

# TIN HỌC ĐẠI CƯƠNG

## BÀI 5: CÁC CÂU LỆNH LỰA CHỌN

---

Phạm Xuân Cường  
Khoa Công nghệ thông tin  
[cuongpx@tlu.edu.vn](mailto:cuongpx@tlu.edu.vn)

- Câu lệnh if
- Câu lệnh if—else
- Viết hàm đệ quy
- Câu lệnh switch—case

## Câu lệnh if

---

## Câu lệnh if: Ví dụ mở đầu

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Nhap vao 1 hoac 2: ";
    cin >> n;
    if (n == 1)
        cout << "Chuc may man!\n"; // neu n = 1
    if (n == 2)
        cout << "Chuc vui ve!\n"; // neu n = 2

    return 0;
}
```

- Cú pháp:  
    if (điều—kiện)  
        công—việc
- Giải thích:
  - điều—kiện là một biểu thức logic
  - công—việc là câu lệnh đơn hoặc khối lệnh
  - Nếu điều—kiện đúng, thực hiện công—việc
  - Nếu điều—kiện sai, bỏ qua công—việc

## công–việc là câu lệnh/khối lệnh

- công–việc là một câu lệnh đơn:

```
if (x != 1)  
    y = 1 / (x - 1);
```

- công–việc là một khối lệnh (đặt giữa cặp dấu {}):

```
if (x != 1)  
{  
    y = 1 / (x - 1);  
    cout << y;  
}
```

## Sự khác nhau khi có/không có cặp dấu { }

- Khi có cặp dấu { }

```
double x = 1.2, y = 3.4;
```

```
if (x < 0) {
```

```
    x = 5.6;
```

```
    y = 7.8;
```

```
}
```

```
cout << x << " " << y; // sẽ in ra 1.2 và 3.4
```

- Khi không có cặp dấu { }

```
double x = 1.2, y = 3.4;
```

```
if (x < 0)
```

```
    x = 5.6;
```

```
    y = 7.8;
```

```
cout << x << " " << y; // sẽ in ra 1.2 và 7.8
```

## Kiểm tra tính chẵn lẻ: Dùng hai câu lệnh if

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Nhap vao mot so nguyen: ";
    cin >> n;
    if (n % 2 == 0)
        cout << n << " la mot so chan\n";
    if (n % 2 != 0)
        cout << n << " la mot so le\n";

    return 0;
}
```



## Kiểm tra tính chẵn lẻ: Dùng câu lệnh if-elses

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Nhap vao mot so nguyen: ";
    cin >> n;
    if (n % 2 == 0)
        cout << n << " la mot so chan\n";
    else // nguoc lai
        cout << n << " la mot so le\n";

    return 0;
}
```

## Câu lệnh if–else

---

- Cú pháp:

```
if (điều–kiện)  
    công–việc–1  
else  
    công–việc–2
```

- Giải thích:

- điều–kiện là một biểu thức logic
- công–việc–1 và 2 là câu lệnh đơn hoặc khối lệnh
- Nếu điều–kiện đúng, thực hiện công–việc–1
- Nếu điều–kiện sai, thực hiện công–việc–2

# Giải phương trình bậc nhất $ax + b = 0$

```
#include <iostream>
using namespace std;
int main()
{
    double a, b, x;
    cout << "Chương trình giải phương trình bậc nhất  $ax + b = 0$ \n";
    cout << "Nhập hệ số a: "; cin >> a;
    cout << "Nhập hệ số b: "; cin >> b;
    if (a != 0) {
        x = -b/a;
        cout << "Phương trình có nghiệm duy nhất x = " << x << endl;
    } else if (b == 0)
        cout << "Phương trình nghiệm đúng với mọi x" << endl;
    else
        cout << "Phương trình vô nghiệm" << endl;
    return 0;
}
```

- **Bài 1** (về nhà): Giải phương trình bậc hai
- **Bài 2** (tại lớp): Nhập vào ba số thực và hiển thị số lớn nhất ra màn hình
- **Bài 3** (tại lớp):
  - Nhập một đường tròn có tâm  $O(x_0, y_0)$  và bán kính  $R$
  - Nhập một điểm  $M(x_m, y_m)$
  - Kiểm tra xem điểm  $M$  nằm trong, nằm trên hay nằm ngoài đường tròn  $(O, R)$ ?

## Viết hàm đệ quy

---

- Hàm đệ quy là hàm được định nghĩa bằng chính bản thân nó
- Ví dụ: dãy Fibonacci được định nghĩa theo kiểu đệ quy

$$F_0 = 1$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \quad (n > 1)$$

- C++ cho phép viết hàm đệ quy tương tự như vậy

# Viết hàm đệ quy tính số Fibonacci

