

实验报告

指导老师：陈家骏 黄书剑

姓名：王晨渊, 学号：181220057

1 概念题

1.1 C++ 中构造函数和析构函数的作用分别是什么？它们分别在什么时候会被调用？

构造函数的作用是初始化对象并调用成员对象类和基类的构造函数；仅在创建对象时被调用。

析构函数的作用是释放该对象申请的额外资源并调用成员对象类和基类的析构函数；显式地在程序中被调用或在对象消亡时被隐式的调用。

1.2 成员对象初始化和消亡处理的次序是怎样的？

按类中定义成员对象的顺序进行初始化，按类中定义成员对象的逆序进行消亡。

1.3 什么是拷贝构造函数？在哪些情况下，系统会调用类的拷贝构造函数？

拷贝构造函数是在类中按原型 `< 类名 >(const < 类名 >&);` 定义的函数。在定义对象即用同类对象进行初始化时，在将对象作为值参数传给函数时，和把对象作为返回值时，系统会调用拷贝构造函数。

2 编程题

2.1 固定高度箱体类

```
1  class BoxWithFixedHeight
2  {
3      public:
4          BoxWithFixedHeight(): fixed_height(10), len(5), wid(5) // 默认尺寸
5          {
6          }
7          BoxWithFixedHeight(double l, double w, double h)
8          {
9              len = l;
10             wid = w;
11             fixed_height = h;
```

```

12     }
13     double get_volume()
14     {
15         return fixed_height*len*wid;
16     }
17     void set_len(double length)
18     {
19         len = length;
20     }
21
22     private:
23         double fixed_height, len, wid;
24 };

```

2.2 矩阵类

```

1     class Matrix
2     {
3     public:
4         Matrix(int row, int col)
5         {
6             matrix_col = col;
7             matrix_row = row;
8             matrix = new int*[row];
9             for(int i=0;i<row;i++)
10            {
11                matrix[i] = new int[col];
12            }
13        }
14        ~Matrix()
15        {
16            for(int i=0;i<matrix_row;i++)
17            {
18                delete[] matrix[i];
19            }
20            delete[] matrix;
21        }
22
23        void input(int* a, int length)

```

```

24 {
25     for(int count=0,i=0;i<matrix_row;i++)
26     {
27         for(int j=0;j<matrix_col;j++,count++)
28         {
29             if(count < length)
30             {
31                 matrix[i][j] = a[count];
32             }
33             else
34             {
35                 matrix[i][j] = 0;
36             }
37         }
38     }
39 }
40
41 void print()
42 {
43     for(int i=0;i<matrix_row;i++)
44     {
45         for(int j=0; j<matrix_col;j++)
46         {
47             cout<<matrix[i][j]<<" ";
48         }
49         cout<<endl;
50     }
51 }
52 void transpose()
53 {
54     for(int i=0;i<matrix_row-1;i++)
55     {
56         for(int j=i+1;j<matrix_col;j++)
57         {
58             int temp = matrix[i][j];
59             matrix[i][j] = matrix[j][i];
60             matrix[j][i] = temp;
61         }
62     }

```

```
63         }  
64  
65     private:  
66         int matrix_col, matrix_row;  
67         int** matrix;  
68     };
```