

Nội dung tuần 01

Luyện tập cách xây dựng Hàm dựng và Hàm hủy cho lớp đối tượng.

Hướng dẫn

Xem xét thứ tự khởi tạo cũng như hủy đối tượng (thời điểm hàm dựng hay hàm hủy được gọi) thông qua ví dụ sau:

```
static int numMax = 0;
class Object
{
private:
    int info;
    int curNum;

public:
    Object();
    Object(int value);
    ~Object();
};

Object::Object()
{
    info = 0;
    curNum = ++numMax;
    cout << "Khoi tao (mac dinh) doi tuong " << curNum << endl;
}

Object::Object(int value)
{
    info = value;
    curNum = ++numMax;
    cout << "Khoi tao (co doi so) doi tuong " << curNum << endl;
}

Object::~~Object()
{
    cout << "Huy doi tuong " << curNum << endl;
}

void Create()
{
    Object obj1(5);    //biến cục bộ hàm Create
    Object obj2;        //biến cục bộ hàm Create
}

Object objGlobal;    //biến toàn cục
void main()
{
    Object obj1;        //biến cục bộ hàm main
    Create();            //gọi hàm Create()
    Object obj2(3);    //biến cục bộ hàm main
}
```

```
}
```

Có nhận xét gì về kết quả xuất của chương trình.

```
Khởi tạo <mac dinh> doi tuong 1
Khởi tạo <mac dinh> doi tuong 2
Khởi tạo <co doi so> doi tuong 3
Khởi tạo <mac dinh> doi tuong 4
Hủy doi tuong 4
Hủy doi tuong 3
Khởi tạo <co doi so> doi tuong 5
Hủy doi tuong 5
Hủy doi tuong 2
Hủy doi tuong 1
Press any key to continue . . .
```

Bài tập

Bài 1

Khai báo và cài đặt lớp Ngày sao cho hàm main sau chạy đúng

```
void main()
{
    Ngay n1;                //1/1/1
    n1.Xuat();
    Ngay n2(02,10,2014);    //2/10/2014
    n2.Xuat();
    Ngay n3(-10,16,2000);   //1/12/2000
    n3.Xuat();
    Ngay n4(1000);          //?/9/3
    n4.Xuat();
}
```

Bài 2

Khai báo và cài đặt lớp Thời Gian để chạy đúng với hàm main sau:

```
void main()
{
    ThoiGian tg1;                //00:00:00
    tg1.Xuat();
    ThoiGian tg2(1212);          //00:20:12
    tg2.Xuat();
    ThoiGian tg3(125,45);        //02:05:45
    tg3.Xuat();
    ThoiGian tg4(12,239,-78);    //15:59:00
    tg4.Xuat();
}
```

Bài 3

Khai báo và cài đặt lớp Sinh Viên để chạy đúng với hàm main sau

```
void main()
{
    //MSSV: "xxxxxxx", HoTen: "xxxx", NgaySinh: xx/xx/xxxx, DLT: 10, DTH: 10.
    class SinhVien sv1;
    sv1.Xuat();
    SinhVien sv2("1363001");
    sv2.Xuat();
    SinhVien sv3("1363002", "Nguyen Van A", 7, 8.5);
    sv3.Xuat();
    SinhVien sv4("1363003", "Tran Thi B", 8, 9.5, 12, 3, 1996);
    sv4.Xuat();
    NgayThang n(30,2,1996);
    SinhVien sv5("1363004", "Ngo Van C", 5, 6, n);
    sv5.Xuat();
    SinhVien sv6(sv5);
    sv6.Xuat();
}
```

Handwritten annotations:

- class* is written in red next to `class SinhVien`.
- string* is written in red above the second parameter of `sv3`.
- d + n* is written in red above the last two parameters of `sv4`.
- A red arrow points from `n` in `sv5` to the `n` in `sv4`.
- A red bracket is under `sv5` in the `sv6(sv5)` line.

Bài 4

Khai báo và cài đặt lớp Mảng Số Nguyên để chạy đúng với hàm main sau

```
void main()
{
    MangSoNguyen msn1;
    msn1.Xuat();
    MangSoNguyen msn2(5);
    msn2.Xuat();
    int arr[10];
    for (int i = 0; i < 8; i++)
    {
        arr[i] = i * i;
    }
    MangSoNguyen msn3(arr, 8);
    msn3.Xuat();
    MangSoNguyen msn4(msn3);
    msn4.Xuat();
}
```