Mr. Giansante

Visual Basic

Loops

August 2016

Loops: For - Next, Do - While, Do - Until

For - Next Loops

Visual Basic, like most computer languages, provides several looping structures.

Normal counting

```
For i = 1 To 100
   (your code goes here)
Next i
```

Counting by an Increment

```
For i = 1 to 100 Step 5
   (your code goes here)
Next i
```

Counting in Reverse Order

```
For i = 200 to 100 Step -1
   (your code goes here)
Next i
```

"Nesting" Loops

You can nest For-Next loops by placing one For-Next loop within another. Give each loop a unique variable name as its counter.

```
For i = 1 To 10
   For j = 1 To 10
      For k = 1 To 10
      ...
      Next k
   Next j
Next i
```

Exiting the Loop Before it is Done

The `Exit For` statement can be used to provide an alternate way to exit a For-Next loop. It is usually used with and If-Then statement.

Do - While Loops

Repeats a block of statements while a condition is True.

```
Do While (condition)
   (place your code here)
Loop
```

You can use the **Exit Do** command as an alternate way to exit the loop

Example. Counting from 1 to 10.

```
x = 1
Do While (x <= 10)
   MessageBox.Show(x)
   x = x + 1
Loop
```

Do - Until Loops

Repeats a block of statements until a condition becomes True.

```
Do
   (place your code here)
Loop Until (condition)
```

You can use the **Exit Do** command as an alternate way to exit the loop

Example. Counting from 1 to 10.

```
x = 1
Do
    MessageBox.Show(x)
    x = x + 1
Loop Until (x > 10)
```

Loop Exercises - Part I

_____ _____

Name Date

Write the sequence of values generated by the following For-Next statements.

For i = 1 To 8 _____, _____, _____, _____, _____, _____, _____, _____

For i = 60 To 55 Step -1 _____, _____, _____, _____, _____, _____, _____, _____

For i = 2 To 10 Step 2 _____, _____, _____, _____, _____, _____, _____, _____

For i = 0 To 255 Step 64 _____, _____, _____, _____, _____, _____, _____, _____

For i = 255 To 0 Step -50 _____, _____, _____, _____, _____, _____, _____, _____

For i = -1 To 1 Step 0.5 _____, _____, _____, _____, _____, _____, _____, _____

Write a For-Next statement to generate each sequence of values.

1 to 50 For i = _____ To _____ Step _____

0 to 50 by 2s For i = _____ To _____ Step _____

0, 3, 6, 9 For i = _____ To _____ Step _____

10, 8, 6, 4, 2, 0 For i = _____ To _____ Step _____

1000, 2000, 3000 For i = _____ To _____ Step _____

1, 2, 3, 4 ... 365 For i = _____ To _____ Step _____

365, 364, 363, 362 ... 1 For i = _____ To _____ Step _____

0, 0.125, 0.25, 0.375, ... 1 For i = _____ To _____ Step _____

3, 2.5, 2, 1.5 ... 0 For i = _____ To _____ Step _____

Loop Exercises - Part II

Name _____ Date _____

Write For-Next loop statements to do the following. Use the integer variable "tot" in addition to the loop index "i".

example. Add up the integers from 1 to 100 inclusive.

```
tot = 0
For i = 1 to 100
   tot = tot + i
Next i
```

a. Add up the integers from 3 to 17 inclusive.

b. Add up the EVEN integers from 4 to 16 inclusive.

c. Calculate 20! (! means factorial and evaluates to 20 x 19 x 18 x 17 x … x 3 x 2 x 1).

Write the code to fill a ComboBox control (named ComboBox1) with the numbers 2, 4, 6, 8, 10 … 40 (ie. making a "Font Size" ComboBox). Use a **For…Next** loop!

Write the code that will display the numbers 1 to 25 along with the square of each number.
Use **Console.WriteLine()** to display the output. To view the results, select **View | Other Windows | Output**

example.

1	1
2	4
3	9
4	16
…	…
25	625

Repeat the above for numbers that are powers of 2, starting with 1 and ending with 2048.

Loop Exercises - Part III

Name _____ Date _____

Note: Use the Visual Basic program to assist you with the following tasks.

Write the code to generate the numbers from 1 to 50 ...

i. Using a **Do…Until** Loop

ii. Using a **Do…While** Loop

Write the code to generate the numbers from 0 to 100 by 4s …

i. Using a **Do…Until** Loop

ii. Using a **Do…While** Loop