

# Hàm toán tử

GV. Nguyễn Minh Huy



- Khái niệm cơ bản.
- Các toán tử đặc biệt.
- Hàm Friend.



- **Khái niệm cơ bản.**
- Các toán tử đặc biệt.
- Hàm Friend.



## ■ Sự khác biệt giữa int và PhanSo?

*// Thao tác trên kiểu int*

```
void main()
{
    int  a, b;
    int  c = a + b;
}
```

*// Thao tác trên kiểu PhanSo*

```
void main()
{
    PhanSo  p1, p2;
    PhanSo  p3 = p1.cong(p2);
}
```

## ■ Chương trình tìm min có gì sai?

**template <class T>**

```
T timMin(T a, T b)
{
    return (a < b) ? a : b;
}
```

void main()

```
{
    int  a, b;
    int c = timMin(a, b);

    PhanSo p1, p2;
    PhanSo p3 = timMin(p1, p2);
}
```



## ■ Hàm toán tử:

### ■ Khái niệm:

- Một hàm trong chương trình.
- Có tên là ký hiệu toán tử.
- Khai báo: từ khóa **operator** <toán tử>.

*PhanSo **operator** +(const PhanSo &p1, const PhanSo &p2);*

### ■ Công dụng:

- Sử dụng toán tử trên kiểu người dùng định nghĩa.
- Định nghĩa nhiều cách thực hiện toán tử (nạp chồng hàm).

*float **opeartor** +(const PhanSo &p, float so);*

*float x = p1 + 3.14;*



## ■ Phân loại hàm toán tử:

### ■ Toán tử độc lập:

*PhanSo **operator** +(const PhanSo &p1, const PhanSo &p2);*

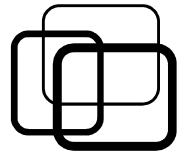
- Không thuộc lớp nào.
- Số lượng tham số = ngôi toán tử.

### ■ Toán tử thuộc lớp:

*PhanSo **PhanSo::operator** +(const PhanSo &p);*

- Là phương thức của lớp.
- Số lượng tham số = ngôi toán tử - 1.

### ■ Cách sử dụng 2 loại là như nhau!!



## ■ Những toán tử có thể định nghĩa lại:

Ngôi	Nhóm	Toán tử
1 Ngôi (Unary)	Tăng giảm	++, --
	Dấu số học	+, -
	Logic	!, ~
	Con trỏ	*, &
	Ép kiểu	int, float, double, ...
2 Ngôi (Binary)		
	Số học	+, -, *, /, %, +=, -=, *=, /=, %=
	So sánh	>, <, ==, >=, <=, !=
	Logic	&&,   , &,
	Nhập xuất	<<, >>
	Gán	=
	Lấy chỉ số mảng	[ ]



## ■ Những toán tử không thể định nghĩa lại:

Toán tử	Ý nghĩa
.	Truy xuất phần tử
.*	Truy xuất con trỏ phần tử
::	Toán tử ::
? :	Toán tử điều kiện
#	Chỉ thị tiền xử lý
# #	Chỉ thị tiền xử lý





- Những hạn chế (4 không):
  - Không thể tạo toán tử mới.
  - Không thể định nghĩa lại toán tử trên kiểu cơ bản.
  - Không thể thay đổi ngôi của toán tử.
  - Không thể thay đổi độ ưu tiên của toán tử.

# Khái niệm cơ bản



## ■ Dr. Guru khuyên:

### ■ Quy tắc định nghĩa toán tử:

- Tên hàm: **operator** <tên toán tử>.
- Số lượng tham số: **ngôi toán tử**.
- Kiểu tham số: **kiểu toán hạng**.
- Kiểu trả về: **kết quả toán tử**.

## ■ Ví dụ:

- Toán tử `>` (lớp `PhanSo`).
- Toán tử `[ ]` (lớp `Mang`).





- Khái niệm cơ bản.
- **Các toán tử đặc biệt.**
- Hàm Friend.

