LÝ THUYẾT OOP

Câu 1: Lập trình hướng đối tượng là gì?

- Lập trình hướng đối tượng là phương pháp lập trình dựa trên kiến trúc lớp (class) và đối tượng (object).

- Uu điểm của lập trình hướng đối tượng:
  - ✓ Nguyên lý kế thừa: tránh lặp, tái sử dụng.
  - ✓ Nguyên lý đóng gói – che dấu thông tin: chương trình an toàn không bị thay đổi bởi những đoạn chương trình khác
  - ✓ Dễ mở rộng, nâng cấp
  - ✓ Mô phỏng thế giới thực tốt hơn.

Câu 2: Các đặc điểm quan trọng của lập trình hướng đối tượng? (4 đặc tính)

❖ Các đặc điểm quan trọng của lập trình hướng đối tượng:

- ✓ Các lớp đối tượng – Classes – Trừu tượng hóa: Cách nhìn khái quát hóa về một tập các đối tượng có chung các đặc điểm được quan tâm (và bỏ qua những chi tiết không cần thiết).

- ✓ Đóng gói – Encapsulation: Nhóm những gì có liên quan với nhau vào làm một,để sau này có thể dùng một cái tên để gọi đến. Đóng gói để che một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không nhìn thấy.

- ✓ Thừa kế – Inheritance: Là cơ chế cho phép một lớp D có được các thuộc tính và thao tác của lớp C, như thể các thuộc tính và thao tác đó đã được định nghĩa tại lớp D. Cho phép cài đặt nhiều quan hệ giữa các đối tượng: Đặc biệt hóa – Tổng quát hóa.

- ✓ Đa hình – Polymorphism: Là cơ chế cho phép một tên thao tác hoặc thuộc tính có thể được định nghĩa tại nhiều lớp và có thể có nhiều cài đặt khác nhau tại mỗi lớp trong các lớp đó.

Câu 3: Lớp đối tượng là gì? Đối tượng là gì?

- Lớp là một mô tả trừu tượng của nhóm các đối tượngcùng bản chất, ngược lại mỗi một đối tượng là một thểhiện cụ thể cho những mô tả trừu tượng đó.
- Đối tượng là cái ta tạo (từ một lớp) tại thời gian chạy.

Câu 4: Phân biệt các phạm vi truy xuất private, protected, public?(mô hình truy xuất)

- Các thuộc tính và phương thức được khai báo bên trong phạm vi private của một lớp thì chỉ được phép truy xuất từ bên trong lớp và không được quyền truy xuất từ bên ngoài lớp. Hơn nữa, các thuộc tính và phương thức này không cho phép kế thừa ở lớp dẫn xuất.

- Các thuộc tính và phương thức được khai báo bên trong phạm vi protected của một lớp thì chỉ được phép truy xuất từ bên trong lớp và không được quyền truy xuất từ bên ngoài lớp. Hơn nữa, các thuộc tính và phương thức này cho phép kế thừa ở lớp dẫn xuất.

- Các thuộc tính và phương thức được khai báo bên trong phạm vi public của một lớp thì được phép truy xuất từ bên trong lớp, bên ngoài lớp. Hơn nữa, các thuộc tính và phương thức này cho phép kế thừa ở lớp dẫn xuất.

Câu 5: Constructor là gì? Constructor mặc định?

- Khái niệm Constructor: là một phương thức của lớp dùng để tạo dựng một đối tượng mới. Hệ điều hành sẽ cấp phát bộ nhớ cho đối tượng và gọi đến hàm tạo. Hàm tạo sẽ khởi gán giá trị cho các thuộc tính của đối tượng.

- Các đặc điểm của Constructor:
  - ✓ Tên phương thức thiết lập trùng với tên lớp.
  - ✓ Không có giá trị trả về (no return type).
  - ✓ Được tự động gọi thực hiện ngay khi đối tượng được khai báo.
  - ✓ Có thể có nhiều phương thức thiết lập trong 1 lớp.
  - ✓ Trong một quá trình sống của đối tượng thì chỉ có 1 lần duy nhất một phương thức thiết lập được gọi thực hiện mà thôi đó là khi đối tượng ra đời.

- Constructor mặc định (default constructor): là constructor được gọi khi thể hiện được khai báo mà không có đối số nào được cung cấp. Ngược lại, nếu tham số được cung cấp tại khai báo thể hiện, chương trình sẽ gọi constructor khác theo cơ chế overload.

