

# TIN ĐẠY CƯỜNG

---

## BÀI 6: LỰA CHỌN

# Nội dung buổi trước

---

- Cú pháp vòng lặp FOR, WHILE và DO-WHILE
- Vòng lặp FOR phù hợp nhất với những loại lặp biết trước số lần (thường sử dụng biến đếm)
- Vòng lặp WHILE và DO-WHILE thường sử dụng khi biết trước điều kiện kết thúc lặp
  - WHILE kiểm tra điều kiện trước khi lặp
  - DO-WHILE kiểm tra điều kiện sau khi lặp
- Ngắt vòng lặp bằng lệnh **break**
- Kết thúc sớm bước lặp bằng lệnh **continue**

# Nội dung chính

---

1. Đặt vấn đề
2. Khái niệm câu lệnh lựa chọn
3. Các loại lệnh lựa chọn
4. Câu lệnh `if` và `if-else`
5. Kết hợp lựa chọn và vòng lặp
6. Câu lệnh `switch`
7. Bài tập

Phần 1

# Đặt vấn đề

# Đặt vấn đề

---

- Ba cấu trúc điều khiển cơ bản trong máy tính
  - Tuần tự ← Đã học trong bài trước nữa
  - Lặp ← Chương 3 (bài trước)
  - **Lựa chọn** ← **Chương 4 (bài này)**
- Lựa chọn là một thao tác ra quyết định cơ bản trong cuộc sống
  - Sử dụng chọn làm một trong nhiều việc
  - Sử dụng khi suy diễn logic
  - Trong ngôn ngữ tự nhiên, thường được phát biểu dưới dạng điều kiện: **nếu ... thì ...** (if ... then ...)

# Đặt vấn đề

---

- Nhiều thuật toán cũng yêu cầu có sự chọn lựa
  - Các hành động xử lý ứng với từng tình huống
  - Kiểm tra điều kiện thực hiện công việc
  - ...
- Ví dụ:
  - Kiểm tra xem số nguyên  $a$  có là số chẵn không?
  - Giải phương trình bậc 2 (biện luận giá trị  $\Delta$ )
  - Ngày mai là ngày bao nhiêu?
  - Giá trị nào là lớn nhất trong 3 số  $a, b, c$ ?
  - ...

Phần 2

# Khái niệm câu lệnh lựa chọn

# Khái niệm câu lệnh lựa chọn

---

- **Khái niệm:** câu lệnh trong máy tính cho phép quyết định làm một việc hay không dựa trên một điều kiện cụ thể
- **Mức ngôn ngữ:**
  - Nếu  $a$  chia hết cho 2 thì  $a$  là số chẵn
- **Mức thuật toán:**
  - Nếu  $a$  chia 2 dư 0 thì in ra là  $a$  chẵn
- **Mức lập trình:**
  - `if ((a%2)==0) { cout << "A là số chẵn"; }`



Phần 3

# Các loại lệnh lựa chọn

# Các loại lệnh lựa chọn

---

- Có nhiều quan điểm phân loại lệnh lựa chọn
- Thường chia làm 3 loại:
  - Có làm việc A hay không?
  - Chọn làm 1 trong 2 việc
  - Chọn làm 1 trong nhiều việc
- Nếu muốn chọn n việc trong số m việc thì sao?
  - Không có câu lệnh cho tình huống này
  - Chia nhỏ thành những câu lệnh đơn giản (thuộc một trong 3 loại trên)

# Ví dụ về “làm việc A hay không”

---

```
if (delta < 0) {  
    cout << "Phuong trinh vo nghiem." << endl;  
}
```

```
if (delta == 0) {  
    cout << "Phuong trinh co nghiem duy nhat:" << endl;  
    cout << "    X = " << -b/2/a << endl;  
}
```

```
if (delta > 0) {  
    cout << "Phuong trinh co 2 nghiem phan biet:" << endl;  
    cout << "    X1 = " << (-b - sqrt(delta))/2/a << endl;  
    cout << "    X2 = " << (-b + sqrt(delta))/2/a << endl;  
}
```

# Ví dụ “chọn 1 trong 2 việc”

---

```
if ((a*a + b*b) == c*c) {  
    cout << "Tam giac ABC vuong tai C" << endl;  
}  
else {  
    cout << "Goc C khong phai goc vuong" << endl;  
}  
  
if (nhietdo < 10) {  
    cout << "Hom nay hoc sinh duoc nghi hoc!" << endl;  
}  
else {  
    cout << "Hom nay hoc sinh di hoc binh thuong!" << endl;  
}
```

# Ví dụ “chọn 1 trong nhiều việc”

---

```
switch (thang) {  
    case 1:  
    case 3:  
    case 5:  
    case 7:  
    case 8:  
    case 10:  
    case 12: ngay = 31; break;  
    case 4:  
    case 6:  
    case 9:  
    case 11: ngay = 30; break;  
    default: ngay = 28;  
}
```

