Inheriting from a class is a powerful mechanism, but the real power of inheritance comes from inher-

iting from an interface. An interface does not contain any code or data; it just specifies the methods

and properties that a class that inherits from the interface must provide. Using an interface makes it

possible for you to completely separate the names and signatures of the methods of a class from the

method’s implementation.

Abstract classes are similar in many ways to interfaces except that they can contain code and data.

However, you can specify that certain methods of an abstract class are virtual so that a class that in-

herits from the abstract class must provide its own implementation of these methods. You frequently

use abstract classes with interfaces, and together they provide a key technique with which you can

build extensible programming frameworks.

If a class implements an interface, the interface guarantees that the

class contains all the methods specified in the interface.

An interface cannot contain any data; you cannot add fields (not even private ones) to an interface.