

CÁC CẤU TRÚC ĐIỀU KHIỂN

BUỔI 5- CÂU LỆNH ĐIỀU KHIỂN



- Sau khi học xong buổi học, sinh viên có khả năng:
 - Hiểu và vận dụng được các cấu trúc điều khiển để viết được chương trình trên máy tính



1. Khái niệm câu lệnh và khối lệnh trong lập trình
2. Phạm vi hoạt động của biến trong các khối lệnh
3. Giới thiệu về cấu trúc điều khiển
4. Cấu trúc rẽ nhánh
 1. Cấu trúc rẽ nhánh if
 2. Cấu trúc rẽ nhánh if-else
 3. Cấu trúc rẽ nhánh switch-case
5. Một số ví dụ minh họa

1. Khái niệm câu lệnh và khối lệnh trong lập trình



- Câu lệnh

- Một câu lệnh (statement) xác định một công việc mà chương trình phải thực hiện để xử lý dữ liệu đã được mô tả và khai báo.
- Các câu lệnh được ngăn cách với nhau bởi dấu chấm phẩy (;).
- VD:

```
int n;  
cout<< "Nhap vao so nguyen n = ";  
cin>>n;  
cout<< "So n= "<<n;
```

- Khối lệnh

- Một dãy các câu lệnh được bao bởi các dấu { } gọi là một khối lệnh.

```
{  
    int n;  
    cout<< "Nhap vao so nguyen n = ";  
    cin>>n;  
    cout<< "So n= "<<n;  
}
```

2. Phạm vi hoạt động của biến trong các khối lệnh



- Tất cả các biến mà chúng ta sẽ sử dụng đều phải được khai báo trước.

```
void main()  
{  
    cout<<x;    // Error identifier 'x' is undefined  
               // Build: error C2065: 'x' : undeclared  
    identifier  
    int y=5;  
    cout<<y;    // 5  
}
```

- Phạm vi hoạt động của một biến chính là khối lệnh mà nó được khai báo.
- Nếu nó được khai báo trong một hàm tầm hoạt động sẽ là hàm đó, còn nếu được khai báo trong vòng lặp thì tầm hoạt động sẽ chỉ là vòng lặp đó.

2. Phạm vi hoạt động của biến trong các khối lệnh



- Ví dụ:

```
#include <iostream>
using namespace std;
int x=3;
void main()
{
    cout<<x;    // 3
    int x=5;
    {
        cout<<x;    // 5
        int x=7;
        cout<<x;    // 7
        cout<<::x;  // 3
    }
    cout<<x;    // 5
    cout<<::x;   // 3
}
```

3. Giới thiệu về cấu trúc điều khiển



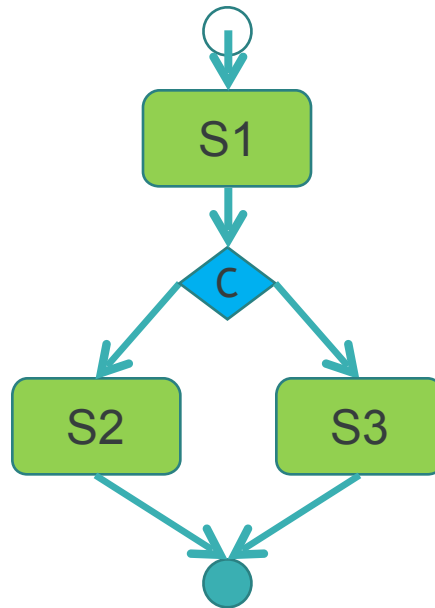
Có 3 loại cấu trúc điều khiển các lệnh cơ bản:

- **Cấu trúc tuần tự** : là cách tổ chức các lệnh thành từng khối.
Phần cấu trúc khối lệnh đã được trình bày trong chương 1.
- **Cấu trúc rẽ nhánh**: có các cấu trúc if và switch.
- **Cấu trúc lặp** : có các cấu trúc for, while, do while.

