Iteration (Loops)

Iteration or looping is used when the program needs to repeat the same operations a number of times. Usually the operations are using variables, where the values may be changing during the repetitions.

Visual Basic calls the two types of loops - counter controlled and sentinel controlled.

A **counter controlled loop** is controlled by some variable, often called the index variable, and it will use this variable to count the number of times to execute. In Basic it is a "FOR" loop and has the following construction:

```
FOR (integer variable) = (integer or integer variable) TO (integer or integer variable)
        [some action]
NEXT
```

In the FOR loop header (the first line), suppose that we have *FOR x =* . The variable x is the index variable, the next number is the **initial value** and after the TO, the next number is the **final value**. In the loop, the variable x is initialized to the initial value, incremented by 1 each time the loop is executed, and terminating the loop after the final value is reached.

.

If the initial value is 1, the loop will be executed for the number of times specified after the TO in the final value:
e.g.,

```
‘ will be repeated 5 times
FOR x = 1 TO 5
        Grade = text1.text
        Total = total + grade
NEXT

‘ will be repeated for numgrades times
FOR x = 1 TO numgrades
        Grade = text1.text
        Total = total + grade
NEXT
```

But the initial and final values can be any numbers. In the following loop, x is initialized to the value of a, incremented by 1 after every loop iteration, and terminated after the value of numgrades is reached.

```
FOR x = a to numgrades
        Grade = text1.text
        Total = total + grade
NEXT
```

The variable x can also be used as a value in the loop:

```
FOR x = a to b
    Total = Total + x
NEXT
```

Finally, the FOR loop can be made to increment (or decrement) the variable by different amounts using the optional STEP construct:

```
FOR (integer variable) = (initial value) TO (final value) STEP (integer)
    [some action]
NEXT (integer variable)
```

For example,

```
FOR x = a to b  STEP 2
    Total = Total + x
NEXT
```

In this loop, the variable x will start at a and be incremented by 2 each time around the loop.

A **sentinel controlled loop** is controlled by a condition. It will execute while a condition is true. There are several loop structures in Visual Basic, but we will focus on the WHILE structure, which executes the actions in the body of the loop as long as the condition is true.

```
WHILE (condition)
    [some actions]
END WHILE
```

The condition can use any of the following relational operators:

=	equal to
<>	not equal to
>	greater than
<	less than
<=	less than or equal to
>=	greater than or equal to)

e.g.:

```
total = 1
WHILE (total < 100)
    total = total * 2
END WHILE
```