     }
48     void Output()

```

```
49 {  
50     if (grade != 'W')  
51         std::cout << "name: " << name << ", total: " << total << ", grade: " <<  
52         grade << std::endl;  
53 };  
54  
55 int main()  
56 {  
57     Score* s1 = new Score;  
58     s1->Input();  
59     s1->Evaluate();  
60     s1->Output();  
61     delete s1;  
62     return 0;  
63 }
```



5 Hotel

【描述】

编写程序，统计某旅馆住宿客人的总数。要求输入客人的姓名，输出客人的编号（按先后顺序自动生成）、姓名以及总人数。

使用如下 main 函数对程序进行测试

```

1  int main()
2  {
3      Hotel h[100];
4      h[0].add("Susan");
5      h[1].add("Peter");
6      h[2].add("John");
7      h[3].add("Mary");
8      h[4].add("Alice");
9
10     string name;
11     cin>>name;
12
13     for(int i=0;i<Hotel::getTotal();i++)
14     {
15         if(h[i].getName()==name)
16         {
17             h[i].print();
18             break;
19         }
20     }
21     return 0;
22 }
```

【输入描述】

输入一行，输入客人的姓名（不超过 100 个字符的由英文大小写字母组成的字符串）。

【输出描述】

输出一行，输出客人的编号，姓名及总人数，空格分隔。

【输入样例】

Peter

【输出样例】

2 Peter 5

【提示】

请注意，必须要用类（class）来实现代码，否则不得分。

main.cpp

```

1  #include<iostream>
2  #include<string>
3  #include<vector>
4  using namespace std;
5
6  class Hotel
7  {
8  private:
9      static int total;
10     std::string name;
11     int id;
12 public:
13     Hotel();
14     void print() const;
15     void add(const std::string& name);
16     static int getTotal() { return total; }
17     string getName() const;
18 };
19
20 Hotel::Hotel() {}
21
22 void Hotel::add(const std::string& name) {
23     this->name = name;
24     total++;
25     this->id = total;
26 }
27
28 string Hotel::getName() const
29 {
30     return name;
31 }
32
33 void Hotel::print() const
34 {
35     cout << id << " " << name << " " << getTotal() << endl;
36 }
37
38 int Hotel::total = 0;
39
40 int main()
41 {
42     Hotel h[100];
43     h[0].add("Susan");
44     h[1].add("Peter");
45     h[2].add("John");
46     h[3].add("Mary");
47     h[4].add("Alice");
48

```

```

49     string name;
50     cin >> name;
51     for (int i = 0; i < Hotel::getTotal(); i++)
52     {
53         if (h[i].getName() == name)
54         {
55             h[i].print();
56             break;
57         }
58     }
59     return 0;
60 }
61
62
63 /*
64 题目是不恰当的
65 Hotel h[100]这种定义之后，再去统计入住人数，static数据成员没有必要
66 每次调用add，可以直接加1。
67 h[1].add这种写法不对，应该类似 h[1].set("Mary")
68 总之，就不该用数组来add
69 而应该单独定义一个hotel，其数据成员为客人的数组
70 正解如下
71 */
72 class Guest
73 {
74 private:
75     static int count; //创建了多少个Guest
76     std::string name;
77     int id;
78 public:
79     Guest(const std::string& name)
80     {
81         this->name = name;
82         count++;
83         this->id = count;
84     }
85     void Print() const { std::cout << id << " " << name << " " << Count() << std::
        endl; }
86
87     static int Count(){ return count;}
88     std::string GetName() const { return name;}
89 };
90 int Guest::count=0;
91
92 class MHotel
93 {
94 private:
95     std::vector<Guest> guests;
96 public:
97     MHotel() {};
```

```
98
99 void Insert(const std::string& name)
100 {
101     guests.push_back(Guest(name));
102 }
103
104 bool Find(const std::string& name)
105 {
106     for (int i = 0; i < guests.size(); i++)
107     {
108         if (guests[i].GetName() == name)
109         {
110             guests[i].Print();
111             return true;
112         }
113     }
114     return false;
115 }
116 };
117
118 void Test()
119 {
120     MHotel hotel;
121     hotel.Insert("Susan");
122     hotel.Insert("Peter");
123     hotel.Insert("John");
124     hotel.Insert("Mary");
125     hotel.Insert("Alice");
126
127     std::string name;
128     std::cin >> name;
129     hotel.Find(name);
130 }
```