  
  
  
  
  
  
  
  
  
  


**Part01**

**Question: Why can't a struct inherit from another struct or class in C#?**

الـ struct في C# معمول عشان يكون حاجة خفيفة وصغيرة، ومافيهوش وراثة من struct أو class تاني، لأن الوراثة هتزود التعقيد وتكبر الحجم، وده عكس الهدف منه.

لكن عادي جدًا الـ struct يطبّق Interface، لأنه بيديله سلوك من غير ما يزود بيانات.

**Question: How do access modifiers impact the scope and visibility of a class member?**

، Access Modifiers هما اللي بيحددوا مين يقدر يشوف أو يستخدم حاجة جوه الكلاس:

public:   
أي حد يقدر يشوفها

private:  
 بس الكلاس نفسه يشوفها

Protected  
: الكلاس نفسه والكلاسات اللي وارثه منه

internal:  
 أي حاجة جوه نفس المشروع

protected internal:   
نفس المشروع أو كلاس وارث

private protected:  
 نفس المشروع + لازم يكون وارث

يعني هي زي بوابات بتتحكم مين يدخل يشوف إيه

**Question: Why is encapsulation critical in software design?**

لأن الكبسلة (Encapsulation) بتخلي البيانات والسلوكيات جوه الكلاس محمية ومش أي كود خارجي يقدر يغيرها أو يوصل لها بطريقة غلط، وده بيمنع الأخطاء ويحافظ على الكود منظم وسهل التعديل من غير ما يكسر أجزاء تانية من البرنامج.

**Question: what is constructors in structs?**

الـ constructor في الـ struct هو دالة بتشتغل أول ما تعمل نسخة من الـ struct، وظيفته يدي قيم مبدئية للمتغيرات.

وفي الـ struct لازم يكون فيه باراميترات، لأن فيه constructor افتراضي جاهز بيحط القيم الأساسية لوحده.

**Question: How does overriding methods like ToString() improve code readability?**

لما تعيد كتابة دوال زي ToString()، بتقدر تتحكم في شكل النص اللي بيظهر عن الكائن، بدل ما يطلع اسم النوع بس.

ده بيخلي الكود أوضح وأسهل في الفهم، لأنك هتشوف معلومات مفيدة عن الكائن مباشرة بدل بيانات غامضة.

**Question: How does memory allocation differ for structs and classes in C#?**

الـ struct بيتخزن في الـ stack وده بيخليه أسرع وأخف، أما الـ class بيتخزن في الـ heap وبيكون أبطأ شوية لأنه بيحتاج إدارة للذاكرة عن طريق الـ garbage collector.

الفرق الأساسي إن struct نوع قيمة (Value Type) و class نوع مرجع (Reference Type).

**Part02**

**What is copy constructor?**

الـ copy constructor هو نوع من الـ constructor بياخد نسخة من نفس النوع كـ باراميتر، ويعمل نسخة جديدة طبق الأصل من الكائن ده، بنفس القيم اللي فيه.

الغرض منه إنه يسهّل عمل نسخ مستقلة من الكائنات بدل ما تنسخ القيم يدويًا.

**What is Indexer, when used, as business mention cases u have to utilize it?**

الـ Indexer بيخلي الكائن يتعامل زي المصفوفة، فتقدر تجيب أو تغيّر بياناته باستخدام أقواس `[]`.

بنستخدمه في حالات زي:

* تجيب بيانات موظف برقم الـ ID
* توصل لإعداد في النظام باسم المفتاح
* تجيب سعر منتج برقم المنتج أو اسمه

**Summarize keywords we have learnt last lecture**

* \* struct
* \* class
* \* interface
* \* access modifiers (public, private, protected, internal)
* \* encapsulation
* \* constructor
* \* copy constructor
* \* overriding زي (ToString)
* \* memory allocation (stack, heap)
* \* indexer

**Part03 Bonus**

**1) Can Constructor be private? (Rare Business Case)**

أيوه، ممكن، وده بيحصل قليل في مواقف خاصة، زي لما تعمل Singleton أو تمنع أي حد من إنشاء كائن من الكلاس إلا من جوه الكلاس نفسه.

**2) BCL >> Override tostring()**

الـ BCL (Base Class Library) فيها كل الأنواع الأساسية في .NET، ومعظم الكلاسات بتعمل Override لدالة ToString() عشان تعرض بيانات مفهومة بدل اسم النوع بس. مثال: DateTime.ToString() بيعرض التاريخ بدل ما يكتب "System.DateTime".

**3) struct : u can't create less ctor in struct in .net 5 >> .net 6.0 or greater >> why ?**

* في .NET 5 كان الـ struct بياخد Constructor فاضي جاهز بيحط القيم الافتراضية، ومش مسموح تغيّره.
* في .NET 6 وفوق، بقوا يسمحوا تعمل Constructor فاضي بنفسك عشان تحط القيم اللي انت عايزها بدل القيم الافتراضية.