

## Nội dung tuần 03

Luyện tập cách xây dựng các hàm toán tử cho lớp đối tượng.

### Bài tập

❖ Yêu cầu

A: 2 bài bất kỳ

H: Làm hết 6 bài

Lưu ý: mục tiêu quan trọng là các thành phần dữ liệu hợp lệ của các lớp đối tượng, gợi ý kết quả chỉ là một trong số các cách khởi tạo giá trị với tham số.

Khi khởi tạo với nhiều thành phần dữ liệu có mối liên quan với nhau thì nên thực hiện mối liên quan đó (ví dụ số ngày dư thì đổi sang tháng, tháng dư thì đổi sang năm,...)

### Bài 1

Cài đặt lại ví dụ 1.

### Bài 2

Cài đặt lại ví dụ 2.

### Bài 3

Cài đặt lại ví dụ 3.

### Bài 4

Khai báo và cài đặt lớp đối tượng **Ngày** để chạy đúng với hàm main sau:

```
int main()
{
    Ngày n1;                // 1/1/1
    Ngày n2(2, 10, 2014);    // 2/10/2014
    Ngày n3(-10, 16, 2000);  // 10/04/2001
    Ngày n4(1000);           // 27/9/3
    Ngày n5 = n2 + n3;       // 12/2/4016
    Ngày n6 = n1 + 5000;     // 10/10/15
    Ngày n7 = 1234 + n4;     // 14/2/7
    Ngày n8 = 190 + n6 + n7; // 2/7/23
    Ngày n9 = n8 - n6;       // 1/9/7
    Ngày n10 = 12000 - n9;   // 9/2/26
    if (n10 > n6)
    {
        n10 = n2 - 1000 + n6;
    }
    cout << n1 << endl << n2 << endl << n3 << endl << n4 << endl;
    cout << n5 << endl << n6 << endl << n7 << endl << n8 << endl;
    cout << n9 << endl << n10 << endl;
    system("pause");
}
```

## Hướng dẫn thực hành PP LT hướng đối tượng

```
}    return 0;
```

### Bài 5

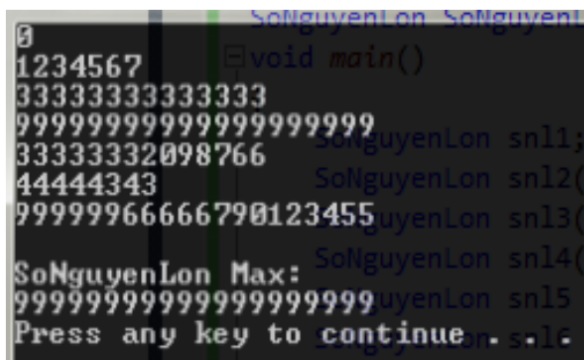
Khai báo và cài đặt lớp đối tượng **SoNguyenLon** để chạy đúng với hàm main sau

```
int main()
{
    SoNguyenLon snl1;
    SoNguyenLon snl2(1234567);
    SoNguyenLon snl3(3, 14);
    SoNguyenLon snl4(9, 20);
    SoNguyenLon snl5 = snl3 - snl2;
    SoNguyenLon snl6 = 45678910 - snl2;
    SoNguyenLon snl7 = snl4 - snl3 + 123456789;

    cout << snl1 << endl << snl2 << endl << snl3 << endl;
    cout << snl4 << endl << snl5 << endl << snl6 << endl;
    cout << snl7 << endl << endl;

    cout << "SoNguyenLon Max:" << endl;
    cout << SoNguyenLon::SNLMax() << endl;
    system("pause");
    return 0;
}
```

Với kết quả như sau:



The screenshot shows the output of the C++ program. It displays the values of several SoNguyenLon objects: snl1 (0), snl2 (1234567), snl3 (3333333333333333), snl4 (9999999999999999), snl5 (33333332098766), snl6 (44444343), and snl7 (9999996666790123455). It also shows the result of the subtraction snl3 - snl2 (SoNguyenLon snl5) as 33333332098766. The program then prints "SoNguyenLon Max:" followed by the value of SNLMax() which is 9999999999999999. Finally, it prompts the user to press any key to continue.

