**Interfaces: -** It is basically used to achieve Multiple Inheritance which cannot be achieved by using Classes. A Class can inherit from Multiple Interfaces but cannot inherit from Multiple Classes. A class or struct can implement multiple interfaces and must define all the members of all interfaces.

* An Interface can contain Properties, Methods, Delegates, or Events but only the declaration and not the implementation.
* Interface methods are Public by default. You cannot explicitly use Public keyword for an interface method.
* Interface members cannot have an implementation. It should be provided by the Derived Classes.
* An Interface cannot be Instantiated i.e. we cannot create an object of an Interface; however, an Interface reference variable can point to a Derived Class object.
* An Interface cannot contain Fields.
* An Interface cannot contain virtual methods.
* It is compulsory for the Derived Classes to provide an implementation for the Interface members.
* An Interface can inherit from other interfaces but not from any classes.
* All the members of the interface must be implemented with the public modifier in a class or struct.
* Do not use public modifier with an explicit implementation. It will give compile time error. (e.g. interface\_name.member\_name)
* An interface cannot contain constructors.

