Random Numbers

There are times when it is necessary to simulate some kind of random event in a program. For example, if you wanted a program to model a board game that uses dice to determine events in the game, you would have to simulate the roll of the dice. The number that comes up on a roll of the dice is random, because you can't predict what number will come up next. The .Net library has a class, ***Random***, built into it that generates Pseudo-Random Numbers. They are called pseudo-random numbers because they are not truly random. To generate random numbers, we need an object of the Random class. Create an object of the Random class by writing

```
Random randomNums = new Random( );
```

The Random class has a method called ***Next( )*** that returns a pseudo random number between 0 and 2,147,483,646. You can scale the values returned by Next( ) by including an integer argument when calling the method. For example,

```
int dieValue = randomNums.Next( 6 );
```

will return one of six values, starting at zero and going up to five. Since we need values for our dice between one and six, we will add a shift value when we call the Next method.

```
int dieValue = randomNums.Next( 1, 7 );
```

This form of the Next method returns pseudo-random numbers starting from one and going up to, but not including seven.

Example

Let's code up our dice example to illustrate how to generate random numbers within a range. The following code will generate 10 random dice throws:

```
using System;

class Program
{
   const int BOX = 6;

   static void Main( )
   {
      Random randomNums = new Random( );

       for (int i = 0; i < 10; i++)
      {
         int d1 = randomNums.Next(1, BOX+1);
         int d2 = randomNums.Next(1, BOX+1);
          Console.WriteLine("You rolled {0} and
{1}", d1, d2);
      }
   }
   Console.ReadLine( );
}
```