

**TegoSoft Inc.**

Copyright © 1994-1996 TegoSoft Inc. ® All Rights Reserved

*P.O.Box 389, Bellmore, NY 11710*

Web Site: <http://www.tegosoft.com>

 e-mail: [tegosoft@msn.com](mailto:tegosoft@msn.com)


# Lesson 15 - Random Numbers


In this lesson you'll learn to generate random numbers.


## **Creating the Directory of the Project and Saving the Files of the Project**

As usual, you'll start by creating the directory of the project, and saving the files of the project.


 Create the **C:\VBMyProg\Lesson15** directory. You'll save the files of this lesson to this directory.

 Select **New Project** from the **File** menu of Visual Basic.

 Make the window of Form1 the selected window, select **Save File As** from the **File** menu of Visual Basic, and save the file as **Random.FRM** inside the **C:\VBMyProg\Lesson15** directory.

 Select **Save Project As** from the **File** menu of Visual Basic, and save the project as **Random.VBP** inside the **C:\VBMyProg\Lesson15** directory.


# Always Use Option Explicit

 Inside the general declarations section of Form1 type the following code:

**Option Explicit**


## Generate a Random Number Between 0 and 1

You'll now place a CommandButton inside Form1. When the user will click this button, a random number will be generated.

 Place a CommandButton inside Form1. Then set the properties of the CommandButton as follows:

**Name:** cmdGenerate1

**Caption:** Generate Number Between &0 and 1


 Place a Label control Form1. Then set the properties of the Label control as follows:

**Name:** lblMyLuckyNumber

**Caption:** Make it empty

**Alignment:** 2 - Center

**BorderStyle:** 1 - Fixed Single

 Arrange the **cmdGenerate1** button and the **lblMyLuckyNumber** label inside Form1 so that Form1 will look as shown in Figure 15.1.

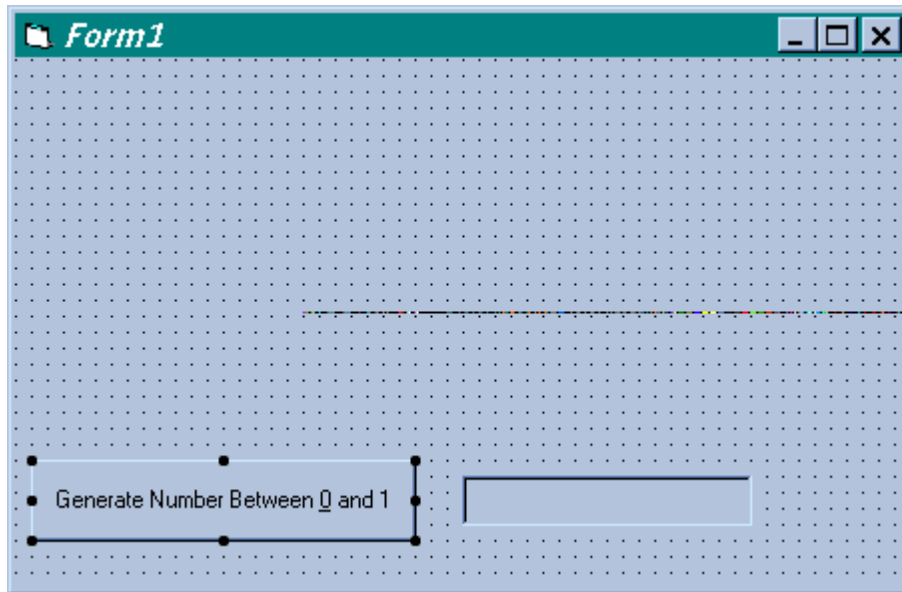



Figure 15.1. Form1 with the button and label in it.

## Attaching Code to the Click Event of the cmdGenerate1 Button

You'll now attach code to the **Click** event of the **cmdGenerate1** button.

 Type the following code inside the **cmdGenerate1\_Click()** procedure:

```
Private Sub cmdGenerate1_Click()  
  
Dim MyLuckyNumber  
  
' Initialize the random-number generator.  
Randomize  
  
MyLuckyNumber = Rnd()  
  
lblMyLuckyNumber.Caption = Str(MyLuckyNumber)
```

**End Sub**

The code that you typed is executed automatically whenever the user clicks the **cmdGenerate1** button.

You declared a variable called **MyLuckyNumber**:

```
Dim MyLuckyNumber
```

You then execute the **Randomize** statement:

```
Randomize
```

The **Randomize** statement initializes the random number generator mechanism.

