

# LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG



## CHƯƠNG 3 TOÁN TỬ

ThS: Phạm Nguyễn Sơn Tùng

Email: [pnstung@fit.hcmus.edu.vn](mailto:pnstung@fit.hcmus.edu.vn)

# NỘI DUNG BÀI HỌC

**1** Giới thiệu về toán tử (operator)

**2** Các toán tử minh họa.

**3** Con trỏ this, từ khóa friend.

**4** Demo bài tập ứng dụng tại lớp.

**5** Bài tập về nhà.

# ĐỊNH NGHĨA TOÁN TỬ

**Định nghĩa:** Các phép toán (toán tử) mà ngôn ngữ lập trình cung cấp chỉ thực hiện trên những kiểu dữ liệu cơ bản (**int**, **float**, **double**...)

```
void main()
{
    int a = 3;
    int b = 9;
    cout<<a + b;
    a++;
    cout<<a;
}
```

# ĐỊNH NGHĨA TOÁN TỬ

Nếu a và b là **đối tượng** thuộc một **lớp** nào đó thì không thực hiện được phép cộng như ở trên. Vậy giải quyết ra sao?

```
void main()  
{  
    CPhanSo a;  
    CPhanSo b(3, 4);  
    cout<<a + b;  
    a++;  
    cout<<a;  
}
```



**SAI**

# GIẢI QUYẾT VẤN ĐỀ

## Giải pháp 1: Xây dựng phương thức (method)

PhanSo.h

```
CPhanSo Cong (CPhanSo) ;
```

PhanSo.cpp

```
CPhanSo CPhanSo::Cong (CPhanSo b)
{
    CPhanSo c;
    c.TuSo = TuSo*b.MauSo + MauSo*b.TuSo;
    c.MauSo = MauSo*b.MauSo;
    return c;
}
```

# GIẢI QUYẾT VẤN ĐỀ

## Giải pháp 1: Xây dựng phương thức (method)

++ main.cpp

```
void main()  
{  
    CPhanSo a;  
    CPhanSo b(3,4);  
    CPhanSo c = a.Cong(b);  
    c.Xuat();  
}
```

# GIẢI QUYẾT VẤN ĐỀ

**Giải pháp 2:** Xây dựng toán tử (Operator). Hay còn gọi là nạp chồng toán tử (operator overloading)

- **Loại 1:** Toán tử thuộc lớp, là phương thức của lớp.
- **Loại 2:** Toán tử độc lập, không thuộc về lớp nào (**friend**).



# GIẢI QUYẾT VẤN ĐỀ

Ví dụ: Xây dựng toán tử (Operator)

 PhanSo.h

```
CPhanSo operator+ (CPhanSo) ;
```

PhanSo.cpp

```
CPhanSo CPhanSo::operator+ (CPhanSo b)
{
    TuSo = TuSo*b.MauSo + MauSo*b.TuSo;
    MauSo = MauSo*b.MauSo;
    return *this;
}
```



# GIẢI QUYẾT VẤN ĐỀ

## Giải pháp 2: Xây dựng toán tử

++ main.cpp

```
void main()
{
    CPhanSo a;
    CPhanSo b(3,4);
    CPhanSo c = a + b;
    c.Xuat();
}
```

# CON TRỎ THIS

**Định nghĩa:** là con trỏ đặc biệt trong C++. Có 2 cách làm việc với con trỏ this.



# CON TRỎ THIS

**Từ khóa this:** dùng để truy cập thuộc tính phương thức của đối tượng.

```
void CPhanSo::Nhap ()
{
    cout<<"Nhap tu so: ";
    cin>>this->TuSo;
    cout<<"Nhap mau so: ";
    cin>>this->MauSo;
}
```

# CON TRỎ THIS

**Con trỏ this:** giữ địa chỉ của đối tượng đang gọi phương thức.

```
CPhanSo& CPhanSo::operator+=(const CPhanSo &b)
{
    *this = *this + b;
    return *this;
}
```

# CÁC LOẠI TOÁN TỬ THÔNG DỤNG

- 1. Toán tử một ngôi:** sử dụng một đối tượng để thay đổi đối tượng đó: `operator-`, `operator+`, `operator!`, `operator ++` (tăng sau), `--` (giảm sau), `++` (tăng trước), `--` (giảm trước)
- 2. Toán tử hai ngôi:** thao tác đến 2 đối tượng để tìm kết quả mong muốn:
  - `operator-(class)`, `+`, `*`, `/`
  - `operator+=(class)`, `-=`, `*=`, `/=`

