

# TIN HỌC ĐẠI CƯƠNG

---

## Bài 5: Vòng lặp trong C++ (phần 2)

# Nội dung chính

---

1. Vòng lặp FOR
2. Vòng lặp WHILE
3. Vòng lặp DO-WHILE
4. Từ khóa `break` và `continue`
5. Bài tập

Phần 1

# Vòng lặp FOR

# Lặp FOR

Ví dụ: in các số từ 1 đến 20 ra màn hình

```
for (int x = 1; x <= 20; x = x + 1) {  
    cout << x << endl;  
}
```

Quá trình thực hiện:

1. Khai báo biến  $x$  và gán giá trị  $x = 1$

→ 2. Kiểm tra  $x \leq 20$  không? Nếu không thì dừng

3. In giá trị  $x$  ra màn hình

4. Tăng  $x$  lên 1

→ 5. Chuyển về bước 2

# Lặp FOR

Ví dụ: in các số từ 1 đến 20 ra màn hình

```
for (int x = 1; x <= 20; x = x + 1) {  
    cout << x << endl;  
}
```

Quá trình thực hiện:

- Khai báo biến `x` và gán giá trị `x = 1`
- Kiểm tra xem `x <= 20` không? Không (vì `x` đang bằng 1)
- In ra 1 và xuống dòng mới
- Tăng giá trị của `x` lên 1 (`x = 2`)
- Kiểm tra xem `x <= 20` không? Không (vì `x` đang bằng 2)
- In ra 2 và xuống dòng mới
- Tăng giá trị của `x` lên 1 (`x = 3`)
- Kiểm tra xem `x <= 20` không? Không (vì `x` đang bằng 3)
- ...

# Lặp FOR

Cú pháp:

```
for (<khởi tạo>; <điều kiện>; <thay đổi>) {  
    <công việc>  
}
```

Quá trình thực hiện:

1. Chạy khối <khởi tạo>

→ 2. Kiểm tra <điều kiện>, nếu sai thì dừng lặp

3. Thực hiện <công việc>

4. Thực hiện <thay đổi>

↳ 5. Chuyển về bước 2

# Lặp FOR: vài ví dụ

```
// tính tổng các số từ 1 đến 100
```

```
for (int i = 1, tong = 0; i <= 100; i++) {  
    tong = tong + i;  
}
```

```
// tính tổng các số lẻ trong khoảng từ 1 đến n
```

```
for (int i = 1, tong = 0; i <= n; i = i + 2) {  
    tong = tong + i;  
}
```

```
// đoán thử xem vòng lặp này thực hiện điều gì
```

```
for (a = n - 1; 0 != (n % a); a = a - 1) {}
```

# Lặp FOR: ứng dụng

- Cấu trúc của vòng lặp FOR rất thích hợp với việc triển khai ý tưởng lặp dùng biến đếm
  - Đây cũng là cách dùng thông dụng nhất của FOR
- Nếu biến khai báo trong phần <khởi tạo> thì chỉ có thể sử dụng trong vòng FOR và bị hủy khi kết thúc vòng lặp
- Một số phần trong lặp FOR có thể để trống nếu không cần thiết: **for(;;) {}**
- Theo thống kê thì vòng lặp FOR là vòng lặp sử dụng nhiều nhất trong lập trình C/C++



Phần 2


# Vòng lặp WHILE

# Lặp WHILE

Cú pháp:

```
while (<điều kiện>) {  
    <công việc>  
}
```

Quá trình thực hiện:

- 
1. Kiểm tra <điều kiện>, nếu sai thì dừng lặp
  2. Thực hiện <công việc>
  3. Chuyển về bước 1

Ví dụ:

```
a = n - 1;  
while (0 != (n % a)) { a = a - 1; }
```

# Lặp WHILE: ứng dụng

- Vòng lặp WHILE sử dụng khi:
  - Không biết chính xác cần phải lặp lại bao nhiêu lần
  - Biết điều kiện để dừng việc lặp
- Ví dụ: nhập mật khẩu cho đến khi nhập đúng
- Nếu viết điều kiện dừng không tốt, có thể bị lặp “vô tận” (chương trình chạy không dừng)
  - Vòng lặp vô tận “**while(1) {...}**” lại là cách dùng ưa thích của nhiều lập trình viên
- Khi dùng lặp WHILE, chú ý khởi tạo giá trị các biến cần thiết trước khi vào vòng lặp

Phần 3


# Vòng lặp DO-WHILE

# Lặp DO-WHILE

Cú pháp:

```
do {  
    <công việc>  
} while (<điều kiện>);
```

Quá trình thực hiện:

- 
1. Thực hiện <công việc>
  2. Kiểm tra <điều kiện>, nếu sai thì dừng lặp
  3. Chuyển về bước 1

Ví dụ:

```
a = n;  
do { a = a - 1; } while (0 == (n % a));
```

# Lặp DO-WHILE: ứng dụng

- Vòng lặp DO-WHILE tương đối giống vòng lặp WHILE, nhưng sẽ kiểm tra điều kiện sau
  - Lặp WHILE kiểm tra điều kiện trước
  - Lặp DO-WHILE kiểm tra điều kiện sau
  - Lặp DO-WHILE sẽ thực hiện ít nhất 1 lần
- Một số thống kê chỉ ra lặp DO-WHILE dễ gây lỗi hơn vòng lặp FOR và lặp WHILE
  - Tuy nhiên không nhất thiết phải tránh dùng vòng lặp này chỉ vì sợ gây lỗi

Phần 4

# Từ khóa `break` và `continue`

# Từ khóa “break”

Sử dụng **break** khi cần dừng vòng lặp

Dùng được cho cả for, while và do-while:

```
for (int i = 0; i < 100; i++) {  
    <lệnh A>; // thực hiện <lệnh A>  
    break; // dừng vòng lặp for ngay lập tức  
    <lệnh B>; // <lệnh B> sẽ không được thực hiện  
}
```

Thường sử dụng khi phải chấm dứt lặp vì một lý do bất thường nào đó

- Ví dụ phần mềm đang thực hiện gửi email cho 10000 người, trong lúc đang gửi ta phát hiện ra lỗi nên dừng lại để sửa nội dung email



# Từ khóa “continue”

Sử dụng `continue` khi cần dừng bước lặp hiện tại, tiếp tục luôn bước lặp mới

Dùng được cho cả `for`, `while` và `do-while`:

```
for (int i = 0; i < 100; i++) {  
    <lệnh A>; // thực hiện <lệnh A>  
    continue; // trở về đầu vòng lặp, chạy bước mới  
    <lệnh B>; // <lệnh B> sẽ không được thực hiện  
}
```

Ví dụ: gửi email cho 10000 người, kiểm tra thấy địa chỉ người thứ 9 bị sai, ta không cần gửi tiếp mà chuyển sang người thứ 10,...

Phần 5

# Bài tập

# Bài tập

1. Nhập  $n$ , in ra màn hình  $n$  số nguyên kể từ 100 trở đi.
2. Nhập  $n$ , in ra màn hình các số chẵn nhỏ hơn  $n$ .
3. Nhập số  $n$  nguyên dương, tính và in các số chính phương từ  $1^2$  đến  $n^2$  ra màn hình, mỗi số một dòng

4. Tính giá trị số  $X$  dưới đây

$$X = 1 \times 2 + 2 \times 3 + \dots + 99 \times 100$$

5. Nhập số  $n$  nguyên dương, tính giá trị số  $Y$  dưới đây

$$Y = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

6. Nhập số  $n$  chẵn và tính giá trị của số  $Z$  sau đây

$$Z = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots - \frac{1}{n}$$