

1512432 - Nguyễn Đăng Quang

Lớp 15CTT2

Bài Tập Lý Thuyết Tuần 8

Bài 8.1

```
int main() {  
    //Cho biết thứ tự gọi hàm dựng với :  
    //theo thứ tự từ lỗi -> vỏ, tức là lớp cơ sở được tạo  
    trước, lớp kế thừa được tạo sau  
  
    { C obj(1, 2, 3); } //theo thứ tự, hàm dựng lớp A(1)  
    được gọi trước -> hàm dựng B(1,2) -> C(1,2,3)  
  
    { C obj(4); } //theo thứ tự, hàm dựng A(0) được tạo  
    trước nhất-> hàm dựng mặc định lớp B() -> hàm dựng C(4)  
  
    { C obj; } //theo thứ tự từ lỗi đến vỏ, hàm dựng A(0)  
    -> hàm dựng mặc định B() -> hàm dựng mặc định C()  
  
    return 0;  
}
```

Bài 8.2

```
/*Xây dựng hàm dựng cho lớp GiaoVien, khởi tạo với:  
- Họ tên cho trước, mức lương 500000, số ngày nghỉ 0.  
- Họ tên, mức lương cho trước, số ngày nghỉ 0.  
- Họ tên, mức lương, số ngày nghỉ cho trước*/
```

```
GiaoVien::GiaoVien(const string ho_ten) {  
    int n = ho_ten.length() + 1;  
    m_strHoTen = new char[n];  
    for (int i = 0; i < n; i++) {  
        m_strHoTen[i] = ho_ten[i];  
    }  
}
```

```

    }
    m_fMucLuong = 500000;
    m_iSoNgayNghỉ = 0;
}
GiaoVien::GiaoVien(const string ho_ten, float mucLuong) {
    int n = ho_ten.length() + 1;
    m_strHoTen = new char[n];
    for (int i = 0; i < n; i++) {
        m_strHoTen[i] = ho_ten[i];
    }
    m_fMucLuong = mucLuong;
    m_iSoNgayNghỉ = 0;
}
GiaoVien::GiaoVien(const string ho_ten, float mucLuong,
int so_ngay_nghỉ) {
    int n = ho_ten.length() + 1;
    m_strHoTen = new char[n];
    for (int i = 0; i < n; i++) {
        m_strHoTen[i] = ho_ten[i];
    }
    m_fMucLuong = mucLuong;
    m_iSoNgayNghỉ = so_ngay_nghỉ;
}
//giải quyết vấn đề con trỏ trong phương thức huỷ
GiaoVien::~GiaoVien() {
    if (m_strHoTen != NULL)
        delete[]m_strHoTen;
}

// giải quyết vấn đề con trỏ trong hàm dựng sao chép
GiaoVien::GiaoVien(const GiaoVien& x) {
    int n = strlen(x.m_strHoTen) + 1;
    m_strHoTen = new char[n];
    for (int i = 0; i < n; i++)
        m_strHoTen[i] = x.m_strHoTen[i];
    m_fMucLuong = x.m_fMucLuong;
    m_iSoNgayNghỉ = x.m_iSoNgayNghỉ;
}

//vấn đề con trỏ trong toán tử gán =
GiaoVien& GiaoVien::operator=(const GiaoVien& x) {
    if (this != &x) {

```

```

        if (m_strHoTen != NULL)
            delete[] m_strHoTen;
        int n = strlen(x.m_strHoTen) + 1;
        m_strHoTen = new char[n];
        for (int i = 0; i < n; i++)
            m_strHoTen[i] = x.m_strHoTen[i];
        m_fMucLuong = x.m_fMucLuong;
        m_iSoNgayNghỉ = x.m_iSoNgayNghỉ;
    }
    return *this;
}

```

//xây dựng phương thức lấy địa chỉ đối tượng trong class GiaoVien

```

    GiaoVien* GetThis() {
        return this;
    }

```

/*Xây dựng hàm dựng cho lớp GVCN, khởi tạo với:

- Họ tên, lớp chủ nhiệm cho trước, mức lương 500000, số ngày nghỉ 0.
- Họ tên, mức lương, lớp chủ nhiệm cho trước, số ngày nghỉ 0.
- Họ tên, mức lương, số ngày nghỉ, lớp chủ nhiệm cho trước.*/*

```

GVCN::GVCN(const string ho_ten, const string
lop_chu_nhiem) :GiaoVien(ho_ten) {
    int n = lop_chu_nhiem.length() + 1;
    m_strLopCN = new char[n];
    for (int i = 0; i < n; i++) {
        m_strLopCN[i] = lop_chu_nhiem[i];
    }
}

```

```

GVCN::GVCN(const string ho_ten, float mucLuong, const
string lop_chu_nhiem) :GiaoVien(ho_ten, mucLuong) {
    int n = lop_chu_nhiem.length() + 1;
    m_strLopCN = new char[n];
}

```

```

        for (int i = 0; i < n; i++) {
            m_strLopCN[i] = lop_chu_nhiem[i];
        }
    }

GVCN::GVCN(const string ho_ten, float muLuong, int
so_ngay_nghi, const string lop_chu_nhiem)
:GiaoVien(ho_ten, muLuong, so_ngay_nghi) {
    int n = lop_chu_nhiem.length() + 1;
    m_strLopCN = new char[n];
    for (int i = 0; i < n; i++) {
        m_strLopCN[i] = lop_chu_nhiem[i];
    }
}

//vấn đề con trỏ trong phương thức huỷ
GVCN::~~GVCN() {
    if (m_strLopCN != NULL)
        delete[]m_strLopCN;
}

//vấn đề con trỏ trong hàm dựng sao chép
GVCN(const GVCN& x) :GiaoVien(x) {
    int n = strlen(x.m_strLopCN) + 1;
    m_strLopCN = new char[n];
    for (int i = 0; i < n; i++)
        m_strLopCN[i] = x.m_strLopCN[i];
}

//vấn đề con trỏ trong toán tử gán bằng =
GVCN& operator=(GVCN& x) {
    if (this != &x) {
        GiaoVien *u1, *u2;
        u1 = this->GetThis();
        u2 = x.GetThis();
        *u1 = *u2;

        if (m_strLopCN != NULL)
            delete[]m_strLopCN;
        int n = strlen(x.m_strLopCN) + 1;
        m_strLopCN = new char[n];
        for (int i = 0; i < n; i++)
            m_strLopCN[i] = x.m_strLopCN[i];
    }
    return *this;
}

```

}

Bài 8.3

//Cho biết thứ tự gọi hàm dựng :

```
{
    Z obj(5); //theo thứ tự từ lõi đến vỏ, hàm dựng
    mặc định X() được tạo ->
    // -> hàm dựng của lớp Y(5) (vì gọi hàm dựng xong
    i mới ++)-> Z(5)
}

{
    Y obj1(6); //hàm dựng mặc định lớp X() -> Y(6)
    Y obj2(obj1); //gọi hàm dựng sao chép mặc định
    của Y
}

{
    Z obj1(7); //hàm dựng Y(7) -> Z(7)
    Z obj2(obj1); //gọi hàm dựng sao chép mặc định
    của Z
}
```

Bài 8.5