# Các toán tử đặc biệt



- Toán tử gán (`=`, `+=`, `-=`, `*=`, `/=`, ...):
  - Trang bị toán tử `+=` cho lớp **PhanSo**.
  - Toán tử `+=` mấy ngôi?
  - Toán tử `+=` trả về gì?

```
PhanSo& PhanSo::operator +=( const PhanSo &p );
```



## ■ Toán tăng giảm (++ , --):

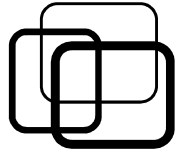
- Trang bị toán tử ++ cho lớp **PhanSo**.
- Toán tử ++ mấy ngôi?
- Toán tử ++ trả về gì?
- Phân biệt tiền tố, hậu tố?

PhanSo& PhanSo::operator ++( );

// Tiền tố.

PhanSo& PhanSo::operator ++( int x );

// Hậu tố, dùng tham số giả.



- Khái niệm cơ bản.
- Các toán tử đặc biệt.
- **Hàm Friend.**



## ■ Xét toán tử +:

- Trang bị toán tử + cho lớp **PhanSo**.

- Dùng toán tử độc lập.

```
PhanSo operator + ( const PhanSo &p1, const PhanSo &p2 );
```

- Làm sao truy xuất thành phần **private** của lớp?

## ■ Xét toán tử xuất <<:

- Trang bị toán tử << cho lớp **PhanSo**.

```
PhanSo p( 1, 3 );  
cout << p;
```

- Toán tử << thuộc về lớp nào?



## ■ Hàm Friend:

- Hàm có thể truy xuất thành phần **private** của lớp.
- Cách sử dụng:
  - Khai báo trong lớp: từ khóa **friend** <chữ ký hàm>.
  - Cài đặt ngoài lớp: không có **friend**.

```
class PhanSo
{
    friend ostream& operator <<(ostream &os, const PhanSo &p);
};
ostream& operator <<(ostream &os, const PhanSo &p)
{
    os << p.m_tu << "/" << p.m_mau << endl;
    return os;
}
```





## ■ Hàm toán tử:

- Hàm có tên là ký hiệu toán tử.
- Dùng để trang bị toán tử cho kiểu tự định nghĩa.
- Có 2 loại hàm toán tử:
  - Toán tử độc lập.
  - Toán tử thuộc lớp.

## ■ Các toán tử đặc biệt:

- `=`, `+=`, `-=`, `++`, `--`.

## ■ Hàm Friend:

- Hàm truy xuất được thành phần của lớp.

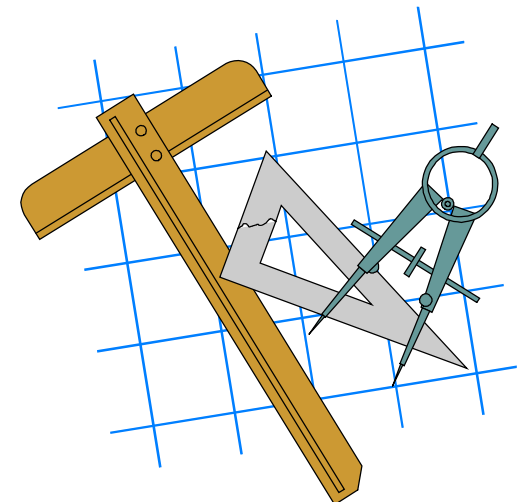




## ■ Bài tập 4.1:

Trang bị cho lớp **phân số** những toán tử sau:

- Toán tử số học:  $+$ ,  $*$ .
- Toán tử so sánh:  $>$ ,  $<$ ,  $==$ ,  $>=$ ,  $<=$ ,  $!=$ .
- Toán tử gán:  $=$ ,  $+=$ ,  $*=$ .
- Toán tử một ngôi:  $++$ ,  $--$  (tăng, giảm 1 đơn vị).
- Toán tử ép kiểu:  $(float)$ ,  $(int)$ .
- Toán tử nhập, xuất:  $>>$ ,  $<<$ .

