Interface & Abstract Class Comparison

Cheat Sheet

	Class	Abstract Class	Interface
Instantiated	✔	✘	✘
Extend/Implement Multiple	✘	✘	✔
Contract	✘	✔	✔
Implementation	✔	✔	✘
Single Place Versioning	✔	✔	✘
All Methods Public (Access Modifiers)	✘	✘	✔
Properties	✔	✔	✘
Constants	✔	✔	✔
Overridable Constant	✔	✔	✘
Type Checked	✔	✔	✔

Abstract Class

When determining if an **abstract class** is right for your situation, you should consider the following:

- Do you want to share code among several closely related classes?
- Do you expect classes that extend your abstract class will have many common methods or properties?
- Do you require access modifiers other than public (such as protected and private)?

Interface

You should consider using **interfaces** if any of these statements apply to your situation:

- You expect that unrelated classes would implement your interface. For example, the interfaces savoryFlavor and Countable, are implemented by many unrelated classes.
- You want to specify the behavior of a particular data type, but are not concerned about how that behavior is implemented.
- You want to take advantage of multiple inheritance of type.