

# C#AssignmentDay09Part02

## 1- LinkedIn article about class types ?

23:23 2.00 3% ← ⋮

Mahmoud Elsy >You · Full stack .Net Web Developer Trainee @ Digital Egypt now • 5

الـ Class في C # هي البنية الأساسية لـ Object-Oriented (Object-Oriented). هي عبارة عن الـ Blueprint اللي ممكن الـ Objects Properties والـ Methods من المبنية منه تستخدمها.

إيه هو الـ Class بالضبط؟

الـ Class عبارة عن نموذج بوصف الـ data والـ methods مع بعض، يعني ثاني، لو عايزين نعمل نموذج لـ printer، ممكن نعرف الـ Model properties فيها Printer Class Scans ()Print ذي methods create أي عدد من الـ objects اللي تتمثل printers فففففة.

أنواع الـ Classes في C

1. Concrete Class

ده النوع اللي تقدر تعمل منه Objects مباشرة، بيعتني على properties و methods.

مثال:

```
class Car
{
    public string Brand;
    public void Drive() { Console.WriteLine("Car is driving"); }
}
```

هنا نقدر نعمل object :

```
Car myCar = new Car();
```

2. Abstract Class

ده نوع من الكلاسات مينفعش تعمل منه Object مباشرة، بيعتني على Abstract Methods لازم أي Class يورث منه يطبقها، ودول عاديّة ممكن تستخدمها الكلاسات الوراثة منه.

مثال:

```
abstract class Animal
{
    public abstract void MakeSound();
```

Add a comment...

23:23 5.00 3% ← ⋮

```
public abstract void MakeSound();
public void Eat() { Console.WriteLine("Eating"); }
```

أي Class يورث من الـ Animal لازم يعرف .()MakeSound.

3. Sealed Class

ده كلاس مينفعش أي Class تاني يورث منه، غالباً بستخدمه لو عايزين تحمي الكود من التعديل عن طريق الوراثة.

مثال:

```
sealed class Utility
{
    public void DoWork()
    { Console.WriteLine("Working"); }
```

أي محاولة لعمل Utility : Error هتدى.

4. Static Class

كلس كل عناصره Static مينفعش تعمل منه Objects. بستخدمه لو عايزين مجموعة ثابتة أو Constants methods.

مثال:

```
static class MathHelper
{
    public static int Add(int a, int b)
    { return a + b; }
```

نقدر نستخدمه مباشرة:

```
int sum = MathHelper.Add(5, 3);
```

5. Partial Class

ده نوع من الكلاسات بيتقسم على files مختلفة، مفيدي لو الشغل على project كبير أو لو الكود بيتوارد تلقائي (مثال: Windows Forms).

مثال:

```
partial class Person
{
    public string Name;
}
```

وافي file تاني:

```
partial class Person
```

Add a comment...

23:23 5.00 3% ← ⋮

وفي file تاني:

```
partial class Person
{
    public void Greet() { Console.WriteLine("Hello " + Name); }
```

الخلاصة: الـ Classes هي الأساس لـ Object-Oriented Inheritance، النوع المحظوظ من الكلاسات يعطيك تحكم في الـ Reusability، Code Protection

Show translation

### Types of Classes in C#

Concrete Class
Regular class that can be instantiated. Example: class Car { public void Drive(); }

Abstract Class
Cannot be instantiated, used for inheritance. Example: abstract class Animal { public abstract void MakeSound(); }

Sealed Class
Cannot be inherited or extended. Example: sealed class Utility { public void DoWork() { ... } }

Static Class
Contains only static members. Example: static class MathHelper { public static int Add(int a, int b) { ... } }

Partial Class
Allows multiple files to define parts of a single class. Example: partial class Person { }

Add a comment...

## **2) What we mean by Generalization concept using Generics ?**

Generalization means creating one reusable structure that works for multiple types instead of writing separate code for each type.

### **How Generics Achieve Generalization?**

- Instead of creating multiple versions of a class or method for int, string, double, etc., we write one generic class or method.
- The type parameter T acts as a placeholder for any type.

### **Benefits of Generalization:**

1. Code Reusability → Write once, use for any type.
2. Type Safety → Compiler checks types; no runtime casting.
3. Maintainability → Easier to update one generic implementation.
4. Flexibility → Works with value types, reference types, or user-defined types.

## **3)What we mean by hierarchy design in real business ?**

Hierarchy design means organizing entities into parent-child relationships to reflect “is-a” relationships or levels of specialization.

### **What It Looks Like in Business?**

1. Employees Example
  - Employee → base/general class
  - Manager → inherits from Employee
  - Developer → inherits from Employee

- Intern → inherits from Employee

## Benefits of Hierarchy Design:

1. **Reflects Real-world Structure** → Easy for business to relate objects.
2. **Code Reusability** → Common functionality in parent class.
3. **Polymorphism** → Use child objects via parent references.
4. **Maintainability & Extensibility** → Add new specialized types without changing existing code.