```
#include <iostream>
using namespace std;
int fibo(int n) {
    if (n <= 1)
        return 1;
    else
        return fibo(n-1) + fibo(n-2) ;
}
int main()
{
    int n;
    cout << "Nhap so tu nhien n: "; cin >> n;
    int fn = fibo(n); // tính số thu n trong dãy Fibonacci
    cout << "Số thu " << n << " trong dãy Fibonacci là ";
    cout << fn << endl;
    return 0;
}
```



## Viết hàm đệ quy tính $n!$

```
#include <iostream>
using namespace std;
int giai_thua(int n) {
    if (n <= 1)
        return 1;
    else
        return n * giai_thua(n-1) ;
}
int main()
{
    int n;
    cout << "Nhap so tu nhien n: "; cin >> n;
    int gt = giai_thua(n); // tinh n!
    cout << n << "! = " << gt << endl;
    return 0;
}
```

## Câu lệnh switch–case

---

## Câu lệnh switch—case: Ví dụ mở đầu

```
#include <iostream>
using namespace std;
int main()
    int n;
    cout << "Nhap vao 1, hoac 2: ";
    cin >> n;
    switch (n) {
        case 1:
            cout << "Good morning!"; // nếu n = 1
            break;
        case 2:
            cout << "Good afternoon!"; // nếu n = 2
            break;
        default:
            cout << "Ban da nhap sai du lieu!"; // nếu n khác 1,2
            break;
    }
    return 0;
}
```

# Cú pháp câu lệnh switch-case

**switch** (biểu-thức)

{

case hằng-1:

    công-việc-1

break;

case hằng-2:

    công-việc-2

break;

...

default:

    công-việc-ngầm-định

break;

}

Giải thích:

- Tùy theo giá trị của biểu thức bằng hằng nào thì công việc tương ứng sẽ được thực hiện, trong khi các công việc khác sẽ bị bỏ qua
- Nếu giá trị của biểu thức không bằng bất kỳ hằng nào, công việc ngầm định trong phần **default** sẽ được thực hiện

## Chú ý thêm về câu lệnh switch—case

- biểu—thức phải có giá trị thuộc các kiểu sau:
  - số nguyên
  - ký tự
  - logic(biểu—thức kiểu số thực không được phép)
- Phần **default** không bắt buộc phải có
- Các câu lệnh **break** cũng không bắt buộc phải có  
→ nhưng phải thận trọng khi bỏ **break**!

## Điều gì xảy ra khi bỏ break?

```
#include <iostream>
using namespace std;
int main()
{
    double x = 2.1;
    int k = 100;
    cout << "Truoc switch: x = " << x << endl; // x = 2.1
    switch (k) {
        case 100:
            x *= 2;
        case 200:
            x *= 3;
            break;
    }
    cout << "Sau switch: x = " << x << endl; // x = 12.6
    return 0;
}
```

# Tính tổng hoặc hiệu theo yêu cầu

```
#include <iostream>
using namespace std;
int main() {
    double x, y, tong, tich;
    int n;
    cout << "Nhap vao hai so thuc: "; cin >> x >> y;
    cout << "[1] Tinh tong, [2] Tinh tich: "; cin >> n;
    switch (n) {
        case 1:
            tong = x + y;
            cout << "Tong = " << tong << endl;
            break;
        case 2:
            tich = x * y;
            cout << "Tich = " << tich << endl;
            break;
        default:
            cout << "Ban da nhap sai du lieu!" << endl;
            break;
    }
    return 0;
}
```

# Một số phương thức làm việc với tập

```
int tinh_so_ngay(int thang, int nam) {  
    switch (thang) {  
        case 1:  
        case 3:  
        case 5:  
        case 7:  
        case 8:  
        case 10:  
        case 12:  
            return 31;  
        case 4:  
        case 6:  
        case 9:  
        case 11:  
            return 30;  
        case 2:  
            if ((nam % 100 != 0 && nam % 4 == 0) || (nam % 100 == 0 && nam % 400 == 0))  
                return 29;  
            else  
                return 28;  
    }  
}
```



**Questions?**