You then execute the **Rnd()** function:

```
MyLuckyNumber = Rnd()
```

The **Rnd()** function returns a number that is equal or larger than **0**, but less than **1**. You assigned the returned value from the **Rnd()** function to the **MyLuckyNumber** variable.

Finally, you converted the **MyLuckyNumber** number to a string, and you updated the **Caption** property of the **lblMyLuckyNumber** label:


```
lblMyLuckyNumber.Caption = Str(MyLuckyNumber)
```


Let's see your code in action:



Select **Save Project** from the **File** menu of Visual Basic to save your work.

=====

 Select **Start** from the **Run** menu of Visual Basic to execute the Random program.

 Click the **Generate Number Between 0 and 1** button, and verify that a random number appears inside the label after each click.

 Experiment with the Random program, then terminate the program.



### **NOTE**

As you execute the Random program, you'll see that the generated number is greater or equal to **0**, and less than **1**.

For example you may see the generated numbers as follows:

0.5432123

0.1234567

and so on.

Visual Basic may also display the generated number in the following format:

1.234567E-2


The preceding number means 0.01234567

(E-2 means 10 to the power -2).

## **Generating Random Integers**

In your future projects you will probably need to generate random integers. Let's add a button to the Random program that lets you generate random integers.

=====

 Place a CommandButton inside Form1, and set its properties as follows:

**Name:** cmdGenerate2

**Caption:** Generate &Integer Between 0 and 15


 Place a Label control inside Form1, and set its properties as follows:

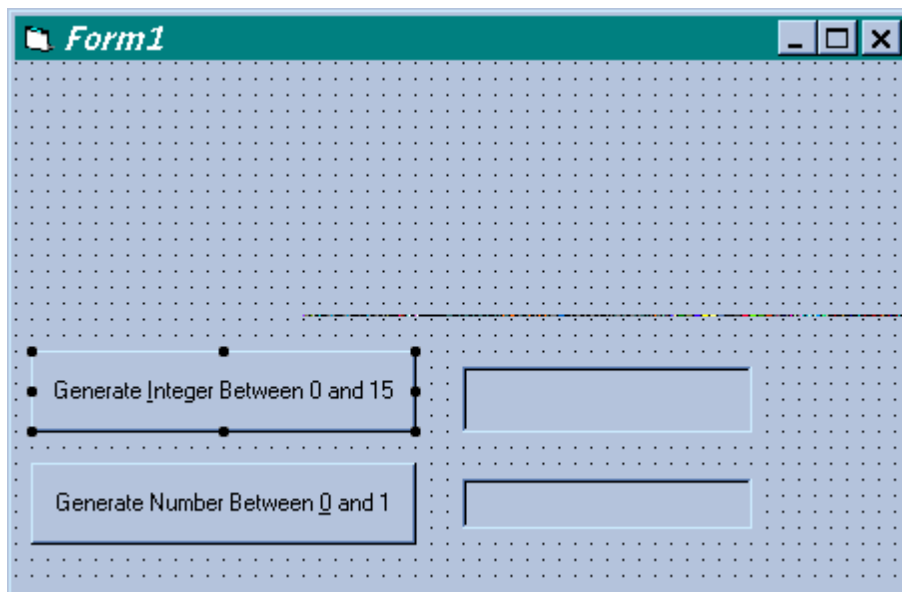
**Name:** lblOurLuckyInteger

**Caption:** Make it empty

**Alignment:** 2- Center

**BorderStyle:** 1-Fixed Single


 Size and drag the controls so that Form1 will look as shown in Figure 15.2.



**Figure 15.2.** Form1 with the second button and label in it.

# Attaching Code to the Click Event of the cmdGenerate2 Button

You'll now attach code to the **Click** event of the **cmdGenerate2** button, so that when the user clicks this button, a random integer will be generated:

 Type the following code inside the **cmdGenerate2\_Click()** procedure:

```
Private Sub cmdGenerate2_Click()  
  
Dim MaxNum, MinNum  
  
Dim OurLuckyInteger  
  
Randomize  
  
MaxNum = 15  
MinNum = 0  
  
OurLuckyInteger = _  
    Int((MaxNum - MinNum + 1) * Rnd() + MinNum)  
  
lblOurLuckyInteger.Caption = Str(OurLuckyInteger)  
  
End Sub
```

Let's go over the code that you typed.