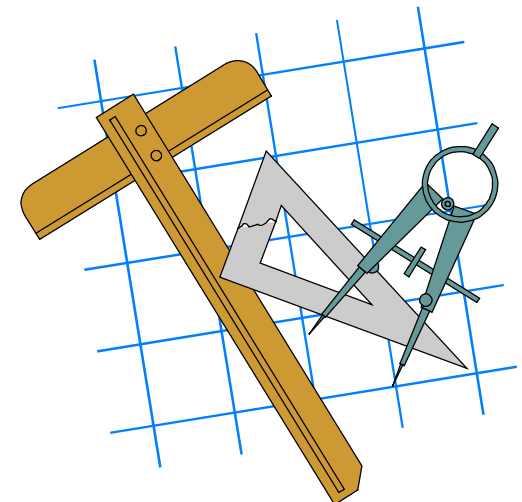




## ■ Bài tập 4.2:

Trang bị cho lớp **đơn thức** những toán tử sau:

- Toán tử số học:  $+$  (cùng bậc),  $*$ .
- Toán tử so sánh:  $>$ ,  $<$ ,  $==$ ,  $>=$ ,  $<=$ ,  $!=$ .
- Toán tử gán:  $=$ ,  $+=$  (cùng bậc),  $*=$ .
- Toán tử một ngôi:
  - $++$ ,  $--$  (tăng, giảm bậc).
  - $!$  (đạo hàm).
- Toán tử nhập, xuất:  $>>$ ,  $<<$ .

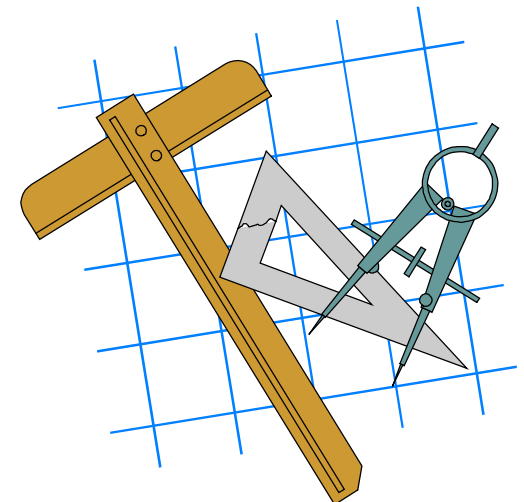




## ■ Bài tập 4.3:

Trang bị cho lớp **mảng** (phần kiểu bất kỳ) những toán tử sau:

- Toán tử gán:  $=$ .
- Toán tử truy xuất phần tử:  $[ ]$ .
- Toán tử ép kiểu:  $(T *)$  (ép về con trỏ kiểu T).
- Toán tử nhập, xuất:  $>>$ ,  $<<$ .





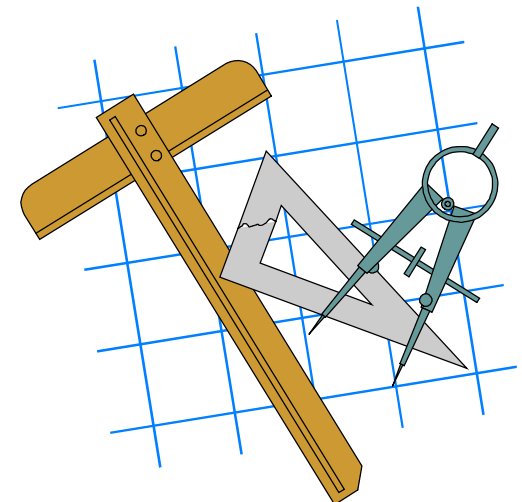
## ■ Bài tập 4.4:

Xây dựng lớp **Time** có những phương thức sau:  
(Nhóm tạo hủy)

- Khởi tạo mặc định vào thời điểm hiện tại.
- Khởi tạo từ giờ, phút, giây cho trước.
- Khởi tạo từ giây tuyệt đối trong ngày.
- Khởi tạo từ một đối tượng Time khác.

