* **Lập trình hướng đối tượng**: kỹ thuật lập trình dựa trên các đối tượng.
  + Các đối tượng là sự vật, sự việc được trừu tượng hóa thành các cấu trúc dữ liệu ( class ) bao gồm các thuộc tính và phương thức để có thể lưu trữ và tính toán trên máy tính.
* **Trừu tượng hóa (Abstraction)**: chọn lọc, xác định các đặc điểm và hành vi tương đồng nhau của những sự vật, sự việc trong thực tế thành cấu trúc dữ liệu (class) mà máy tính có thể lưu trữ, tính toán. VD: em và anh đều có hành vi chung là ăn ngủ code và CCCD, em chọn ra 3 hành vi và thuộc tính tuổi đó đế tạo thành class lập trình viên.
* **Đóng gói (Encapsulation)**: sử dụng phạm vi (mqh giữa các class là cha con, ông cháu) kết hợp phương thức truy cập (public, protected,...) để quản lý việc truy xuất và sửa đổi thông tin của các đối tượng => mã nguồn minh bạch, an toàn hơn.
* **Đa hình (Polymorphism):** cùng một hành động tồn tại nhiều nơi trong nhiều đối tượng nhưng cách thức thực hiện khác nhau.
* Overload: cùng tên hàm nhưng khác tham số truyền vào và trả về
* Override: khác body, mã bên trong.
* **Abstract class**: giống class thông thường trừ:
  + astract method không có thân.
  + không thể triển khai trực tiếp.
* **Interface:** không có biến, chỉ định nghĩa hàm, đa kế thừa
* **Dependency injection:** class này phải phụ thuộc vào chức năng của class khác mới có thể chạy:
  + Có thể thay đổi, sửa chữa code tại thời điểm runtime.
* **IEnumerable:** 
  + Các kiểu dữ liệu danh sách thường implement từ IEnumrable giúp duyệt foreach ( lặp tuần tự )
  + Là giao diện.
* **IQueryable:** 
  + chuyển đổi truy vấn LINQ thành ngôn ngữ truy vấn cụ thể tới nguồn dữ liệu. Tức là không truy vấn lập tức, chỉ truy vấn khi gọi ToList(), First(), Count().
  + Còn các dạng dữ liệu collection khác thì tải dữ liệu từ CSDL lên vào bộ nhớ rồi truy vấn luôn.
* **List/Array:** lưu trữ data trong bộ nhớ một cách liền mạch => truy cập data với chỉ số.
  + List thay đổi kích thước linh hoạt còn array thì không.