You declared a few variables:

```
Dim MaxNum, MinNum  
Dim OurLuckyInteger
```

=====

You then initialize the random number generator:

```
Randomize
```

You want to generate an integer that is greater or equal to **0** and is less than or equal to **15**. So you set the following values for the **MaxNum** and **MinNum** variables:

```
MaxNum = 15
```

```
MinNum = 0
```

Then you generate the random integer as follows:

```
OurLuckyInteger = _  
    Int((MaxNum - MinNum + 1) * Rnd() + MinNum)
```

Recall that **Rnd()** returns a number greater or equal to **0** but less than **1**. Suppose that **Rnd()** returns **0**. In this case, the generated integer is:

```
Int( (15-0+1)*0 + 0) =  
Int( 16*0 + 0)=  
Int( 0 + 0 ) =  
Int ( 0 ) = 0
```

Suppose that **Rnd()** returns **0.5** In this case, the generated integer is:

```
Int( (15-0+1)*0.5 + 0) =  
Int( 16*0.5 + 0)=  
Int( 8 + 0 ) =  
Int ( 8 ) = 8
```

Suppose that **Rnd()** returns **0.95** In this case, the generated integer is:

```
=====
```

```
Int( (15-0+1)*0.95 + 0) =  
Int( 16*0.95 + 0)=  
Int( 15.2 + 0 ) =  
Int ( 15.2 ) = 15
```

After examining the preceding examples, you'll realize that the generated integer can be an integer number between **0** and **15**.



## NOTE

The **Int()** function returns the integer part of the number that is supplied to the parameter of the **Int()** function.

### Example:

`Int(4.999)` returns the integer 99.

Finally, you update the **Caption** property of the label with the generated random integer as follows:

```
lblOurLuckyInteger.Caption = Str(OurLuckyInteger)
```

Let's see your code in action:



Select **Save Project** from the **File** menu of Visual Basic to save your work.



Execute the Random program, click the **Generate Integer Between 0 and 15** button, and notice the integer that is displayed inside the label.



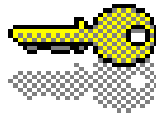
Experiment with the Random program, then terminate the program.



(b) False

2. What code will you use to generate random integer between 10 and 100?

## Answers to Exam



1. (a) True, the **Rnd()** function generates a random number that is greater or equal to **0**, and less than **1**.

2. You can use the following code:

```
Dim MaxNum, MinNum
```

```
Dim GeneratedInteger
```

```
Randomize
```

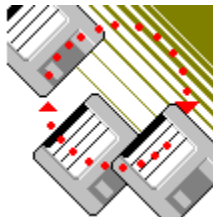
```
MaxNum = 100
```

```
MinNum = 10
```

```
GeneratedInteger = _
```

```
Int((MaxNum - MinNum + 1) * Rnd() + MinNum)
```

## Project



Modify the Random program so that it will make use of the generated integers as follows.

 Modify the **cmdGenerate2\_Click()** procedure so that it will look as follows:

```
Private Sub cmdGenerate2_Click()  
  
Dim MaxNum, MinNum  
  
Dim OurLuckyInteger  
  
Randomize  
  
MaxNum = 15  
MinNum = 0  
  
OurLuckyInteger = _  
    Int((MaxNum - MinNum + 1) * Rnd() + _  
    MinNum)  
  
lblOurLuckyInteger.BackColor = QBColor(OurLuckyInteger)  
lblOurLuckyInteger.Caption = Str(OurLuckyInteger)  
  
End Sub
```


You added the following statement:


```
lblOurLuckyInteger.BackColor = QBColor(OurLuckyInteger)
```

**QBColor()** is a function that represents a color. The color is specified in the parameter of the **QBColor()** function, and the parameter can be any integer between **0** and **15**. So the preceding statement sets the **BackColor** property of the label to a color that corresponds to the generated random integer.

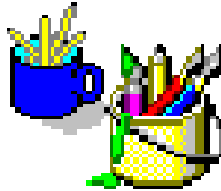
=====

 Select **Save Project** from the **File** menu of Visual Basic to save your work, and then execute the Random program.

 Click the **Generate Integer Between 0 and 15** button, and notice that the background color of the label changes to a random color.

 Experiment with the Random program, then terminate the program.

## Cosmetic Considerations



In the Project section of this section you modified the Random program so that the **BackColor** property of the **lblOurLuckyInteger** label changes according to the random integer that was generated.

