**C# Complex Data Structures**  
  
The .NET Framework includes various built-in data types, such as int, decimal, string, and Boolean. These data types can be identified as simple data types because they consist of a single, simple value such as a number, text value, or a true or false setting.  You could argue that String is not a simple data type, and to a certain extent, that is correct.  C/C++ and other similar languages consider a String to be a character array.  Because this module will cover arrays, we could agree with that and we do.  The key difference for C# is that we can use a String value as a simple data structure without worrying about accessing it as a character array, dealing with a pointer to the start of the array, or any other string functions found in C.  We could, because C# is flexible enough to allow it, but for the purpose of data types in C#, consider the String to be a simple data type.  
  
The complex data types are suited for scenarios where you need to store multiple items in a single entity. Consider a deck of cards, the days of the week, months of the year, and even more complex items such as an object in code to represent a car.  Each of these examples requires multiple values to reflect the concept.  We think of a week as a single entity that consists of seven days.  When we talk about a week, we do so with the inherent knowledge of what a week consists of.    
  
This module will focus on arrays, enums, and structs in C#.  We can represent the deck of cards as an array, days of the week as an enum, and the car as a struct. The more common way to represent complex "objects" such as a car, is to use object-orientation and class files.  You will find object-oriented programming concepts in modules 5 and 6.