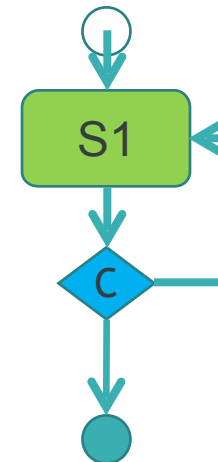
3. Giới thiệu về cấu trúc điều khiển



Cấu trúc tuần tự



Cấu trúc rẽ nhánh



Cấu trúc lặp



4. Các cấu trúc rẽ nhánh

- Cấu trúc rẽ nhánh có thể chia làm hai loại:
 - Cấu trúc rẽ một trong hai nhánh : như cấu trúc if, if..else và lệnh (? :).
 - Cấu trúc rẽ một, hai hoặc nhiều nhánh : cấu trúc switch..case.
- Trong hai cấu trúc này thì cấu trúc hai nhánh tổng quát hơn vì nó có thể áp dụng cho mọi loại biểu thức điều kiện rẽ nhánh và cấu trúc này cho phép lồng nhau để tạo thành các cấu trúc rẽ nhiều nhánh. Còn cấu trúc rẽ nhiều nhánh switch chỉ có thể áp dụng với biểu thức điều kiện rẽ nhánh với các giá trị rời rạc.

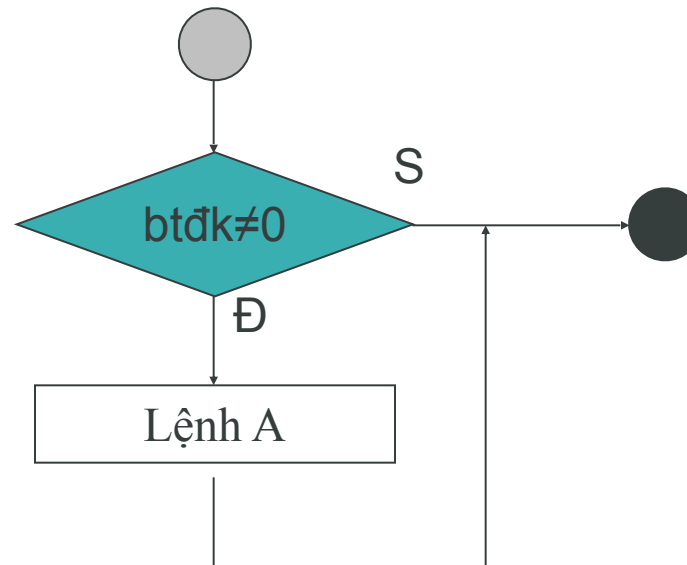
4.1 Cấu trúc rẽ nhánh if



- Cú pháp:

if (btđk)
 Lệnh A ;

- Lưu đồ:

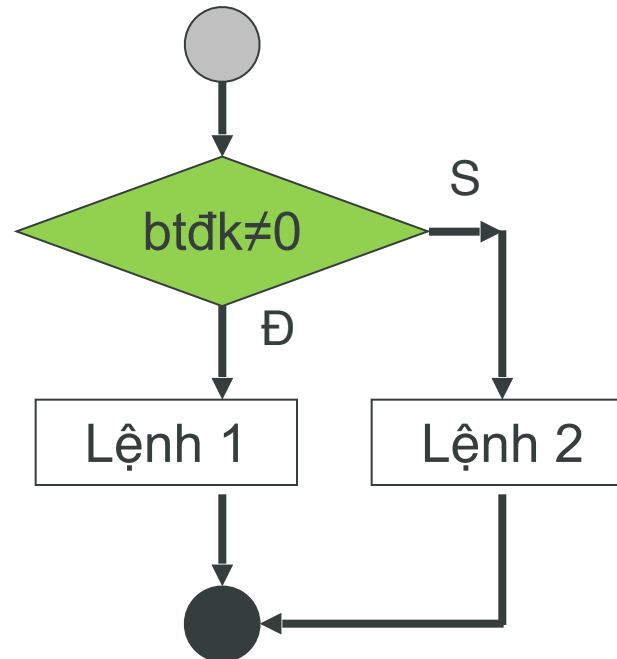


4.2 Cấu trúc rẽ nhánh if..else..

- Cú pháp:

```
if (btđk)  
    lệnh 1 ;  
else  
    lệnh 2 ;
```

- Lưu đồ:



