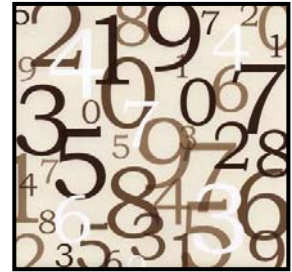


GENERATING RANDOM NUMBERS

Games, simulators, screen savers, and many other types of applications require random numbers. Visual Basic includes a **Random** class that generates a random number.

In order to use the **Random** class, the first thing you need to do is create a new **Random** object as follows:

```
'Declare and initialize Random object
Dim randomObject As New Random
```



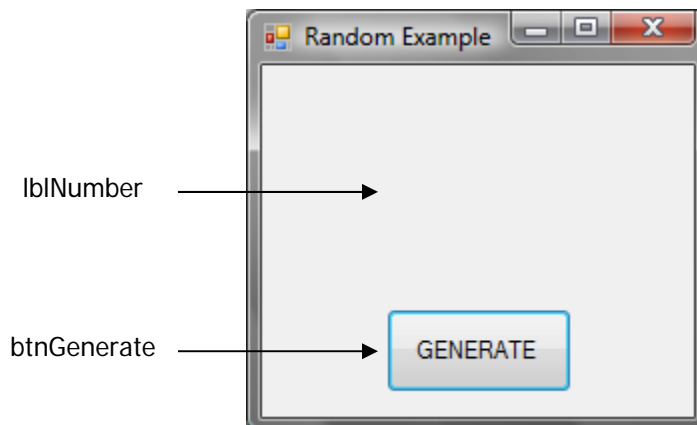
The next thing you'll need to do is declare a variable that will store the number that is randomly generated by the Random object that we created above:

```
'Declare and initialize variable to store random number
Dim randomNumber As Integer
```

To generate a random number, we will need to use the **Next()** method included in the **Random** class which generates a positive integer value greater than or equal to zero and less than 2,147,483,647.

```
'Generate and store random number
randomNumber = randomObject.Next()
```

Let's now add a user interface to see how this program would work. Create a form and add a label and button as follows:



Add the following code to the button:

```
Private Sub btnGenerate_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnGenerate.Click

    'Declare and initialize Random object
    Dim randomObject As New Random
```

```

'Declare and initialize variable to store random number
Dim randomNumber As Integer

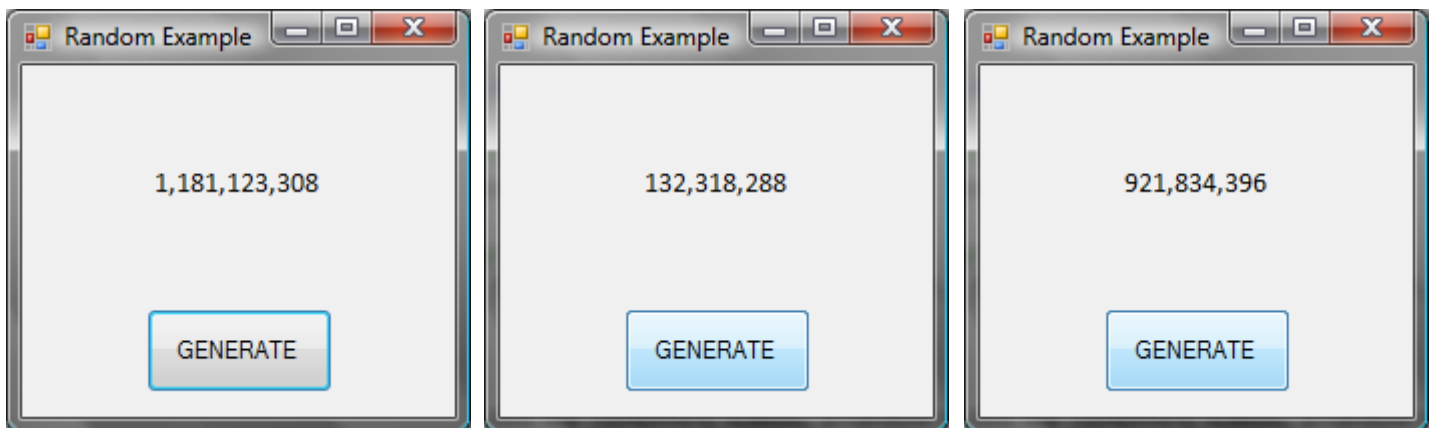
'Randomly generate and store random number
randomNumber = randomObject.Next()

'Output and format the random number
lblNumber.Text = FormatNumber(randomNumber, 0, , , TriState.True)

End Sub

```

When you run the program, and click the GENERATE button, a random number is outputted in the label.



SPECIFYING A RANGE OF NUMBERS

We can specify a range of numbers by passing a single argument to the **Next()** method. For example, if I wanted to generate a random number between 0 and 5, I would use the **Next()** method as follows:

```
randomNumber = randomObject.Next(6)
```

When a single argument is passed to the **Next()** method, the values returned by **Next()** will be in the range from 0 to (but not including) the value of that argument. So, in the above example, the program will generate one of six possible numbers in the range 0-5.

Let's say I wanted to generate a random number between 1 and 6. In this case, there are two ways we can do this. The first is simply pass a single argument to the **Next()** method to represent the amount of possible values the program will generate and simply add the minimum value to the equation, as follows:

```
randomNumber = 1 + randomObject.Next(6)
```

Alternatively, you can produce the same result by passing two arguments to the **Next()** method: the first representing the minimum value in our range and the second representing the maximum value + 1.

```
randomNumber = randomObject.Next(1, 7)
```

GENERATING FLOATING-POINT VALUES

The `Random()` class includes a `NextDouble()` method to generate a random floating-point value greater than or equal to 0 and less than 1.

```
randomNumber = randomObject.NextDouble()
```

To specify a range of values, such as 1.0 – 10.0, we will need to multiply the result of `NextDouble()` by the amount of possible integer values - 1 (i.e. 10 - 1) and then add the minimum value, as follows:

```
randomNumber = randomObject.NextDouble() * 9 + 1
```