

# **Chương 4**

## **SƠ LƯỢC THƯ VIỆN IOSTREAM.H**

# 1. VÍ DỤ DẪN NHẬP 1

- Bài toán: Viết lệnh nhập giá trị cho một số nguyên a và xuất số nguyên ra màn hình bằng cách sử dụng thư viện `iostream.h`
- Phong cách cũ
  1. `int a;`
  2. `printf("Nhap mot so nguyen:");`
  3. `scanf("%d", &a);`
  4. `printf("So nguyen vua nhap:%d", a);`
- Đoạn chương trình với thư viện `iostream.h`
  1. `int a;`
  2. `cout<<"Nhap mot so nguyen: ";`
  3. `cin>>a;`
  4. `cout<<"So nguyen vua nhap:"<<a;`

## 2. VÍ DỤ DẪN NHẬP 2

- Bài toán: Viết hàm nhập thông tin của một phân số bằng cách sử dụng thư viện `iostream.h`
- Cấu trúc dữ liệu
  1. `struct phanso`
  2. `{`
  3. `int tu;`
  4. `int mau;`
  5. `};`
  6. `typedef struct phanso`  
`PHANSO;`
- Định nghĩa hàm

## 2. VÍ DỤ DẪN NHẬP 2 (tiếp)

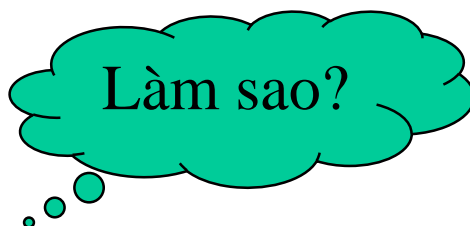
### – Định nghĩa hàm

```
11. void Nhap (PHANSO &x)
12. {
13.     cout<<"Nhap tu:";
14.     cin>>x.tu;
15.     cout<<"Nhap mau:";
16.     cin>>x.mau;
17. }

18. void Xuat (PHANSO x)
19. {
20.     cout<<x.tu<<"/"<<x.mau;
21. }
```

### 3. ĐẶT VẤN ĐỀ

- Nhập xuất một đối tượng phân số
  1. CPhanSo a;
  2. a.Nhap();
  3. a.Xuat();
- Nhập, xuất một đối tượng phân số với thư viện iostream.h
  1. CPhanSo a;
  2. cin>>a;
  3. cout<<a;
- Lưu ý
  - + Ký hiệu >> được gọi là **toán tử vào**.
  - + Ký hiệu << được gọi là **toán tử ra**.



## 4. GIẢI QUYẾT VẤN ĐỀ

- Để giải quyết vấn đề trên ta phải định nghĩa
  - ✓ Toán tử vào (operator >>)
  - ✓ Toán tử ra (operator <<)

cho lớp đối tượng CPhanSo.

- Ngoài ra, trong khi giải quyết vấn đề này ta còn sử dụng kỹ thuật hàm bạn (friend function) của phương pháp lập trình hướng đối tượng.
- Một “hàm bạn” của lớp đối tượng được phép truy xuất đến tất cả các thành phần của đối tượng thuộc về lớp đó bất chấp thành phần được khai báo trong phạm vi nào.

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

operator >>

21. };



## 4. GIẢI QUYẾT VẤN ĐỀ (tiếp)

- Khai báo lớp

## 11. class CPhanSo

12. {

13. | private:

```
14. int tu;
```

```
15. int mau;
```

16. public:

operator >>

```
(istream &is
```

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

operator >>

(istream &is,CPhanSo &x);

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

istream& operator >>

(istream &is,CPhanSo &x);

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

friend istream& operator >>

(istream &is,CPhanSo &x);

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

friend istream& operator >>

(istream &is,CPhanSo &x);

operator <<

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

friend ostream& operator >>

(ostream &is, CPhanSo &x);

operator <<

(ostream &os &x);

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

          friend istream& operator >>

                  (istream &is,CPhanSo &x);

          operator <<

                  (ostream &os,CPhanSo &x);

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

          friend ostream& operator >>

                  (ostream &is,CPhanSo &x);

          ostream& operator <<

                  (ostream &os,CPhanSo &x);

21. };



## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

friend ostream& operator >>

(ostream &is,CPhanSo &x);

friend ostream& operator <<

(ostream &os,CPhanSo &x);

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (tiếp)

– Định nghĩa toán tử vào

```
1. istream& operator>>(istream &is,  
                           CPhanSo &x)  
2. {  
3.     cout << "Nhap tu";  
4.     is >> x.tu;  
5.     cout << "Nhap mau";  
6.     is >> x.mau;  
7.     return is;  
8. }
```

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

friend istream& operator >>

(istream &is,CPhanSo &x);

operator <<

21. };

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Định nghĩa toán tử ra

```
1. ostream& operator << (ostream &os,  
                           CPhanSo &x)  
2. {  
3.     os<< x.tu<<"/"<<x.mau;  
4.     return os;  
5. }
```

Tại sao phải trả về một đối tượng thuộc lớp ostream?

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Khai báo lớp

11. class CPhanSo

12. {

13.     private:

14.         int tu;

15.         int mau;

16.     public:

friend ostream& operator >>

(ostream &is,CPhanSo &x);

friend ostream& operator <<

(ostream &os,CPhanSo &x);

21. };

## 5. HƯỚNG DẪN SỬ DỤNG 1

- Hãy xem xét đoạn chương trình sau:
  1. CPhanSo a;
  2. cin >> a;
  3. cout <<a ;
- Trong câu lệnh thứ hai của đoạn chương trình trên ta nói: hàm **operator >>** được gọi thực hiện với 2 đối số là cin và đối tượng a.

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Định nghĩa toán tử vào

```
1. istream& operator>>(istream &is,  
                           CPhanSo &x)  
2. {  
3.     cout << "Nhap tu";  
4.     is >> x.tu;  
5.     cout << "Nhap mau";  
6.     is >> x.mau;  
7.     return is;  
8. }
```

## 5. HƯỚNG DẪN SỬ DỤNG 1

- Hãy xem xét đoạn chương trình sau:  
1. CPhanSo a;  
2. cin >> a;  
3. cout <<a ;
- Trong câu lệnh thứ hai của đoạn chương trình trên ta nói: hàm **operator >>** được gọi thực hiện với 2 đối số là cin và đối tượng a.
- Trong câu lệnh thứ ba của đoạn chương trình trên ta nói: hàm **operator <<** được gọi thực hiện với 2 đối số là cout và đối tượng a.



## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Định nghĩa toán tử ra

```
1. ostream& operator << (ostream &os,  
                           CPhanSo &x)  
2. {  
3.     |   os<< x.tu<<"/"<<x.mau;  
4.     |   return os;  
5. }
```

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

```
1. CPhanSo a, b, c;
```

```
2. cin >>a >>b >>c ;
```

```
3. cout<<a <<b <<c ;
```

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2. cin >>a >>b >>c ;

3. cout<<a <<b <<c ;

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Định nghĩa toán tử vào

```
1. istream& operator>>(istream &is,  
                           CPhanSo &x)  
2. {  
3.     cout << "Nhap tu";  
4.     is >> x.tu;  
5.     cout << "Nhap mau";  
6.     is >> x.mau;  
7.     return is;  
8. }
```

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

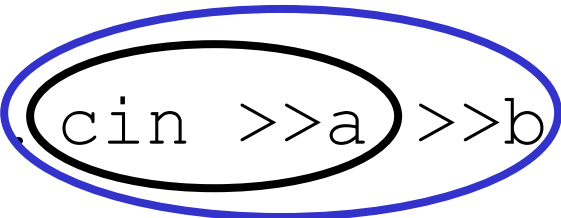
2. cin >>a >>b >>c ;

3. cout<<a <<b <<c ;

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2.  cin >> a >> b >> c ;

3. cout << a << b << c ;

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

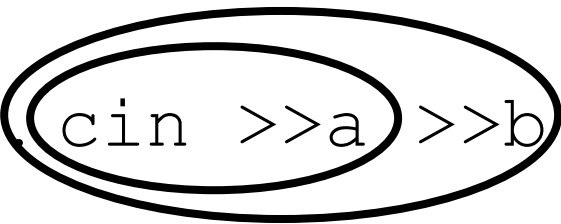
– Định nghĩa toán tử vào

```
1. istream& operator>>(istream &is,  
                           CPhanSo &x)  
2. {  
3.     cout << "Nhap tu";  
4.     is >> x.tu;  
5.     cout << "Nhap mau";  
6.     is >> x.mau;  
7.     return is;  
8. }
```

