Part01

1. Question Why does defining a custom constructor suppress the default constructor in C#?

   Once you created a custom ctor, compiler assumes you want full control over obj initialization

   Then u must declare a default ctor if you need it
2. Question: How does method overloading improve code readability and reusability?

   Readability: we can use the same name (e.g. Sum) for diff parameter sets.

   Reusability: we can use the same name for multiple functions.
3. Question: What is the purpose of constructor chaining in inheritance?

   Check if the base is properly initialized before the child adds its own logic
4. Question: How does new differ from override in method overriding?

   override: Provides polymorphic behavior. The derived method replaces the base method when accessed via a base reference.

   new: Hides the base method. The derived method is called only when accessed via the derived type reference, not polymorphically.
5. Question: Why is ToString() often overridden in custom classes?

   byDefault tostring returns the className => not informative

   so, overriding it provides a meaningful information
6. Question: Why can't you create an instance of an interface directly?

   Interfaces usually defines methods/properties with no implementation

   Then we must implement the Interface in a class then instantiate it.
7. Question: What are the benefits of default implementations in interfaces introduced in C# 8.0?

   - Allow interfaces to evolve without breaking existing implementations.
   - Provide shared logic across multiple classes.
   - Reduce boilerplate code by offering a fallback implementation.
   - Enable more flexible design while maintaining backward compatibility.
8. Question: Why is it useful to use an interface reference to access implementing class methods?

   - Promotes abstraction: You can work with different implementations interchangeably.
   - Supports **d**ependency injection and testing (mocking interfaces).

- Decouples code from specific classes, improving flexibility and maintainability.

9. Question: How does C# overcome the limitation of single inheritance with interfaces?
   Allow class to implement multiple inheritances, interfaces avoided diamond inheritance problems
10. Question: What is the difference between a virtual method and an abstract method in C#?
    Virtual: has a default implementation but can be overridden (optional)
    Abstract: has no implementation and derived classes must provide one (mandatory)

Part02

1. What is the difference between class and struct in C#?

	Class	Struct
Type	reference	value
Less ctor	✅	must initialize all attrs
inheritance	✅	❌
new	4 things	just select the ctor
AM	private, internal, public, protected, private protected, internal protected	private, internal, public

2. If inheritance is relation between classes clarify other relations between classes
   Aggregiation, composition, abstraction, polymarphism