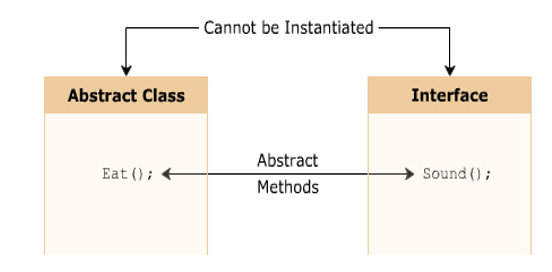
**SIMILARITIES BETWEEN ABSTRACT CLASSES AND INTERFACES IN C#**

* Abstract classes and interfaces both declare methods without implementing them.
* Both, abstract classes as well as interfaces, contain abstract methods.
* Abstract methods of both, the abstract class as well as the interface, are implemented by the inheriting subclass.
* Both, abstract classes as well as interfaces, can inherit multiple interfaces.
* Both abstract class and interfaces can have reference variables.



**DIFFERENTIATE BETWEEN ABSTRACT CLASSES AND INTERFACES IN C#**

|  |  |
| --- | --- |
| ABSTRACT CLASS | INTERFACE |
| An abstract class can inherit a class and multiple interfaces. | An interface can inherit multiple interfaces but cannot inherit a class. |
| An abstract class can have methods with a body. | An interface cannot have methods with a body. |
| An abstract class method is implemented using the override keyword. | An interface method is implemented without using the override keyword. |
| An abstract class is a better option when you need to implement common methods and declare common abstract methods. | An interface is a better option when you need to declare only abstract methods. |
| An abstract class can declare constructors and destructors. | An interface cannot declare constructors or destructors. |
| If you want to declare an abstract member in abstract class then abstract keyword is mandatory. | If you want to declare an abstract member in interface then abstract keyword is not mandatory because members of interface are abstract by default. |