Câu 6: Destructor là gì?

- Khái niệm Destructor: Hàm hủy (destructor) là một phương thức của lớp dùng để thực hiện một số công việc có tính "dọn dẹp" trước khi đối tượng được hủy bỏ.

- Các đặc điểm của Destructor:
  - ✓ Tên phương thức trùng với tên lớp nhưng có dấu ngã ở đằng trước.
  - ✓ Không có giá trị trả về.
  - ✓ Không có tham số đầu vào.
  - ✓ Được tự động gọi thực hiện khi đối tượng hết phạm vi sử dụng.
  - ✓ Phương thức phá huỷ thuộc nhóm các phương thức xử lý.
  - ✓ Có và chỉ có duy nhất một phương thức phá huỷ trong 1 lớp.
  - ✓ Trong một quá trình sống của đối tượng có và chỉ có một lần phương thức phá hủy được gọi thực hiện mà thôi.

Câu 7: Kế thừa là gì?

- Kế thừa là một đặc điểm của ngôn ngữ lập trình hướng đối tượng dùng để biểu diễn mối quan hệ đặc biệt hóa – tổng quát hóa giữa các lớp.

- Ưu điểm của kế thừa trong lập trình:
  - ✓ Kế thừa cho phép xây dựng lớp mới từ lớp đã có.
  - ✓ Kế thừa cho phép tổ chức các lớp chia sẻ mã chương trình chung, nhờ vậy có thể dễ dàng sửa chữa, nâng cấp hệ thống.
  - ✓ Trong C++, kế thừa còn định nghĩa sự tương thích, nhờ đó ta có cơ chế chuyển kiểu tự động.
- Ví dụ:

```
class TamGiac
{
        …
};
class TamGiacCan : public TamGiac {…};
```

Câu 8: Phân biệt các kiểu kế thừa private, protected, public?

- Kế thừa public: Lớp con kế thừa public từ lớp cha thì các thành phần protected của lớp cha trở thành protected của lớp con, cac thành phần public của lớp cha trở thành public của lớp con. Nói cách khác mỗi thao tác của lớp cha được kế thừa xuống lớp con. Vì vậy ta có thể sử dụng thao tác của lớp cha cho đối tượng thuộc lớp con.
- Kế thừa protected: Lớp con kế thừa protected từ lớp cha thì các thành phần protected và public của lớp cha trở thành protected của lớp con. Nói cách khác mỗi thao tác của lớp cha đều được lớp con bảo vệ. Vì vậy ta có thể sử dụng thao tác của lớp cha cho đối tượng thuộc lớp con.
- Kế thừa private: Lớp con kế thừa private từ lớp cha thì các thành phần protected và public của lớp cha trở thành private của lớp con. Nói cách khác mỗi thao tác của lớp cha đều bị lớp con che dấu. Vì vậy trên quan điểm của thế giới bên ngoài lớp con không có các thao tác mà lớp cha có.

	private	protected	public
private	||	||	||
protected	private	protected	protected
public	private	protected	public

Câu 9: Đa hình là gì?

- Đa hình: Là hiện tượng các đối tượng thuộc các lớp khác nhau có khả năng hiểu cùng một thông điệp theo các cách khác nhau.
- Ưu điểm của Đa hình:
  - ✓ Không phải viết lại mã hoặc lớp đã có sẵn.
  - ✓ Dùng một tên duy nhất để lưu trữ biến của nhiều kiểu dữ liệu khác nhau (float, double, long, int,…).
  - ✓ Tạo ra sự linh hoạt trong việc sử dụng tính thừa kế để nâng cấp, phát triển chương trình.

- Ví dụ minh họa: Nhận được cùng một thông điệp "nhảy", một con kangaroo và một con cóc nhảy theo hai kiểu khác nhau: chúng cùng có hành vi "nhảy" nhưng các hành vi này có nội dung khác nhau.

Câu 10: Lớp trừu tượng là gì? Phương thức thuần ảo là gì?

