

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

Because a struct is a *Value Type*, its size is fixed, it already inherits from `System.ValueType`, and multiple inheritance is not supported in C#.

### How do access modifiers impact the scope and visibility of a class member?

They define who can access the member:

- **public** → Accessible by everyone.
- **private** → Accessible only within the same class.
- **protected** → Accessible within the class and its derived classes.
- **internal** → Accessible within the same project (assembly).
- **protected internal** → Accessible by derived classes *or* within the same project.

### Why is encapsulation critical in software design?

To protect data, prevent direct modification from outside the class, and control access through *properties* or *methods*.

### What is constructors in structs?

Special methods that run when a struct is initialized, and can take parameters to set initial values.

### How does overriding methods like `ToString()` improve code readability?

It allows objects to display clear, meaningful information instead of the default, less informative values.

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

- **struct** → Stored on the *stack* (or inside another object), faster.
- **class** → Stored on the *heap*, slightly slower but supports inheritance.

### What is a copy constructor?

A constructor that takes an object of the same type as a parameter to create a new object with the same values.

### Summarize keywords we have learned last lecture:

- **struct** → A value type that stores data on the stack.
- **constructor** → An initialization method for an object.
- **encapsulation** → Hiding data and controlling access to it.