Phần 4

# Câu lệnh `if` và `if-else`

# Câu lệnh `if`

---

- Cú pháp:

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

- Quá trình thực hiện:

1. Kiểm tra <điều kiện>, nếu sai thì bỏ qua
2. Thực hiện <công việc>

- Chú ý:

- Khối <điều kiện> là một biểu thức logic
- Nếu <công việc> chỉ có một lệnh duy nhất thì có thể không cần viết cặp ngoặc nhọn { }

# Câu lệnh if-else

---

- Cú pháp:

```
if (<điều kiện>) {  
    <công việc A>  
}  
else {  
    <công việc B>  
}
```

- Quá trình thực hiện:

1. Kiểm tra <điều kiện>, nếu sai thì xuống bước 3
2. Thực hiện <công việc A> sau đó xuống bước 4
3. Thực hiện <công việc B>



# Một số ví dụ đơn giản

---

```
if ((a * b) < 0) cout << "A & B trái dấu";
```

```
if (0 == (a % b))  
    cout << "A chia hết cho B";  
else  
    cout << "A không chia hết cho B";
```

```
if (delta == 0) {  
    x = -b/2/a;  
    cout << "Có một nghiệm x = " << x;  
}
```

Phần 5

# Kết hợp lựa chọn và vòng lặp

# Kết hợp lựa chọn và lặp

---

- Vòng lặp và lựa chọn có kết hợp với nhau làm chương trình rất uyển chuyển và mạnh mẽ
- Ví dụ: in ra màn hình các số dương chia hết cho 3 hoặc 7 nhỏ hơn 100.
- Cách làm: duyệt các số từ 1 đến 100, in ra nếu nó chia cho 3 hoặc 7 dư 0.

- Đoạn mã:

```
for (int i = 1; i <= 100; i++)  
    if ((i % 3 == 0) || (i % 7 == 0))  
        cout << i << endl;
```

Phần 6

# Câu lệnh `switch`

# Câu lệnh switch

---

- Một số bài toán có quá nhiều lựa chọn:
  - Sử dụng liên tiếp nhiều lệnh if
  - Chương trình dài
  - Dễ nhầm lẫn
- Ví dụ: Nhập vào chữ số X (từ 0 đến 9), hãy in ra màn hình tên (tiếng Việt) của chữ số đó
  - Nhập số 0: in ra “không”
  - Nhập số 1: in ra “một”
  - ...
- Cách làm duy nhất là liệt kê mọi tình huống

# Nhập chữ số X và in tên ra

---

// cách chạy chậm hơn: viết ngắn hơn

```
if (x == 0) cout << "Khong";
```

```
if (x == 1) cout << "Mot";
```

...

```
if (x == 9) cout << "Chin";
```

// cách chạy nhanh hơn: viết dài hơn

```
if (x == 0) cout << "Khong";
```

```
else
```

```
    if (x == 1) cout << "Mot";
```

```
    else
```

...

```
        if (x == 9) cout << "Chin";
```

# Câu lệnh `switch`

---

- Ngôn ngữ C++ cung cấp cấu trúc lệnh `switch` để đơn giản hóa trường hợp nhiều lựa chọn
- Cú pháp:

```
switch (<biểu thức>) {  
    case <giá trị 1>: <công việc 1>; break;  
    case <giá trị 2>: <công việc 2>; break;  
    ...  
    case <giá trị n>: <công việc n>; break;  
    default: <công việc>;  
}
```

# Nhập chữ số X và in tên ra

---

```
switch (x) {  
    case 0: cout << “Không”; break;  
    case 1: cout << “Một”; break;  
    case 2: cout << “Hai”; break;  
    case 3: cout << “Ba”; break;  
    case 4: cout << “Bốn”; break;  
    case 5: cout << “Năm”; break;  
    case 6: cout << “Sáu”; break;  
    case 7: cout << “Bảy”; break;  
    case 8: cout << “Tám”; break;  
    case 9: cout << “Chín”; break;  
    default: cout << “X không phải là chữ số”;  
}
```



Phần 7

# Bài tập

# Bài tập

---

1. *Viết hàm `TamGiacDeu(a, b, c)` trả về `true` nếu tam giác `ABC` là tam giác đều và ngược lại.*
  - *Tương tự hãy viết các hàm `TamGiacCan`, `TamGiacVuong`*
2. *Nhập `n` và in ra các số chẵn nhỏ hơn `n`.*
3. *Nhập `a`, `b` và `n`. In các số nhỏ hơn `n` và chia `a` dư `b`.*
4. *Nhập `a`, `b` và `n`. In các số nhỏ hơn `n` chia cho `a` và `b` đều dư 1.*
5. *Viết hàm `isPrime(int x)` - kiểm tra xem `x` có là số nguyên tố không?*