Biết rằng số chữ số tối đa của **SoNguyenLon** là 100 và không có giá trị âm (số bé hơn trừ số lớn hơn sẽ có kết quả là 0).

### Bài 6

Khai báo và cài đặt lớp đối tượng **ThoiGian** để chạy đúng với hàm main sau

```
int main()
{
    ThoiGian tg1;           // 00:00:00
    ThoiGian tg2(1212);     // 00:20:12
    ThoiGian tg3(125, 45);  // 02:05:45
    ThoiGian tg4(12, 239, -78); // 16:00:18
    ThoiGian tg5 = tg2 + tg3; // 02:25:57
}
```

```
ThoiGian tg6 = 5000 + tg2;           // 01:43:32
ThoiGian tg7 = tg4 - tg6;           // 14:16:46
ThoiGian tg8 = 12300 - tg4;         // 00:00:00
ThoiGian tg9, tg10;
if (tg8 <= tg3)
{
    tg9 = tg1 + tg2 + 36000; // 10:20:12
}
if (12345 <= tg5)
{
    tg10 = tg5 + 12345;           // 05:51:42
}
cout << tg1 << endl << tg2 << endl << tg3 << endl << tg4 << endl;
cout << tg5 << endl << tg6 << endl << tg7 << endl << tg8 << endl;
cout << tg9 << endl << tg10 << endl;
}
```

### Hướng dẫn

#### Ví dụ 1 (bản chất toán tử chính là hàm)

```
class SoPhuc
{
private:
    int _thuc, _ao;
public:
    SoPhuc();
    SoPhuc(const int&, const int&);
    SoPhuc operator+(const int&);

    void operator<<(ostream&);
};

SoPhuc::SoPhuc()
{
    _thuc = _ao = 0;
}

SoPhuc::SoPhuc(const int &t, const int &a)
{
    _thuc = t;
    _ao = a;
}

SoPhuc SoPhuc::operator+(const int& n)
{
    SoPhuc rt = *this;
    rt._thuc += n;
    return rt;
}

void SoPhuc::operator<<(ostream& os)
{
    os << _thuc;
    if (_ao != 0)
    {
        os << ((_ao > 0) ? " + " : " - ") << _ao << "i";
    }
}
```

```
    }  
}  
  
int main()  
{  
    SoPhuc sp1(3, 5);  
    SoPhuc sp2 = sp1.operator+(10);  
    SoPhuc sp3 = sp1 + 10;  
    sp1.operator<<(cout);  
    cout << endl;  
    sp2.operator<<(cout);  
    cout << endl;  
    sp3.operator<<(cout);  
    cout << endl;  
    system("pause");  
    return 0;  
}
```

### Ví dụ 2 (tự động ép kiểu không tường minh)

```
class SoPhuc  
{  
private:  
    int _thuc, _ao;  
  
public:  
    SoPhuc();  
    SoPhuc(const int&);  
    SoPhuc(const int&, const int&);  
    SoPhuc operator+(const SoPhuc&);  
  
    void operator<<(ostream&);  
};  
  
SoPhuc::SoPhuc()  
{  
    _thuc = _ao = 0;  
}  
  
SoPhuc::SoPhuc(const int &t) : SoPhuc()  
{  
    cout << "using... ";  
    _thuc = t;  
}  
  
SoPhuc::SoPhuc(const int& t, const int& a)  
{  
    _thuc = t;  
    _ao = a;  
}  
  
SoPhuc SoPhuc::operator+(const SoPhuc &sp)  
{  
    SoPhuc rt = *this;  
    rt._thuc += sp._thuc;  
    rt._ao += sp._ao;  
    return rt;  
}
```

```
void SoPhuc::operator<<(ostream& os)
{
    os << _thuc;
    if (_ao != 0)
    {
        os << ((_ao > 0) ? " + " : " - ") << _ao << "*i";
    }
}

int main()
{
    SoPhuc sp1(3, 5);
    SoPhuc sp3 = sp1 + 10;
    sp1.operator<<(cout);
    cout << endl;
    sp3.operator<<(cout);
    cout << endl;
    system("pause");
    return 0;
}
```

- ✓ Chương trình chạy bình thường không lỗi dù không cài đặt `operator+` với `int`
- ✓ Bản chất trình biên dịch sẽ tự ép kiểu giá trị `int` 10 sang `SoPhuc` (cần có `constructor` với đối số là `int`) sau đó sử dụng `operator+` với `SoPhuc`.