# CÁC LOẠI TOÁN TỬ THÔNG DỤNG

**3. Toán tử quan hệ:** sử dụng để so sánh các đối tượng với nhau: `operator>`, `<`, `==`, `>=`, `<=`, `!=`

**4. Toán tử gán:** dùng để gán đối tượng này cho đối tượng kia, tương tự như một copy constructor .

- `operator=`

**5. Toán tử gọi hàm:** dùng để tạo ra một phương thức với nhiều tham số đầu vào khác nhau.

- `operator()`

# TOÁN TỬ NHẬP XUẤT

## ➤ Toán tử nhập

```
friend istream& operator>>(istream& , CPhanSo &);
```

## ➤ Toán tử xuất

```
friend ostream& operator<<(ostream& , CPhanSo);
```

# TOÁN TỬ NHẬP XUẤT

## ➤ Toán tử nhập

```
istream& operator>>(istream& is, CPhanSo &p)
{
    cout<<"Nhập tử: ";
    is>>p.TuSo;
    cout<<"Nhập mẫu: ";
    is>>p.MauSo;
    return is;
}
```



# TỪ KHÓA FRIEND

## ➤ Từ khóa Friend:

- Có thể truy xuất thành phần của lớp.
- Được khai báo bên trong lớp.
- Định nghĩa bên ngoài lớp.



# TOÁN TỬ NÂNG CAO

+	-	*	/	%	^	&
	~	!	=	<	>	+=
-=	*=	/=	%=	^=	&=	=
<<	>>	>>=	<<=	==	!=	<=
>=	&&		++	--	->*	,
->	[]	()	new	new[]	delete	delete[]

# BÀI TOÁN MINH HỌA

**Bài 01:** Xây dựng lớp **CThoiGian** thực hiện các toán tử sau:

- Toán tử so sánh:  $\geq$ ,  $\neq$
- Toán tử cộng trừ 2 thời gian:  $+$ ,  $-$
- Toán tử tăng giảm (tăng trước):  $++$ ,  $--$  (tăng và giảm một giây)

# BÀI TOÁN MINH HỌA

**Bài 02:** Xây dựng lớp **CNgay** thực hiện các toán tử sau:

- Toán tử so sánh:  $<$ ,  $==$
- Toán tử tăng giảm (tăng sau):  $++$ ,  $--$  (tăng và giảm một ngày)
- Toán tử  $>>$  và  $<<$

# BÀI TOÁN MINH HỌA

➤ **Bài 03:** Xây dựng lớp **CDonThuc** thực hiện các toán tử sau:

- Toán tử so sánh:  $>$ ,  $<$ ,  $==$ ,  $>=$ ,  $<=$ ,  $!=$
- Toán tử toán học:  $+$ ,  $-$ ,  $*$ ,  $/$
- Toán tử một ngôi:  $!$  (đạo hàm),  $\sim$  (nguyên hàm)
- Toán tử nhập xuất:  $<<$ ,  $>>$

# TÀI LIỆU THAM KHẢO

- 1. Lập trình hướng đối tượng, Trần Đan Thư, Đinh Bá Tiến, Nguyễn Tấn Trần Minh Khang, NXB Khoa Học Kỹ Thuật, 2010.
- 2. Lập trình hướng đối tượng, Trần Văn Lãng, NXB Thống Kê, 2004.
- 3. Object-oriented Programming in c++, 4th Edition, Robert Lafore, SAMS, 1997.
- 4. C++ Primer, Fifth Edition, Stephen Prata, SAMS, 2004.
- 5. Slide bài giảng của: Thầy Nguyễn Minh Huy, Thầy Hồ Tuấn Thanh, Thầy Đinh Bá Tiến, Thầy Trần Văn Lãng, Thầy Đặng Bình Phương, Cô Đặng Thị Thanh Nguyên.
- 6. Các website về lập trình:
  - <http://www.cplusplus.com/>
  - <http://stackoverflow.com/>
  - <http://www.codeproject.com/>