

计算机学院高级语言程序设计课程实验报告

实验题目：实验五 数组、指针与字符串		学号：202300130183
日期：2024 年 3 月 4 日	班级：2023 级智能班	姓名：宋浩宇
Email：202300130183@mail.sdu.edu.cn		
<p>实验目的：</p> <ul style="list-style-type: none">1. 练习对象数组。2. 训练使用指针的能力：深刻认识灵活性与潜在的危险，理解指针与引用的不同。3. 练习动态内存的使用。		
<p>实验软件和硬件环境：</p> <p>实验软件：Windows 11 家庭中文版(x64) Visual Studio 2022</p> <p>硬件环境：处理器：13th Gen Intel(R) Core(TM) i9-13980HX 2.20 GHz RAM 32.0 GB (31.6 GB 可用)</p>		
<p>实验步骤与内容：</p> <p>第 5 章 1.</p> <div><div>Microsoft Visual Studio</div><div>输入一个3×3矩阵： 1 2 3 4 5 6 7 8 9 原矩阵的转置为： 1 4 7 2 5 8 3 6 9 F:\Homework\高程作业\实验5\codes\x64\Debug\codes.exe (进程 12388)已退出，代码为 0。 按任意键关闭此窗口。 . . .</div></div>		

```
Microsoft Visual Studio × + ▾ − □ ×  
输入一个3×3的矩阵：  
9 8 7  
6 5 4  
3 2 1  
原矩阵的转置为：  
9 6 3  
8 5 2  
7 4 1  
  
F:\Homework\高程作业\实验5\codes\x64\Debug\codes.exe (进程 11560)已退出，代码为 0。  
按任意键关闭此窗口 . . .|
```

```
Microsoft Visual Studio × + ▾ − □ ×  
输入字符串：  
abc  
def  
连接之后的字符串为：abcdef  
  
F:\Homework\高程作业\实验5\codes\x64\Debug\codes.exe (进程 9008)已退出，代码为 0。  
按任意键关闭此窗口 . . .|
```

```
Microsoft Visual Studio × + - □ ×
输入字符串：
uvw
xyz
连接之后的字符串为： uvwxyz

F:\Homework\高程作业\实验5\codes\x64\Debug\codes.exe (进程 30752)已退出，代码为 0。
按任意键关闭此窗口 . . .
```

```
Microsoft Visual Studio × + - □ ×
该员工的信息如下：
姓名： 小明
街道地址： 山东大学青岛校区
城市： 青岛
邮政编码： 123456
成功更改姓名
该员工的信息如下：
姓名： 小红
街道地址： 山东大学青岛校区
城市： 青岛
邮政编码： 123456

F:\Homework\高程作业\实验5\codes\x64\Debug\codes.exe (进程 14264)已退出，代码为 0。
按任意键关闭此窗口 . . .
```

```
Microsoft Visual Studio  ×  +  -  □  ×

该员工的信息如下：
姓名：小明
街道地址：山东大学青岛校区
城市：青岛
邮政编码：123456
该员工的信息如下：
姓名：小红
街道地址：山东大学青岛校区
城市：青岛
邮政编码：123456
该员工的信息如下：
姓名：小刚
街道地址：山东大学青岛校区
城市：青岛
邮政编码：123456
该员工的信息如下：
姓名：小华
街道地址：山东大学青岛校区
城市：青岛
邮政编码：123456
该员工的信息如下：
姓名：小李
街道地址：山东大学青岛校区
城市：青岛
邮政编码：123456

F:\Homework\高程作业\实验5\codes\x64\Debug\codes.exe (进程 18612)已退出，代码为 0。
按任意键关闭此窗口 . . .
```

第 6 章 2.这里出的问题是，fn1 函数返回的是一个 int 类型的指针，而 a 是一个 int 类型的变量。修改的话就是将 `int a=fn1();`这一行改为 `int *a=fn1();`;