Ví dụ minh họa



Viết chương trình tìm giá trị bé nhất của ba số a, b, c cho trước.

```
#include <iostream>
using namespace std;
void main(){
    int  a = 10, b = 15, c= 8;
    int  m;

    //Cách 1
    m = a;
    if (b < m) m = b;
    if (c < m) m = c;

    cout << "Gia tri be nhat m =" << m;
} // end main
```

Ví dụ minh họa



Viết chương trình tìm giá trị bé nhất của ba số a, b, c cho trước.

```
#include <iostream>
using namespace std;
void main(){
    int a = 10, b = 15, c = 8;
    int m;

    //Cách 2
    if (a < b)
        if (a < c) m = a;
        else m = c;
    else
        if (b < c) m = b;
        else m = c;

    cout << "Gia tri be nhat m =" << m;
} // end main
```

Ví dụ minh họa



Viết chương trình tìm giá trị bé nhất của ba số a, b, c cho trước.

```
#include <iostream>
using namespace std;
void main(){
    int a = 10, b = 15, c = 8;
    int m;

    //Cách 3
    m = (a < b) ? ((a < c) ? a : c) : ((b < c) ? b : c);

    cout << "Gia tri be nhat m =" << m;
} // end main
```



4.3 Cấu trúc rẽ nhánh switch..case..

- Cú pháp:

```
switch (biểu thức điều kiện) {  
    case hằng số 1 : câu lệnh 1  
    case hằng số 2 : câu lệnh 2  
    ...  
    [default : câu lệnh default]  
}
```

- **Biểu thức điều kiện** là một biểu thức số học nhận giá trị nguyên.
- Hằng số 1, hằng số 2,... là các hằng số chọn kiểu số nguyên khác nhau, tương ứng cho các nhánh chọn *case* khác nhau. Đây là các hằng số mà giá trị biểu thức điều kiện có thể nhận.
- Nhánh *default* là nhánh lựa chọn mặc định khi không có nhánh nào khác được chọn. Nhánh này là không bắt buộc phải có.



4.4 Cấu trúc rẽ nhánh switch..case..

- Ví dụ:

```
#include <iostream>
using namespace std;
void main()
{
    char ch;
    cout << "Nhap gia tri ch=";
    cin >> ch;
    switch (ch)
    {
        case 'a': cout << "Ki tu a da duoc nhap";
                break;
        case 'b': cout << "Ki tu b da duoc nhap";
                break;
        default: cout << "Ki tu khac a va b da duoc
nhap";
    }
}
```




4.4 Cấu trúc rẽ nhánh switch..case..

- Ví dụ:

```
#include <iostream>
using namespace std;
void main()
{
    int a;
    printf("Nhap a: ");
    scanf("%d", &a);

    switch (a)
    {
        case 1 : printf("Mot"); break;
        case 2 : printf("Hai"); break;
        case 3 : printf("Ba"); break;
        default : printf("Ko biet doc");
    }
}
```

Bài tập bắt buộc



1. Viết chương trình Giải phương trình bậc hai $ax^2+bx+c=0$
2. Viết chương trình Tính tiền đi taxi từ số km nhập vào. Biết: 1 km đầu giá 15000đ, từ km thứ 2 đến km thứ 5 giá 13500đ, từ km thứ 6 trở đi giá 11000đ, nếu trên 120km được giảm 10% trên tổng số tiền.
3. Viết chương trình Nhập vào tháng và năm (năm > 1975), kiểm tra tính hợp lệ của tháng, năm và cho biết tháng đó có bao nhiêu ngày.
4. Viết chương trình kiểm tra năm nhuận.
5. Viết chương trình nhập vào số nguyên 3 chữ số (từ 100 - 999), sau đó in ra các chữ số thuộc hàng trăm, hàng chục, hàng đơn vị.