The **ForeColor** property of the **lblOurLuckyInteger** is black. If the generated integer is **0**, the **BackColor** of the label is set to black because **QBColor(0)** represents the color black. So you will not be able to see the digit **0** inside the label because the digit color is black.

Let's add code that takes care of the situation when the generated random color is black:

 Modify the **cmdOurLuckyInteger\_Click()** procedure so that it will look as follows:

```
Private Sub cmdGenerate2_Click()
```

=====

```

Dim MaxNum, MinNum

Dim OurLuckyInteger

Randomize

MaxNum = 15
MinNum = 0

OurLuckyInteger = _
    Int((MaxNum - MinNum + 1) * Rnd() + _
    MinNum)

If OurLuckyInteger = 0 Then
    lblOurLuckyInteger.ForeColor = QBColor(15)
Else
    lblOurLuckyInteger.ForeColor = QBColor(0)
End If

lblOurLuckyInteger.BackColor = QBColor(OurLuckyInteger)
lblOurLuckyInteger.Caption = Str(OurLuckyInteger)

End Sub

```

You added the following **If...Else...End If** statements:

```


If OurLuckyInteger = 0 Then
    lblOurLuckyInteger.ForeColor = QBColor(15)
Else
    lblOurLuckyInteger.ForeColor = QBColor(0)
End If


```


=====

In the preceding statements, if the generated random number is black, the code under the **If** statement is executed, This code sets the **ForeColor** property of the label to white (**QBColor(15)** is white).

If the generated integer is not black, the code under the **Else** is executed. This code sets the **ForeColor** property of the label to black (**QBColor(0)** represents the black color).

 Select **Save Project** from the **File** menu to save your work, then select **Start** from the **Run** menu of Visual Basic to execute the Random program.


 Click the **Generate Integer Between 0 and 15** button a few times. Notice that when the generated number is not **0**, the digit inside the **lblOurLuckyInteger** is displayed as black. When the generated number is **0**, the digit inside the **lblOurLuckyInteger** is displayed as white.

 Experiment with the Random program, then terminate the program.

## How To Contact TegoSoft



You can contact TegoSoft Inc. by any one of the following methods:

 Use TegoSoft Internet Web site:

<http://www.tegosoft.com>

 Send TegoSoft an e-mail:

[tegosoft@msn.com](mailto:tegosoft@msn.com)

 Send TegoSoft a letter:

*TegoSoft Inc.*

*P.O.Box 389*

*Bellmore, NY 11710*


USA

## Technical Support



If you have a technical question, you can post the question to the TegoSoft Technical Support staff, and they will try to answer your question.

The **best** way to post a technical question is by sending TegoSoft an e-mail.

 The e-mail of TegoSoft is:  
tegosoft@msn.com

**When sending TegoSoft an e-mail with a technical question, please follow the following format:**

Date: \_\_\_\_\_  
Your name: \_\_\_\_\_  
Company (if applicable): \_\_\_\_\_  
Your phone number: \_\_\_\_\_  
Your e-mail: \_\_\_\_\_  
Country (if not USA): \_\_\_\_\_  
State (if inside USA): \_\_\_\_\_  
  
Operating System used: \_\_\_\_\_  
Programming language and version : \_\_\_\_\_

My technical question is:

## Copyright © and Notices

=====

**TegoSoft Self Study Tutorials & Software**

**Copyright ©1994, 1995, 1996 by TegoSoft Inc. ® All Rights Reserved**

Although precaution was taken in the preparation of this document, TegoSoft assumes no responsibility for errors or omissions. TegoSoft is not liable for damages resulting from the use of the information contained in this document. Use this document at your own risk. This document is copyright protected. This means that you should treat this document like any other copyright material. No part of this document should be copied in any way. You are not allowed to use this document or any part of this document for any purpose other than read it. You are not allowed to publish this document or any part of this document in any way. You are not allowed to sell this document or any part of this document, you are not allowed to incorporate this document or any part of this document in any book, magazine, Web Site, Electronic forums, disks, CDs, or any other media. The only thing that you are allowed to do with this document is read it for the sole purpose of studying the material that is presented in this document. If this document includes software, the software must be treated in the exact same way that this document is treated. You are not allowed to distribute the software in any way. The sole purpose of supplying the accompanying software is to enable you to experience with it. No part of this document should be modified. If accompanying software is included with this document, you are not allowed to modify the software.

Rev. 031796-1