### Ví dụ 3 (cài đặt operator có sử dụng friend)

- ✓ Chương trình sau sẽ có lỗi biên dịch, vì sao?

```
class SoPhuc
{
private:
    int _thuc, _ao;

public:
    SoPhuc();
    SoPhuc(const int&);
    SoPhuc(const int&, const int&);
    SoPhuc operator+(const int&);
    SoPhuc operator+(const SoPhuc&);

    void operator<<(ostream&);
};
/// phần cài đặt như các ví dụ trước...

int main()
{
    SoPhuc sp1(3, 5);
    SoPhuc sp2 = 10 + sp1;
    sp1.operator<<(cout);
    cout << endl;
    sp2.operator<<(cout);
    cout << endl;
    system("pause");
    return 0;
}
```

```
}
```

- ✓ Phần highlight để thực hiện được tức là phải có **operator+** của số int 10 với SoPhuc (tức `10.operator+(sp1)` nhưng lưu ý int không phải là object nên hàm **operator** cần là: **operator+(10, sp1)** là function không phải method trong trường hợp này).
- ✓ Cụ thể cài đặt thêm hàm **operator+** như sau thì chương trình không lỗi

```
class SoPhuc
{
private:
    int _thuc, _ao;

public:
    SoPhuc();
    SoPhuc(const int&);
    SoPhuc(const int&, const int&);
    SoPhuc operator+(const int&);
    SoPhuc operator+(const SoPhuc&);

    void operator<<(ostream&);
};
/// phần cài đặt như các ví dụ trước...
SoPhuc operator+(const int&, const SoPhuc&);
SoPhuc operator+(const int &n, const SoPhuc &sp)
{
    cout << "using..." << endl;
    SoPhuc spR;
    return spR;
}

int main()
{
    SoPhuc sp1(3, 5);
    SoPhuc sp2 = 10 + sp1;
    sp1.operator<<(cout);
    cout << endl;
    sp2.operator<<(cout);
    cout << endl;
    system("pause");
    return 0;
}
```

- ✓ Tuy nhiên để cài đặt xử lý đúng đắn cho hàm **operator+** nói trên thì cần phải truy xuất đến thành phần của SoPhuc, do đó cần gài hàm **operator+** kể trên vào class SoPhuc với từ khóa **friend** để lấy quyền truy xuất đầy đủ như là method. Ví dụ hoàn chỉnh như sau:

```
class SoPhuc
{
private:
    int _thuc, _ao;

public:
    SoPhuc();
    SoPhuc(const int&);
```

```
SoPhuc(const int&, const int&);

friend SoPhuc operator+(const int&, const SoPhuc&);
friend ostream& operator<<(ostream&, const SoPhuc&);
};

SoPhuc operator+(const int& n, const SoPhuc& sp)
{
    SoPhuc spR(n);
    spR._thuc += sp._thuc;
    spR._ao += sp._ao;
    return spR;
}

SoPhuc::SoPhuc()
{
    _thuc = _ao = 0;
}

SoPhuc::SoPhuc(const int &t) : SoPhuc()
{
    _thuc = t;
}

SoPhuc::SoPhuc(const int& t, const int& a)
{
    _thuc = t;
    _ao = a;
}

ostream& operator<<(ostream& os, const SoPhuc& sp)
{
    os << sp._thuc;
    if (sp._ao != 0)
    {
        os << ((sp._ao > 0) ? " + " : " - ") << sp._ao << "*i";
    }
    return os;
}

int main()
{
    SoPhuc sp1(3, 5);
    SoPhuc sp2 = 10 + sp1;
    cout << sp1 << endl;
    cout << sp2 << endl;
    system("pause");
    return 0;
}
```