***Cấu trúc điều khiển***

**Một số cấu trúc lập trình cơ bản:**

* Cấu trúc tuần tự: Là cấu trúc trong đó các câu lệnh sẽ đc thực hiện tuần tự lần lượt từ trên xuống dưới.
* Cấu trúc điều kiện: Là cấu trúc dung để kiểm tra điều kiện của chương trình. Nếu điều kiện là đúng hoặc thỏa mãn thì thân của cấu trúc điều kiện sẽ đc thực hiện, nếu điều kiện là sai hoặc k thỏa mãn thân của cấu trúc điều kiện sẽ đc bỏ qua. Chương trình sẽ thực hiện câu lệnh ở phía sau cấu trúc điều kiện.
* Cấu trúc lặp: Dùng để thực hiện lặp đi lặp lại các câu lệnh trong chương trình

\* **Cấu trúc lựa chọn ( If )**

- Cấu trúc if là 1 cấu trúc đk dung để kiểm tra 1 điều kiện trong chương trình. Biểu thức điều kiện của if là 1 biểu thức logic hoặc quan hệ. Nếu điều kiện của if thỏa mãn chương trình sẽ thực hiện thân của if , nếu đk của if là sai hoặc k thỏa mãn chương trình sẽ bỏ qua if thực hiện câu lệnh phía sau

**if ( expression ) {**

**//statement;**

**}**

**Chú ý: Nếu đk của if là đúng sau khi thực thi xong than của if chương trình sẽ thực hiện tiếp các câu lệnh phía sau if**

**\* Cấu trúc if – else:**

- Là cú pháp đầy đủ của cấy trúc if, tính đến cả 2 trường hợp. Nếu đk là đúng chương trình sẽ thực hiện khối if, nếu đk là sai chương trình sẽ thực hiện khối else.

- Câu trúc if – else còn đc gọi là cấu trúc rẽ nhánh vì chương trình sẽ thực hiện 1 trong 2 nhánh

***If…else if…else***

* Là cú pháp mở rộng của cấu trúc if – else trong đó sẽ có nhiều nhánh nhiều điều kiện để kiểm tra. Đầu tiên biểu thức điều kiện của if sẽ đc đánh giá nếu điều kiện của if là true thì chương trình sẽ thực hiện thân của if và bỏ qua tất cả nhánh còn lại, nếu điều kiện của if là sai chương trình sẽ lần lượt đánh giá điều kiện của các nhánh else if theo kí tự từ trên xuống dưới
* Nếu biểu thức đk của nhánh else if nào đó là đúng chương trình sẽ thực hiện than của nhánh đó và bỏ qua tất cả nhánh còn lại. Nếu điều kiện của tất cả các nhánh else if đều k thỏa mãn. Chương trình sẽ thực hiện nhánh else
* Nhánh else đại diện cho tất cả trg hợp còn lại
* Cấu trúc if - else if đc sử dụng trong trg hợp ta cần đánh giá nhiều điều kiện
* Mỗi nhánh else if sẽ gán vs 1 biểu thức điều kiện.
* Cấu trúc if – else if còn gọi là cấu trúc **“if else bậc 2”**

***If Lồng ( Nested if ):*** là cấu trúc if có chứa cấu trúc if khác, trong phương pháp này, if bên trong là 1 phần của bên ngoài, if bên trong sẽ phụ thuộc vào if bên ngoài nếu đk của if bên ngoài là đúng thì if bên trong mới có cơ hội đc đánh giá

***Cấu trúc switch - case :*** Là cấu trúc điều kiện dung để so sánh giá trị biến hoặc 1 biểu thức với 1 danh sách các giá trị cho trước. Cấu trúc switch – case đc sử dụng nếu như chương trình có nhiều nhánh mỗi nhánh là 1 case và ta chỉ muốn so sánh bằng

* Cơ chế hoạt động: gtrị của biến trong switch sẽ lần lượt đc so sánh với gtrị của từng case theo thứ tự từ trên xuống dưới
* Nếu gtrị của switch = gtrị của case nào đó thì chương trình sẽ thực hiện than của case đó và bỏ qua các case còn lại
* Nếu gtrị của tất cả các case đều không khớp với biến trong switch thì chương trình sẽ thực hiện nhánh default thể hiện tất cả các trường hợp còn lại

[ ***Chú ý!*** ]