- Hàm thuần ảo (Phương thức ảo thuần tuý) có ý nghĩa cho việc tổ chức sơ đồ phân cấp các lớp, nó đóng vai trò chừa sẵn chỗ trống cho các lớp con điền vào với phiên bản phù hợp. Phương thức ảo thuần tuý là phương thức ảo không có nội dung, được khai báo với từ khóa virtual và được gán giá trị =0.
- Khi lớp có phương thức ảo thuần tuý, lớp trở thành lớp cơ sở trừu tượng. Lớp cơ sở trừu tượng không có đối tượng nào thuộc chính nó.
- Ví dụ minh họa:

```
class Shape
{
        public :
        virtual void draw() = 0;
}
```

- ✓ Trong ví dụ trên, các hàm thành phần trong lớp Shape là phương thức ảo thuần tuý và lớp Shape là lớp cơ sở trừu tượng. Nó bảo đảm không thể tạo được đối tượng thuộc lớp Shape.

Câu 11: Trình bày về lớp cơ sở trừu tượng. Lớp cơ sở trừu tượng được cài đặt như thế nào?

- Trình bày lớp cơ sở trừu tượng:
  - ✓ Lớp cơ sở trừu tượng là lớp cơ sở không có đối tượng nào thuộc chính nó.
  - ✓ Ta có thể thay thế cho nội dung không có ý nghĩa bằng phương thức ảo thuần tuý. Phương thức ảo thuần tuý là phương thức ảo không có nội dung.
  - ✓ Khi lớp có phương thức ảo thuần tuý, lớp trở thành lớp cơ sở trừu tượng. Ta không thể tạo đối tượng thuộc lớp cơ sở thuần tuý.

- Cài đặt của lớp cơ sở trừu tượng được cài đặt trong C++:

```
class Shape
{
        public :
        virtual void draw() = 0;
```

```
}
class Rectangle : public Shape
{
        public:
        void draw()
{
```

Câu 12: Hãy trình bày những đặc điểm của tính đóng gói (encapsulation) trong lập trình hướng đối tượng. Trường hợp nào có thể vi phạm tính đóng gói? Cho ví dụ minh họa.

- Đặc điểm của tính đóng gói
  - ✓ Đóng gói: Nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến.
  - ✓ Che dấu thông tin: đóng gói để che một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không nhìn thấy. − Trường hợp có thể vi phạm tính đóng gói: Khai báo các thuộc tính trong lớp trong phạm vi từ khóa private.
- Ví dụ minh họa:

```
class PhanSo
 {
        public:
        int ts, ms;
 };
```

Câu 13: Cho biết ý nghĩa và mục đích của các hàm get/set trong một lớp

- Ý nghĩa:
  - ✓ Phương thức get/set chỉ có nhiệm vụ cho ta đọc/ghi giá trị cho các thành viên dữ liệu.
- Mục đích:
  - ✓ Ngoài việc bảo vệ các nguyên tắc đóng gói, ta cần kiểm tra xem giá trị mới cho thành viên dữ liệu có hợp lệ hay không.
  - ✓ Sử dụng phương thức truy vấn cho phép ta thực hiện việc kiểm tra trước khi thực sự thay đổi giá trị của thành viên.
  - ✓ Chỉ cho phép các dữ liệu có thể truy vấn hay thay đổi mới được truy cập đến.

Câu 14: Phân biệt khái niệm overload (tải chồng) và override (ghi đè)

	Override	Overload
Khái niệm	Là một tính năng cho phép một lớp con hoặc lớp con cung cấp một triển khai cụ thể của một phương thức đã được cung cấp bởi một trong các lớp siêu hoặc các lớp cha của nó. Nói cách khác, nếu lớp con cung cấp trình triển khai cụ thể của phương thức mà đã được cung cấp bởi một trong các lớp cha của nó, thì đó là ghi đè phương thức.	Nạp chồng phương thức đơn giản là có vài phương thức trùng tên nhưng khác nhau về đối số. Cài chồng phương thức cho phép ta tạo nhiều phiên bản của một phương thức, mỗi phiên bản chấp nhận một danh sách đối số khác nhau, nhằm tạo thuận lợi cho việc gọi phương thức.
Hành vi	Thay đổi hành vi hiện tại của phương thức.	Thêm hoặc mở rộng cho hành vi của phương thức.
Đa hình	Thể hiện tính đa hình tại run time.	Thể hiện tính đa hình tại compile time.
Danh sách tham số	Danh sách tham số phải giống nhau.	Danh sách tham số khác nhau (số lượng, thứ tự, kiểu dữ liệu)
Giá trị trả về	Kiểu trả về bắt buộc phải giống nhau.	Kiểu trả về có thể khác nhau.
Phạm vi	Xảy ra giữa 2 class có quan hệ kế thừa	Xảy ra trong phạm vi cùng 1 class.