1. **What is an interface as the term is used on object-oriented programming?**

An interface is a programming structure/syntax that allows the computer to enforce certain properties on an object (class). It contains only the signatures of methods, properties, events or indexers. A class or struct that implements the interface must implement the members of the interface that are specified in the interface definition.

1. **How do you define an interface?**

Use the interface keyword.

interface Example  
{  
 int example1();  
}

1. **Can an interface have variables, fields, or properties?**

No, an interface can only have methods without statements.

1. **How do you define a method in an interface?**

You define a method in an interface with a return type, method name, parameters, semicolon.

1. **Can you instantiate an object through an interface? Why or why not?**

Yes; because a class and its objects inherit any methods specified by its interface.

1. **Using the new keyword, can you declare a reference to an interface?**

No, but you can assign an interface to a variable of an appropriate class.

1. **Can an object inherit from multiple interfaces? Can a class implement multiple interfaces? If so, how can it do so?**

A class is allowed to implement an unlimited number of interfaces even though it can have at most one base class. A class must implement all the methods declared by these interfaces.

1. **What does it mean to explicitly implement an interface?**

Explicitly implementing an interface means specifying which interface a method belongs to when you implement it.

1. **What are the restrictions on interfaces?**

You’re not allowed to define any fields in an interface, not even static fields. You’re not allowed to define any constructors in an interface. You’re not allowed to define a destructor in an interface. You cannot specify an access modifier for any method. You cannot nest any types inside an interface. An interface is not allowed to inherit from a structure or a class, although an interface can inherit from another interface.

1. **What is the difference between an abstract class and an interface?**

The main difference between them is that a class can implement more than one interface but can only inherit from one abstract class.

1. **What is an abstract method?**

An abstract method is similar in principle to a virtual method except that it does not contain a method body. Abstract method declarations are only permitted in abstract classes.

1. **What is a sealed class?**

Use the sealed keyword to prevent a class from being inherited as a base class.

1. **What is a sealed method?**

Use the sealed keyword to declare that an individual method in an unsealed class is sealed, this means that a derived class cannot override this method.