1/

**-public**, **protected** và **private** đề cập đến khả năng truy cập của các phương thức.

-Mặc định, tất cả phương thức đều ở trạng thái **public**. Nếu không chỉ định khả năng truy cập của phương thức, nó sẽ là **public**.

-Phương thức **protected** và **private** không thể truy cập một cách tự do, và do đó khi có một thể hiện của đối tượng, bạn sẽ không thể gọi được các phương thức đó.

-Phương thức **protected** và **private** cũng có sự khác biết xung quanh cách bạn có thể sử dụng chúng trong ngữ cảnh của đối tượng.

2/

-Khi một class có các data fields là private, bạn buộc phải thay đổi chúng qua một công cụ public nào đó, ở đây là các functions set() và get() được khai báo tại vùng public, điều này giúp ta kiểm soát việc thay đổi và sử dụng các private data members của class, không cho phép một chương trình khác không thuộc class có quyền thay đổi và sử dụng các dữ liệu này.

3/

-Tính đóng gói: Có thể gói dữ liệu (data, ~ biến, trạng thái) và mã chương trình (code, ~ phương thức) thành một cục gọi là lớp (class) để dễ quản lí.

-Tính trừu tượng

-Tính thừa kế

-Tính đa hình: Đối tượng có thể thay đổi kiểu (biến hình)

4/

-Đối tượng trong thế giới thực là một thực thể hay vật mà chúng ta có thể thấy hoặc sờ được. Do đó đối tượng sẽ có hai tính chất đó là **thuộc tính** và **hành động**. **Hướng đối tượng** ý muốn nói rằng lấy một đối tượng cụ thể làm **trung tâm**, khi đó các thuộc tính và hành động sẽ xoay quanh nó. Nói đến đối tượng là nói đến từng trường hợp **cụ thể riêng biệt**

**-Lớp là tập hợp các đối tượng có cùng thuộc tính và hành động** (trong lập trình gọi là phương thức)

5/

-Một class **contructor** là một hàm thành viên đặc biệt của một lớp mà được thực thi bất cứ khi nào chúng ta tạo các đối tượng mới của lớp đó.

-Một **destructor** là một hàm thành viên đặc biệt của một lớp mà được thực thi bất cứ khi nào một đối tượng của lớp đó ra khỏi phạm vi hoặc bất cứ khi nào biểu thức delete được áp dụng tới một con trỏ tới đối tượng của lớp đó.

**Constructor mặc định** không phải là một **constructor**, nó là một khái niệm nhằm nói về loại hàm tạo không cần truyền đối số, được mặc định sử dụng khi ta khai báo một đối tượng của lớp. Cần tránh hiểu nhầm **Constructor mặc định** là **Constructor không đối số**. Vì đôi khi, một **Constructor có đối số mặc định**cũng được sử dụng như **Constructor mặc định** (tất cả đối số của hàm tạo đều có giá trị mặc định).

7/

-Chương trình lỗi khi khởi tạo: Array M1; Do class Array thiếu hàm khởi tạo mặc định.

-Lỗi ko thể truy xuất biến M2.A[3] do class khai báo mảng A ở thuộc tính private.

class Array

{

public:

int A[100];

int n;

public:

Array() {};

Array(int n)

{

//

}

//

};

void main()

{

Array M1();

Array M2(10);

cout << M2.A[3];

}

8/ thiếu constructor, destructor và hàm nhập

class A

{

private:

int i;

public:

A();

~A();

A(int i);

};

A::A()

{

}

A::A(int i)

{

cin >> i;

}

A::~A()

{

}

void main()

{

A a1;

A b2(5);

}