

Case Select

In addition to if, and else-if statements there is another type of select statement called the Case Select statement.

Example. Consider a ten mark quiz. Let declare an integer variable called mark.

mark is the variable that will be evaluated throughout this statement.

```
Dim mark As Integer  
mark = Val(TextBox("Enter the mark(1-10)"))
```

```
Select Case mark  
  Case 10  
    MsgBox("Perfect")  
  Case 9  
    MsgBox("90%")  
  Case 8  
    MsgBox("80%")  
  Case 7  
    MsgBox("70%")  
  Case 6  
    MsgBox("60%")  
  Case 5  
    MsgBox("50%")  
  Case Else  
    MsgBox("Failed")  
End Select
```

Variable goes here after Select Case

If the variable matches up with the case value,ex.8 this is the code that will run.

Case Else handles situation where no Case values match the variable. This is optional.

All Select Case statements must end with End Select

Matching Range of Values

A case statement can be used to match a range of values as follows.

```
Dim age As Integer  
age = Val(InputBox("Enter Your Age"))
```

```
Select Case age  
    Case 5 To 12  
        MsgBox("Elementary School")  
    Case 13 To 18  
        MsgBox("High School")  
    Case 19 To 25  
        MsgBox("University")  
    Case 26 To 65  
        MsgBox("Career")  
    Case 65 To 100  
        MsgBox("Retirement")  
End Select
```

The keyword To is placed between the range of values.

Specific Multiple Values

If a specific values do not fall within a range, they can be separated by a comma.

```
Dim city As String
city = InputBox("Enter the City")

Select Case city
    Case "Toronto", "Montreal", "Vancouver"
        MsgBox("Canada")
    Case "New York", "Los Angeles", "Chicago"
        MsgBox("United States")
    Case "Beijing", "Shanghai", "Hong Kong"
        MsgBox("China")
    Case Else
        MsgBox("Unknown")
End Select
```

Case Is Statements

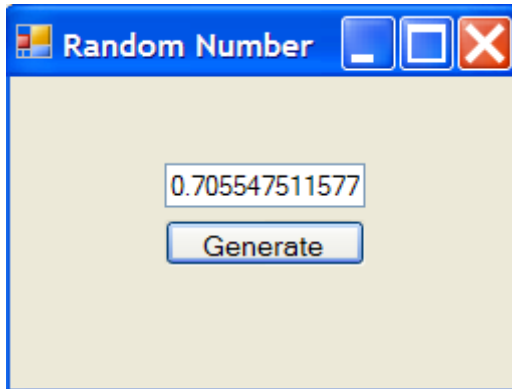
```
Dim mark As Integer
mark = Val(InputBox("Enter the mark(1-100)"))

Select Case mark
    Case Is = 100
        MsgBox("Perfect")
    Case Is > 90
        MsgBox("A+")
    Case Is > 80
        MsgBox("A")
    Case Is > 70
        MsgBox("B")
    Case Is > 60
        MsgBox("C")
    Case Is > 50
        MsgBox("D")
    Case Else
        MsgBox("Failed")
End Select
```

Case Is can be used to cover a range of values with relational operators

The Rnd() Function

The Rnd() function generates a random number using the computers clock.



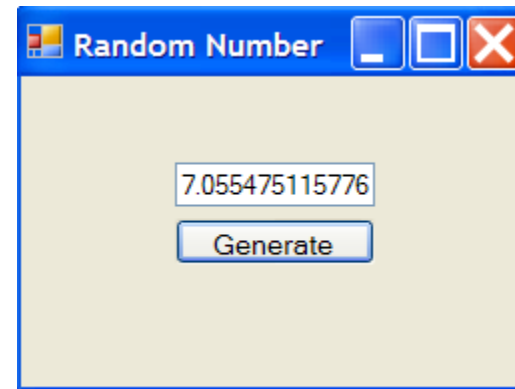
```
Private Sub Button1_Click(ByVal sender  
    Dim random As Double  
    random = Rnd()  
    txtRandom.Text = random  
End Sub
```

The Rnd() function produces numbers that are greater than or equal to zero and less than one.

