Chapter 6 - Formatting and Arrays (pgs. 213 - 261)

*In this section you will learn how to format output, reusability, the scope of variables, arrays, the do-while loop, mouse events and using arrays to store information in a binary file.*

**Formatting**

* Function **Format$**-- takes parameters format type and string, ex. txtprice.text = Format$ (Price, "Currency"), format types include:
  + General Number, Currency, Fixed, Standard, Percent, Scientific, Yes/No, True/False, On/Off
* It is also possible to create formats manually, ex. txtAmount.Text = Format$ (Amount, "##0.####%"), see detail on pg. 216
* VB also includes predefined **time** and **date** formats including: general date, short date.....

**Modules and Code Reusability**

* Reusability is the idea that you should write code with the intent of separating the code that does not change from the code that does in order to reuse blocks of code. Functions and subroutines, are an example of code that can be reused in multiple projects and effectively modularized
* **Module** -- a basic unit of organization in VB projects, containing procedures and declarations in a single file and callable throughout the project
* Modules are either **form modules** (your userform) or a **code module**, you have already seen these

**The Scope of Variables**

* A variable's scope is determined by its: visibility and lifetime
* **Visibility** - the number of procedures, forms, and modules that can user or "see" the variable, ex. global versus local
* **Local variables** can only be used by the event procedure in which you have declared them, they have a limited lifetime as well (as soon as End Sub occurs, the memory that the variable occupied is reclaimed)
* **Static variables** are a type of local variable, a static variable can only be seen by a single event or procedure but when End Sub occurs, the value of a static variable is not lost, declare a static variable: ex. Static count As Integer
* **Global variables** - can be used by all procedures within a form

**The Do-Loop**

* The Do-Loop is indefinite, unlike the For-Next which has a set number of iterations
* While is a reserved word after which a condition is set during which the code block is executed
* There can be a Do-Loop without a while, the Do-While is a variation of the Do-Loop
* A generic Do-While-Loop is written as:

Do While x <= 100

x = x \* x

Loop

* The **Exit Do** statement allows you to exit the do while loop and not loop infinitely which would generate an error

**User Interface**

* The Inputbox and MsgBox function are **(I/O)** input/output functions using the keyboard
* Mouse events are I/O functions using the mouse