**Đặc tính của lập trình hướng đối tượng**

Có 4 đặc tính quan trọng của lập trình hướng đối tượng trong C++ mà chúng ta cần

nắm vững sau đây.

**Inheritance (Tính kế thừa )** trong lập trình hướng đối tượng có ý nghĩa, một class có thể kế thừa các thuộc tính của một class khác đã tồn tại trước đó.

Khi một class con được tạo ra bởi việc kế thừa thuộc tính của class cha thì chúng ta sẽ gọi class con đó là **subclass trong C++**, và class cha chính là **superclass trong C++**.

**Abstraction (Tính trừu tượng)** trong lập trình hướng đối tượng là một khả năng mà chương trình có thể bỏ qua sự phức tạp bằng cách tập trung vào cốt lõi của thông tin cần xử lý.

Điều đó có nghĩa, bạn có thể xử lý một đối tượng bằng cách gọi tên một phương thức và thu về kết quả xử lý, mà không cần biết làm cách nào đối tượng đó được các thao tác trong class.

Ví dụ đơn giản, bạn có thể nấu cơm bằng nồi cơm điện bằng cách rất đơn giản là ấn công tắc nấu, mà không cần biết là bên trong cái nồi cơm điện đó đã làm thế nào mà gạo có thể nấu thành cơm.

**Polymorphism (Tính đa hình)** trong lập trình hướng đối tượng là một khả năng mà một phương thức trong class có thể đưa ra các kết quả hoàn toàn khác nhau, tùy thuộc vào dữ liệu được xử lý.

Ví dụ đơn giản, cùng là một class quản lý dữ liệu là các con vật, thì hành động sủa hay kêu của chúng được định nghĩa trong class sẽ cho ra kết quả khác nhau, ví dụ nếu là con mèo thì kêu meo meo, còn con chó thì sủa gâu gâu chẳng hạn.

**Encapsulation (Tính đóng gói)** trong lập trình hướng đối tượng có ý nghĩa không cho phép người sử dụng các đối tượng thay đổi trạng thái nội tại của một đối tượng, mà chỉ có phương thức nội tại của đối tượng có thể thay đổi chính nó.

Điều đó có nghĩa, dữ liệu và thông tin sẽ được đóng gói lại, giúp các tác động bên ngoài một đối tượng không thể làm thay đổi đối tượng đó, nên sẽ đảm bảo tính toàn vẹn của đối tượng, cũng như giúp dấu đi các dữ liệu thông tin cần được che giấu.

Ví dụ đơn giản, khi bạn dùng một cái iphone, bạn không thể thay đổi các cấu trúc bên trong của hệ điều hành iOS, mà chỉ có Apple mới có thể làm được điều này thôi.

**Namespace là gì?**

**Tình huống:**

Khi đang lập trình trong một file A bạn include 2 file B và C, nhưng 2 file này có

cùng định nghĩa một hàm function() giống nhau về tên và tham số truyền vào, nhưng

xử lý của mỗi hàm ở mỗi file là khác nhau, vấn đề đặt ra là code làm sao để trình

biên dịch hiểu được khi nào bạn muốn gọi function của file B, khi nào bạn muốn gọi function của file C. Khi gọi hàm function() ở file A, trình biên dịch sẽ không biết được hàm function() bạn muốn gọi là hàm được định nghĩa ở file B hay file C. Vì vậy trình biên dịch chương trình sẽ báo lỗi.

**Định nghĩa:**

Namespace là từ khóa trong C++ được sử dụng để định nghĩa một phạm vi nhằm

mục đích phân biệt các hàm, lớp, biến, ... cùng tên trong các thư viện khác nhau.

**Template trong C++ là gì?**

• *Template* (khuôn mẫu) là một từ khóa trong C++, và là một kiểu dữ liệu trừu

tượng tổng quát hóa cho các kiểu dữ liệu int, float, double, bool...

• Template trong C++ có 2 loại đó là function template & class template.

• Template giúp người lập trình định nghĩa tổng quát cho hàm và lớp thay vì

phải nạp chồng (overloading) cho từng hàm hay phương thức với những kiểu

dữ liệu khác nhau.