public interface ISpeaks

{

string Speak();

}

public class Dog : Mammal, ISpeaks

{

public string Speak() { return "Woof!"; }

}

public class Person : Mammal, ISpeaks

{

public string Speak() { return "Hi!"; }

}

//Notice Telephone has a different abstract class

public class Telephone : Appliance, ISpeaks

{

public Telephone(Person p)

{

Person = p;

}

public Person { get; set; }

public string Speak() { p.Speak(); }

}

[Test]

public void Test\_Objects\_Can\_Speak()

{

List<ISpeaks> thingsThatCanSpeak = new List<ISpeaks>();

//We can add anything that implements the interface to the list

thingsThatCanSpeak.Add(new Dog());

thingsThatCanSpeak.Add(new Person());

thingsThatCanSpeak.Add(new Telephone(new Person()));

foreach(var thing in thingsThatCanSpeak)

{

//We know at compile time that everything in the collection can speak

Console.WriteLine(thing.Speak());

}

}