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2.  cin >> a >> b >> c ;

3. cout << a << b << c ;



## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2. cin >> a >> b >> c ;

3. cout << a << b << c ;

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Định nghĩa toán tử vào

```
1. istream& operator>>(istream &is,  
                           CPhanSo &x)  
2. {  
3.     cout << "Nhap tu";  
4.     is >> x.tu;  
5.     cout << "Nhap mau";  
6.     is >> x.mau;  
7.     return is;  
8. }
```

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2. cin >> a >> b >> c ;

3. cout << a << b << c ;

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2. cin >> a >> b >> c ;

3. cout << a << b << c ;

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Định nghĩa toán tử ra

```
1. ostream& operator << (ostream &os,  
                           CPhanSo &x)  
2. {  
3.     |   os<< x.tu<<"/"<<x.mau;  
4.     |   return os;  
5. }
```

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2. cin >> a >> b >> c ;

3. cout << a << b << c ;

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2. cin >> a >> b >> c ;

3. cout << a << b << c ;

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Định nghĩa toán tử ra

```
1. ostream& operator << (ostream &os,  
                           CPhanSo &x)  
2. {  
3.     |   os<< x.tu<<"/"<<x.mau;  
4.     |   return os;  
5. }
```



## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2. cin >> a >> b >> c ;

3. cout << a << b << c ;

## 6. HƯỚNG DẪN SỬ DỤNG 2

- Hãy xem xét đoạn chương trình sau:

1. CPhanSo a, b, c;

2. cin >> a >> b >> c ;

3. cout << a << b << c ;

## 4. GIẢI QUYẾT VẤN ĐỀ (*tiếp*)

– Định nghĩa toán tử ra

```
1. ostream& operator << (ostream &os,  
                           CPhanSo &x)  
  
2. {  
3.     |   os<< x.tu<<"/"<<x.mau;  
4.     |   return os;  
5. }
```

## 7. ỨNG DỤNG

- Yêu cầu: Hãy định nghĩa toán tử vào và toán tử ra cho lớp đối tượng CNgay.

## 7. ỨNG DỤNG (tiếp)

### – Khai báo lớp

```
11. class CNgay
12. {
13.     private:
14.         int ng;
15.         int th;
16.         int nm;
17.     public:
18.         friend istream& operator >>
19.             (istream &is, CNgay &x);
20.         friend ostream& operator <<
21.             (ostream &os, CNgay &x);
22. } ;
```

## 7. ỨNG DỤNG (tiếp)

- Định nghĩa toán tử vào

```
11. istream& operator >>  
    (istream &is, CNgay &x)  
  
12. {  
13.     cout << "Nhap ngay:";  
14.     is >> x.ng;  
15.     cout << "Nhap thang:";  
16.     is >> x.th;  
17.     cout << "Nhap nam:";  
18.     is >> x.nm;  
19.     return is;  
20. }
```

## 7. ỨNG DỤNG (tiếp)

– Định nghĩa toán tử ra

```
1. ostream& operator <<  
    (ostream &os, CNgay&x)  
  
2. {  
3.     os<<x.ng<<" / "<<  
4.     x.th<<" / "<<  
5.     x.nm;  
6.     return os;  
7. }
```

## 8. BÀI TẬP VỀ NHÀ

- Hãy khai báo và định nghĩa hàm toán tử vào và hàm toán tử ra cho các lớp đối tượng sau:
  1. Lớp phân số (CPhanSo)
  2. Lớp điểm (CDiem)
  3. Lớp ngày (CNgay)
  4. Lớp thời gian (CThoiGian)
  5. Lớp đơn thức (CDonThuc)
  6. Lớp điểm không gian (CDiemKhongGian)
  7. Lớp đường thẳng (CDuongThang)
  8. Lớp hỗn số (CHonSo)
  9. Lớp số phức (CSoPhuc)
  10. Lớp đường tròn (CDuongTron)
  11. Lớp tam giác (CTamGiac)
  12. Lớp hình cầu (CHinhCau)