第 6 章 3.第一个是错的，因为在第三行试图将一个 `const int*`赋给 `int*`，这对编译器来说是两种不同的类型。

第 6 章 4.fig07_15to17 使用的是一维数组，因此在输出学生成绩的时候输出的是成绩的值，并且统计结果也只有这 10 个数据；而 fig07_22to24 使用的是二维数组，因此在输出学生成绩时使用同样的函数输出的是是个地址，并且统计结果是 30 个数据。

第 6 章 5.1)这个语句的含义是，先创建一个指向一个 Object 对象的指针 a，在创建一个 Object 对象的引用 b，b 引用的是由 a 所指向的 Object 对象复制的新的 Object 对象。

第 6 章 5.2)若改成 a)则含义为 b 是一个指向由 a 所指向的 Object 对象复制而来的一个新 Object 对象的指针；若改成 b)则含义为 b 是一个由 a 所指向的 Object 对象复制而来的一个新 Object 对象。

第 6 章 5.3)此题通过在 `exec()` 函数中添加 `delete a`;来避免内存泄漏; `cc` 不需要 `delete`。

第 6 章 6.(1)`Point` 不是数组名。

第 6 章 6.(2)它不是数组。

第 6 章 6.(3)因为要在主函数里使用 `points.element(0).move(5, 0)`;这种连续两次点运算的语句来对数据进行修改, 因此需要返回引用; 可以返回指针, 但是需要将主函数里的 `points.element(0).move(5, 0)`;和 `points.element(1).move(15,20)`;修改为 `points.element(0)->move(5,0)`;和 `points.element(1)->move(15,20)`;不能返回值, 因为返回的值是对原本数据的一个复制, 无法对原数据进行修改。

第 6 章 7.首先使用 `vector` 可以自动分配和释放内存; 其次 `vector` 还封装了许多其他实用功能。

第 6 章 8.因为浅拷贝只是将被复制对象里所有的值复制到新的对象中, 而原本的 `ArrayOfPoints` 类中有指针型成员, 这使得两个对象的这个成员指向的是同一块内存, 因此对被复制对象的指针型变量所指的内存空间进行操作, 复制出来的对象也会受到影响; 例 6-16 不会有这种问题, 因为其中的 `Point` 类中没有指针或引用型的成员。

第 6 章 9.首先构造函数和析构函数并不是按照自己所写的那样执行的, 这个构造和析构本身还是系统自动进行的, 而自己所写的构造函数和析构函数只是在系统自动执行这个过程的是时候插入的代码段。因为程序员通过 `new` 申请的内存不会由系统自动释放, 因此想要释放一个对象所占用的内存时, 如果这个对象在构造中设计 `new` 操作, 则需要自行在析构函数中添加对应的 `delete` 语句; 然后浅拷贝只是对各个成员变量值的复制, 因此如果使用的类中有指针或者引用型的成员变量, 如果使用浅拷贝就需要考虑两个对象中的指针指向同一块内存或引用引用的是同一个对象这种情况, 而在写复制构造函数的时候也需要考虑这一点来确定是否要单独再对指针或引用成员变量的复制进行编写; 总之, 由于指针的灵活性和特殊性, 在编写程序的时候必须考虑指针这个因素。

第 6 章 10.(1)因为 `string` 对象在内部已经实现了深拷贝, 即 `string` 类重载了赋值运算符 (`operator=`) 和拷贝构造函数, 这样在对象复制时, 它们会复制字符串的内容, 而不是简单地复制指针。

第 6 章 10.(2)`vector` 和 `string` 都是动态分配内存的, 而普通数组在定义好的时候内存占用就已经确定了; `vector` 和 `string` 在内部定义了很多实用的操作, 比普通数组使用方便很多; 因为它们在内部都实现了深拷贝, 即它们都对赋值运算符进行了重载, 从而实现再复制时复制内容而不是单纯复制指针。

结论分析与体会:

了解并掌握了对象数组这种用法; 了解了指针的灵活性和在对象的浅拷贝中的潜在危险; 了解并掌握了动态内存的分配和删除等使用方法; 了解了构造函数和析构函数的本质; 了解了运算符重载的存在。

就实验过程中遇到的问题及解决处理方法（如有）：
未遇到问题