**Struct**

1. Function1:

* Struct là 1 kiểu dữ liệu do người dùng tự định nghĩa.
* Ta có thể gộp nhiều kiểu dữ liệu vào trong 1 struct.

Vd:

Struct test{

Int a;

String b;

Char ch;

Float f;…..

};

* Trong 1 struct ta có thể thực hiện nhiều chức năng.

1. Syntax:

Struct <dataname>{

Statement;

};

1. Function2:

* Contructor:

+) Chức năng này cho phép ta khai báo 1 hàm có cùng tên với kiểu dữ liệu struct, khi ta khai báo kiểu dữ liệu struct thì nó sẽ tự động vào constructor.

+) Syntax: struct <typename>

{

Typename(parameters)

{

Statement;

}

};

+) Ta có thể nạp chồng các constructor bằng cách khai báo khác tham số và truyền vào tham số theo từng cách khai báo.

* Function:

+) Trong struct ta có thể dùng các function.

+) Syntax: struct <dataname>

{

Int a;

Void nhap(int a)

{

Cin>>a;

}

};

+) Sử dụng hàm như bên ngoài, gọi hàm bằng cách <dataname>.<function name>(parameters).

* Operator overloading:

+) Vì struct là kiểu dữ liệu người dùng tự định nghĩa, nên sẽ không có các hàm nguyên mẫu như +, -, \*, /…, ta cần phải overloading lại để định nghĩa cách sử dụng các operator.

+) syntax:

C1: <data name> operator <operator>(parameters(expect element of struct is called))

{

Statements;

}

C2: friend <dataname> operator <operator>(parameters(full element))

{

Statements;

}

+) Có thể nạp chồng hầu như các toán tử trừ: toán tử .(point), ::, ?:, sizeof().

+) Với toán tử nhập xuất thì, nhập <dataname>= istream&, xuất thì ostream&.

+) Nên giữ nguyên ý nghĩa của các toán tử.

+) Thứ tự ưu tiên của các toán tử sẽ không đổi.

* Ta có thể khai báo struct trong struct được. không vấn đề.