|  |  |  |  |
| --- | --- | --- | --- |
| All about INTERFACES Interface in OOP simply allows you to build easily extendable and flexible software applications. A contract Interface is a set of abstract members (methods, properties, and events). It is a contract that when classes implement, they must provide implementation of all its members.  If the class or structure does not provide an implementation for a particular method, property, or event, it must throw a *NotImplementedException.* Purpose of the members The members are specific and depend on the exact behavior of its modelling. Thus, an interface expresses a model that any class can choose to support. |  | Scope The members of an interface cannot specify an access modifier. All interface members are implicitly public and abstract) |  |
|  | Beginning with C# 8.0, an interface may define default implementations for some or all of its members. A class or struct that implements the interface doesn't have to implement members that have default implementations. |  |
| Interface Vs Abstract | More on Interface..   * Multiple inherittance in C# is supportedd by interfaces and therefore a class can implement as many interfaces * A struct cannot inherit from another struct or a class. It must implement interface for inheritance * Interfaces can inherit from one or more interfaces. The derived interface inherits the members from its base interfaces. | | |
|