* Trong mỗi 1 case cần có lệnh break. Để có thể kết thuc case đó
* Lệnh Break dung để thoát khỏi các case, nếu không có lệnh break chương trình sẽ thực thi các lệnh case tiếp theo cho đến khi gặp lệnh break

***Cấu trúc lặp:***

* Là một cấu trúc lập trình cho phép máy tính thực hiện lặp đi lặp lại các câu lệnh trong chương trình
* Cấu trúc lặp đc sử dụng khi ta cần tự động hóa việc tính toán với số lượng lớn mà k thể tính thủ công
* Cấu trúc lặp còn đc gọi là vòng lặp

**[ Chú ý ]** các vòng lặp sẽ đc thực hiện dưới 1 điều kiện, vòng lặp sẽ chỉ thực hiện chừng nào điều kiện còn đúng hoặc thỏa mãn, biểu thức điều kiện của vòng lặp là 1 biểu thức quan hệ hoặc login và sẽ trả về 1 giá trị Boolean. Khi sử dụng vòng lặp ta nên khai báo các biến đếm để kiểm soát vòng lặp. Ta cần tránh các vòng lặp vô hạn

***Biến đếm:*** là biến đc khai báo trong các vòng lặp và giá trị sẽ thay đổi sau mỗi lần lặp, biến đếm đc dung để kiêm soát hoặc kết thúc vòng lặp

* Điều kiện của vòng lặp sẽ đc kiểm tra trc mỗi lần lặp, nếu đkiện còn đúng, vòng lặp sẽ tiếp tục thực hiện, nếu đkiện là sai vòng lặp sẽ kết thúc
* Vòng lặp for: là vòng lặp đc sử dụng khi ta đã bt trc số lần lặp. Vòng lặp for bao gồm 3 câu lệnh.

+ Lệnh thứ nhất: lệnh khởi tạo dung để khai báo và khởi tạo các biến đếm, lệnh này sẽ đc thực hiện duy nhất 1 lần trc khi thực hiện vòng lặp

+ Lệnh thứ hai: Điều kiện: Lệnh này sẽ đc kiểm tra trc mỗi lần lặp

+ Lệnh thứ ba: Lệnh tăng giảm: Dùng để thay đổi giá trị của biến đếm sau mỗi lần lặp

* Vòng lặp lồng nhau: Là vòng lặp chứa vòng lặp khác, ứng vs mỗi lần lặp của vòng lặp bên ngoài, vòng lặp bên trong sẽ thực hiện n lần
* Vòng lặp while: là vòng lặp đc sử dụngkhi ta chưa bt trc số lần lặp, vòng lặp ưhile sẽ thực hiện chừng nào đkiện còn đúng

While ( condition )

**[ Chú ý ]** Khi thực hiện vòng lặp while ta cần tang giảm giá trị của biến

* ***Vòng lặp do – while***: là vòng lặp sẽ đc thực hiện chừng nào điều kiện còn đúng, được sử dụng khi chưa bt trc số lần lặp. Vòng lặp do – while giống vòng lặp while nhưng có 1 sự khác biệt:
* Vòng lặp while kiểm tra điều kiện trc rồi mới thực hiện vòng lặp. Còn vòng lặp do – while thực hiện vòng lặp trc r mới ktra điều kiện.
* Nếu điều kiện sai ngay từ đầu, vòng lặp while sẽ k thực hiện lần nào còn vòng lặp do – while sẽ thực hiện ít nhất 1 lần
* ***Một số câu lệnh điều khiển vòng lặp:***

+ Lệnh return dung để thoát ra khỏi 1 hàm và thoát ra trả về gtrị của hàm đó

+ Lệnh break dung để kết thúc vòng lặp khi 1 chương trình gặp lệnh break nó sẽ thoát ra khỏi vòng lặp hiện tại và chuyển đến câu lệnh ngay sau vòng lặp

+ Lệnh continue bỏ qua lần lặp hiện tại và chuyển đến lần lặp tiếp theo trong lần lặp, khi 1 chương trình gặp lệnh continue nó sẽ bỏ qua các câu lệnh còn lại của lần lặp hiện tại và chuyển đến lần lặp tiếp theo

+ Lệnh goto dung để chuyển 1 chương trình từ đoạn code này sang đoạn đc chỉ định

K nên sử dụng lệnh goto vì có thể khiên phá vỡ chương trình