Increasing the Range of Rnd()

In instances where you wanted to create a random number greater or equal to 0 and less than 10, you can simply multiply Rnd() by 10.

```
Private Sub Button1_Click(ByVal  
    Dim random As Double  
    random = Rnd() * 10  
    txtRandom.Text = random  
End Sub
```

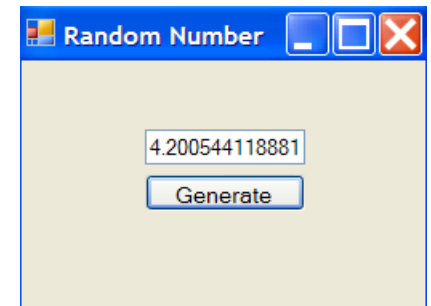
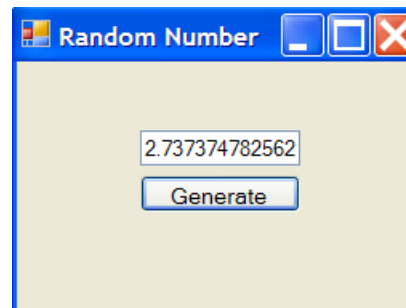
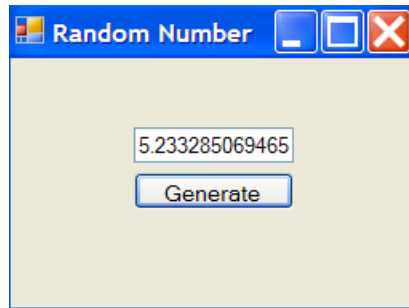


Random Number Within A Range

A random number within a specific range is generated using the following expression.

$$(\text{highnumber} - \text{lownumber} + 1) * \text{Rnd}() + \text{lownumber}$$

```
Private Sub Button1_Click(ByVal sender  
    Dim random As Double  
    random = (6 - 1 + 1) * Rnd() + 1  
    txtRandom.Text = random  
End Sub
```

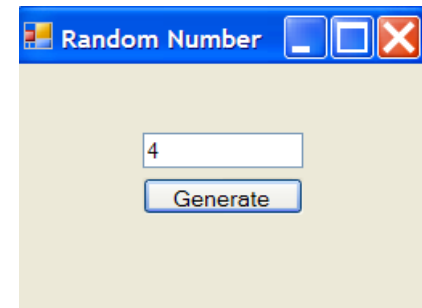
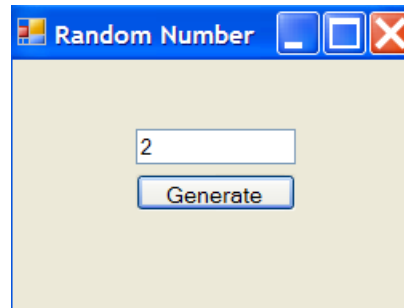
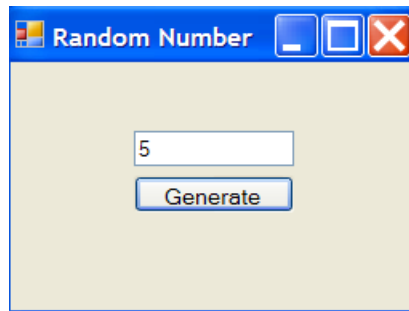


Generating Random Whole Numbers

By using the Int() function in combination with the Rnd() function, random whole numbers can be generated.

Example: A random number between 1 and 6(dice roll)

```
Private Sub Button1_Click(ByVal sender As System.  
    Dim random As Double  
    random = Int((6 - 1 + 1) * Rnd() + 1)  
    'This can be simplified to Int(6 * Rnd() + 1)  
    txtRandom.Text = random  
End Sub
```



A Random Numbers Game

Using the Rnd() function and Select Case Statements create a game that does the following.

“Roll” a random value between 1 and 10

1,4 and 7	add one point
2, 5 and 8	add 2 points
3, 9	lose a point
6	lose 3 points
10	Game Over

An integer variable called points should be declared at the top of your program (Global Variable)

The Game

```
Randomize()  
Dim roll As Double  
roll = Int(10 * Rnd() + 1)  
Select Case roll  
    Case 1, 4, 7  
        points = points + 1  
    Case 2, 5, 8  
        points = points + 2  
    Case 3, 9  
        points = points - 1  
    Case 6  
        points = points - 3  
    Case 10  
        points = 0  
End Select  
  
If (points <= 0) Then  
    MsgBox("Game Over " & points & " points!")  
    btnGenerate.Enabled = False  
Else  
    txtRandom.Text = "You have " & points & " points!"  
End If
```

Include this to avoid same rolls each time the program is run.