第七次作业

概念题

请阐述C++中动态绑定和静态绑定的概念，并说明在什么情况下会发生动态绑定。

编程题

一、阅读以下程序，思考分析后写出其运行结果。

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

class Member
{
public:
    Member() { cout << "member construct" << endl; }
    Member(const Member& M) { cout << "member copy construct" << endl; }
    ~Member() { cout<< "member destruct" << endl; }

};

class Base
{
public:
    Base() { cout << "base construct" << endl; }
    Base(const Base& B) { cout << "base copy construct" << endl; }
    ~Base() { cout << "base destruct" << endl; }
};

class Derived:Base
{
    Member m1, m2;
public:
    Derived(Member mm1, Member mm2):m1(mm1), m2(mm2) { cout << "derived construct" <<
endl; }
    Derived(const Derived& D) { cout << "derived copy construct" << endl; }
    ~Derived() { cout << "derived destruct" << endl; }
};

int main()
{
    Member m1;
    Member m2;
    Derived d1(m1, m2);
    Derived d2(d1);
    return 0;
}
```

二、定义一个引擎类Engine，包含以下操作：

- 打开引擎油仓 open(返回是否打开成功)
- 注油 addoil(返回是否注油成功)
- 关闭引擎油仓 close(返回是否关闭成功)
- 启动引擎 start(返回是否启动成功)
- 熄火 stop(返回是否熄火成功)
- 显示油量 display(返回当前油量)
- 检查油的质量 checkoil(返回油的质量是否为优)

定义一个Motor类，需要先调用Engine类的打开引擎油仓、注油、关闭引擎油仓完成加油操作，然后启动引擎，最后熄火；定义一个AdvancedMotor类，除了需要完成加油、启动、熄火操作外，还需要调用显示油量和检查油的质量方法，返回当前油量和油的质量。请考虑应该使用继承还是聚集比较合理，并给出完整代码。

三、完善下面的代码，使得程序能够正常结束

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

class A
{
    private:
        int x;
    public:
        A(int x)
        {
            this->x = x;
        }
        int get_x()
        {
            return x;
        }
};

class B: public A
{
    private:
        char* str;

    public:
        B(int x, int length, char* s):A(x)
        {
            str = new char[length+1];
            for(int i = 0;i<length;i++)
                str[i]=s[i];
            str[length]='\0';
        }
        ~B()
        {
            if(str!=NULL)
            {
                delete[] str;
                str = NULL;
```

```
            }
        }

        void print() {
            cout << this->str << "," << this->get_x() << endl;
        }
    };

void f()
{
    B b1(10, 5, (char*)"Hello"), b2(20, 5, (char*)"World");
    b1.print();
    b2.print();
    b1 = b2;
    b1.print();
    b2.print();
}

int main()
{
    f();
    cout<<"Hello World"<<endl;
    return 0;
}
```