* **Why does defining a custom constructor suppress the default constructor in C#?**  
  Because when you define a custom constructor, the compiler no longer provides the implicit default constructor automatically.
* **How does method overloading improve code readability and reusability?**  
  It allows multiple methods with the same name but different parameters, making code easier to read and reducing duplication.
* **What is the purpose of constructor chaining in inheritance?**  
  To ensure the base class is properly initialized before the derived class adds its own initialization.
* **How does new differ from override in method overriding?**  
  new hides the base method (no polymorphism), while override replaces the base method and supports polymorphism.
* **Why is ToString() often overridden in custom classes?**  
  To provide meaningful, human-readable information about the object instead of the default class name output.
* **Why can't you create an instance of an interface directly?**  
  Because an interface only defines a contract (methods/properties) without implementation, so it cannot be instantiated.
* **What are the benefits of default implementations in interfaces introduced in C# 8.0?**  
  They allow adding new methods to interfaces without breaking existing implementations and provide shared logic across classes.
* **Why is it useful to use an interface reference to access implementing class methods?**  
  It promotes abstraction, loose coupling, and allows working with different implementations interchangeably.
* **How does C# overcome the limitation of single inheritance with interfaces?**  
  A class can implement multiple interfaces, achieving multiple inheritance of behavior without inheriting multiple base classes.
* **What is the difference between a virtual method and an abstract method in C#?**  
  A virtual method has a default implementation but can be overridden, while an abstract method has no implementation and must be overridden.