(Nhóm truy xuất thông tin)

- Lấy/gán giờ, phút, giây.
- Lấy/gán giây tuyệt đối trong ngày.





## ■ Bài tập 4.4:

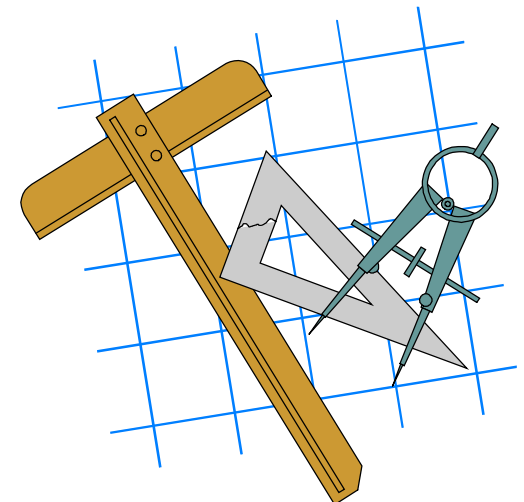
Xây dựng lớp **Time** (tiếp theo):

*(Nhóm xử lý nghiệp vụ)*

- So sánh thứ tự trước sau với đối tượng Time khác.
- Tính khoảng cách đến đối tượng Time khác (đơn vị giây).
- Cộng giờ, cộng phút, cộng giây.

*(Nhóm toán tử)*

- Toán tử so sánh:  $>$ ,  $<$ ,  $==$ ,  $>=$ ,  $<=$ ,  $!=$ .
- Toán tử cộng, trừ giây:  $+$ ,  $-$ .
- Toán tử tăng, giảm 1 giây:  $++$ ,  $--$ .
- Toán tử nhập xuất:  $>>$ ,  $<<$ .





## ■ Bài tập 4.5:

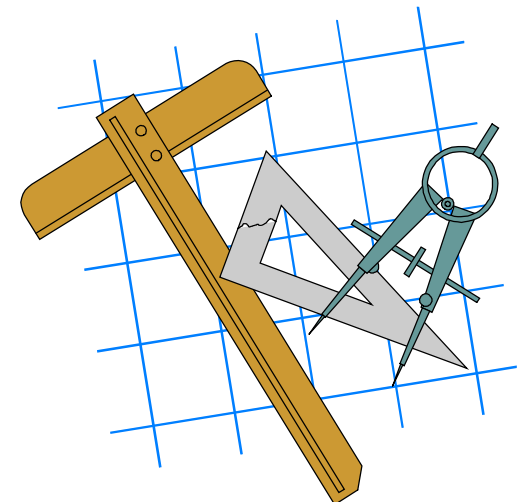
Xây dựng lớp **Date** có những phương thức sau:

*(Nhóm tạo hủy)*

- Khởi tạo mặc định vào ngày hiện tại.
- Khởi tạo từ ngày, tháng, năm cho trước.
- Khởi tạo từ năm và ngày tuyệt đối trong năm.
- Khởi tạo từ một đối tượng Date khác.

*(Nhóm truy xuất thông tin)*

- Lấy/gán ngày, tháng, năm.
- Lấy/gán ngày tuyệt đối trong năm.
- Thông báo thứ trong tuần.
- Thông báo tuần trong năm.





## ■ Bài tập 4.5:

Xây dựng lớp **Date** (tiếp theo):

*(Nhóm xử lý nghiệp vụ)*

- Kiểm tra năm nhuận.
- So sánh thứ tự trước sau với đối tượng Date khác.
- Tính khoảng cách đến đối tượng Date khác (đơn vị ngày).
- Cộng ngày, cộng tháng, cộng năm.

*(Nhóm toán tử)*

- Toán tử so sánh:  $>$ ,  $<$ ,  $==$ ,  $>=$ ,  $<=$ ,  $!=$ .
- Toán tử cộng, trừ ngày:  $+$ ,  $-$ .
- Toán tử tăng, giảm 1 ngày:  $++$ ,  $--$ .
- Toán tử nhập xuất:  $>>$ ,  $<<$ .

