

Đặc điểm	Class Thường (Normal/Instance Class)	Static Class
Mục đích	Mô hình hóa một thực thể có trạng thái và hành vi (ví dụ: Car, Employee).	Chứa các hàm tiện ích (utility) hoặc dữ liệu toàn cục, không có trạng thái cụ thể.
Khởi tạo	Phải được tạo bằng new (ví dụ: Car myCar = new Car();).	Không thể tạo đối tượng (instance).
Bộ nhớ	Mỗi đối tượng có một bản sao riêng của các trường (fields) dữ liệu.	Dữ liệu được chia sẻ và duy nhất trong suốt chương trình (Shared/Global).
Kế thừa	Có thể kế thừa từ class khác và bị class khác kế thừa.	Không thể kế thừa và không thể bị kế thừa.
Phương thức	Chứa cả phương thức thường và tĩnh.	Chỉ chứa các phương thức tĩnh.
Ví dụ	String, List<T>, chef (trong code của bạn).	System.Math, System.Console, Convert.
Khi dùng	Khi bạn cần tạo nhiều bản sao độc lập của một thực thể, mỗi bản sao có dữ liệu riêng.	Khi bạn cần một bộ sưu tập các hàm hỗ trợ hoặc hằng số không liên quan đến dữ liệu của đối tượng nào.
Đặc điểm	Phương thức Thường (Instance Method)	Phương thức Tĩnh (Static Method)
Từ khóa	public void MakePizza()	public static void SayHi(string name)
Sở hữu	Thuộc về một Đối tượng (Instance) cụ thể.	Thuộc về chính Class.
Cách gọi	Qua đối tượng: myChef.MakeSalad()	Qua Class: Usefultool.SayHi("MP")
Truy cập Dữ liệu	Có thể truy cập dữ liệu thường (instance fields) và tĩnh (static fields) của lớp.	Chỉ có thể truy cập dữ liệu tĩnh (static fields) của lớp.
Ví dụ	myCar.Accelerate(), myChef.MakeChicken() (cần biết con gà này thuộc về đầu bếp nào).	Math.Sqrt(25), Console.WriteLine(), Usefultool.SayHi() (chỉ thực hiện một hành động chung).
Khi dùng	Khi hành động cần thay đổi trạng thái của một đối tượng cụ thể hoặc cần sử dụng dữ liệu riêng của đối tượng đó.	Khi hành động là độc lập với bất kỳ đối tượng nào và chỉ thực hiện một tác vụ chung, tiện ích.
public class Car // Class Thường		
{		
// Dữ liệu Thường (Instance Field): Có thể truy cập bằng phương thức thường		
public string Color;		
public int Speed = 0;		
// Phương thức Thường (Instance Method): Thay đổi trạng thái của đối tượng cụ thể		
public void Accelerate()		
{		
Speed += 10;		
Console.WriteLine(\$"The {Color} car is now at {Speed} km/h.");		
}		
}		
// Cách sử dụng		
Car myCar = new Car { Color = "Red" };		
myCar.Accelerate(); // Output: The Red car is now at 10 km/h.		
Car yourCar = new Car { Color = "Blue" };		
yourCar.Accelerate(); // Output: The Blue car is now at 10 km/h.		
// Hai đối tượng hoạt động độc lập.		
public static class Calculator		
{		
// Dữ liệu Tĩnh (Static Field): Được chia sẻ bởi mọi nơi gọi class này		
public static readonly double PI = 3.14159;		
// Phương thức Tĩnh (Static Method): Chỉ thực hiện tác vụ chung		

public static double CalculateCircleArea(double radius)	
{	
// Có thể truy cập dữ liệu tĩnh (PI)	
return PI * radius * radius;	
}	
}	
// Cách sử dụng	
// Không cần khởi tạo đối tượng	
double area = Calculator.CalculateCircleArea(5); // Gọi trực tiếp qua tên Class	
Console.WriteLine(\$"Area: {area}");	
double piValue = Calculator.PI; // Truy cập trực tiếp qua tên Class	