**When to use an abstract class**

* An abstract class is a good choice if we are using the inheritance concept since it provides a common base class implementation to derived classes.
* An abstract class is also good if we want to declare non-public members. In an interface, all methods must be public.
* If we want to add new methods in the future, then an abstract class is a better choice. Because if we add new methods to an interface, then all of the classes that already implemented that interface will have to be changed to implement the new methods.
* If we want to create multiple versions of our component, create an abstract class. Abstract classes provide a simple and easy way to version our components. By updating the base class, all inheriting classes are automatically updated with the change. Interfaces, on the other hand, cannot be changed once created. If a new version of an interface is required, we must create a whole new interface.
* Abstract classes have the advantage of allowing better forward compatibility. Once clients use an interface, we cannot change it; if they use an abstract class, we can still add behavior without breaking the existing code.
* If we want to provide common, implemented functionality among all implementations of our component, use an abstract class. Abstract classes allow us to partially implement our class, whereas interfaces contain no implementation for any members.

**When to use an interface**

* If the functionality we are creating will be useful across a wide range of disparate objects, use an interface. Abstract classes should be used primarily for objects that are closely related, whereas interfaces are best suited for providing a common functionality to unrelated classes.
* Interfaces are a good choice when we think that the API will not change for a while.
* Interfaces are also good when we want to have something similar to multiple inheritances since we can implement multiple interfaces.
* If we are designing small, concise bits of functionality, use interfaces. If we are designing large functional units, use an abstract class.