

# 第三次作业

---

## 概念题

- 一、C++ 中的成员对象是指什么？创建包含成员对象的类的对象时，构造函数和析构函数的调用次序是怎样的？
- 二、在哪些情况下，会调用类的拷贝构造函数？什么时候需要自定义拷贝构造函数，为什么？
- 三、请说明 C++ 中 `const` 和 `static` 关键字的作用。
- 四、简述 C++ 中友元的概念、友元的特性以及友元的利弊。

## 编程题

- 一、阅读并执行下面的代码，根据运行结果指出代码存在的问题，并进行相应修改。

要求：根据运行结果详细说明源代码存在的问题，给出相应的改进方法；对源代码进行修改，根据修改后程序的运行结果说明修改的正确性。需要提交源代码的运行结果、源代码存在的问题、改进后的代码（改进部分写备注）、改进后代码的运行结果并说明其正确性。

### 3\_1.cpp

```
#include <iostream>
#include <cstring>
using namespace std;

class Matrix
{
private:
    int dim;
    double *m_data;
public:
    Matrix(int d);
    ~Matrix();
};

Matrix::Matrix(int d)
{
    dim = d;
    m_data = new double[dim*dim];
    cout << "Matrix" << endl;
}
```

```

Matrix::~Matrix()
{
    cout << "~Matrix " << (int)m_data << endl;
    delete [] m_data;
    m_data = NULL;
}

int main()
{
    {
        Matrix m1(5);
        Matrix m2(m1);
    }
    system("pause");
    return 0;
}

```

二、按照要求补全以下代码，完成成员对象的初始化。

要求：

1. 补全代码中所缺部分，共4处。

具体要求：

- 补全1：补充 NBAPlayer 类的构造函数，对成员变量 name 初始化；要求补全后程序调用的是 Shooting 类的默认构造函数对 shoot 初始化；
- 补全2：补充 NBAPlayer 类的构造函数，对成员变量 name 初始化；要求补全后程序调用的是 Shooting(float ftp, float fgp, float tpp) 的构造函数对 shoot 初始化；
- 补全3：实例化 NBAPlayer 对象 p1，要求 p1.name 初始化为 Curry，p1.shoot 的各项命中率初始化为 0.2（即采用默认初始化）；
- 补全4：实例化 NBAPlayer 对象 p2，要求 p2.shoot 的 FTPercentage 初始化为 0.9，FGPercentage 初始化为 0.71，TPPercentage 初始化为 0.44。

2. 提交完整的补全后的代码。

3\_2.cpp

```

#include <iostream>
using namespace std;

class Shooting
{
    float FTPercentage; // 罚球命中率
    float FGPercentage; // 投篮命中率
    float TPPercentage; // 三分命中率
public:
    Shooting()
    {
        FTPercentage = 0.2;
    }
}

```

```

        FGPercentage = 0.2;
        TPPercentage = 0.2;
    }
    Shooting(float ftp, float fgp, float tpp)
    {
        FTPercentage = ftp;
        FGPercentage = fgp;
        TPPercentage = tpp;
    }
};

class NBAPlayer
{
    Shooting shoot; // 实例化 Shooting 对象 shoot
    string name;
public:
    //补全1. 调用 Shooting 的默认构造函数对 shoot 初始化;
    //补全2. 调用 Shooting(float ftp, float fgp, float
tpp) 构造函数对 shoot 初始化;
};

int main()
{
    //补全3. p1.name 初始化为 Curry, p1.shoot的各项命中率采用默
认初始化;
    //补全4. p2.name 初始化为 Curry, p2.shoot的 FTPercentage
初始化为 0.9,FGPercentage 初始化为 0.71, TPPercentage 初始化为
0.44
    return 0;
}

```

三、定义一个 Component（零件）类，拥有整型数据成员 Weight、静态整型数据成员 TotalWeights 以及整型成员函数 GetWeights()。

要求：每定义一个对象，TotalWeights 增加该零件的重量 Weight；成员函数 GetWeights() 获取 Weight；TotalWeights 初始化为0；实例化两个对象之后，输出后者的 TotalWeights。

给出 main() 函数部分：

```

int main()
{
    int a, b;
    cin>>a>>b;
    Component A(a);
    Component B(b);
    cout<<B.TotalWeights<<endl;
    return 0;
}

```

Sample Input

12 24

Sample Output

36

四、定义狮子 **Lion** 与老虎 **Tiger** 两个类，二者都有 **weight** 私有整型属性，定义二者的一个友元函数 **totalWeights()**，计算二者的重量和。设计程序，定义狮子与老虎两个对象之后，调用 **totalWeights()**，计算二者的重量和，然后输出。

给出 **main** 函数部分：

```
int main()
{
    int w1, w2;
    cin>>w1>>w2;
    Lion L(w1);
    Tiger T(w2);
    cout<<totalWeights(L, T)<<endl;
}
```

Sample Input

12 24

Sample Output

36