Applications Programming

Introduction to VBA
Part II

Named Constant

- A named constant is a named element whose value is read-only, and can not be changed by a programming statement while the program is running.
- Purpose:
  - Give a literal value (data) its meaning through the constant’s name
  - If the constant value needs to be changed, we only need to change it in one place where the constant is declared
- To declare a constant:
  Const constantName As dataType = Value
- Requirement
  - constant name must follow the variable naming rules
  - constant name should be meaningful

Math Expression

- Algebra expressions involving numerical values/variables and math operators (expressed with special symbols)
- Operators
  - negation (-)
  - exponentiation (^)
  - multiplication (*), floating-point division (/), integer division (\), Modulo (MOD)
  - addition (+), subtraction (-)

Math Expression Syntax

```
Math_Exp ::= literal_numerical_value

           | numerical_variable

           | numerical_constant

           | (Math_Exp)

           | - Math_Exp

           | Math_Exp operator Math_Exp
```

String Expression Syntax

```
String_Exp ::= literal_string_value
             | string_variable
             | Math_Exp
             | String_Exp & String_Exp
```

Assignment Statement

- syntax:
  variableName = expression

- An expression is a combination of keywords, operators, variables, literal values and constants that yields a string, number, or object. An expression can be used to perform a calculation, manipulate characters, or test data.

- Note that = is not called "equal to", but rather "becomes".

- The left hand side of the assignment statement must be an element that indicates a container, such as a variable name.

Input Statement

- From Excel sheets, there are two ways to identify the cells:
  - Relative to the active cell, using offset.
    Example:
    ```
    ' two rows down and 3 columns right to the active cell
    price = ActiveCell.Offset(2, 3)
    ' one row up and 6 columns left to the active cell
    data = ActiveCell.Offset(-1, -6)
    ```
  - directly identify the row and column number.
    Example:
    ```
    price = Cells(12, 35)    ' the cell in row 12 and column 35
    price = Cells(12, “D”)  ‘ the cell in row 12 and column D
    ```
- Using Input Box
  example:
  ```
  studentName = InputBox("Enter your name")
  ```

Output Statement

- To Excel sheets: cell reference

  - Relative to the active cell, using offset.
    Example:

    ```
    ' two rows down and 5 columns right to the active cell
    ActiveCell.Offset(2, 5) = price * units
    ```

  - directly identify the row and column number.
    Example:

    ```
    Cells(12, 35) = charge    ' the cell in row 12 and column 35
    Cells(12, “D”) = charge   ‘ the cell in row 12 and column D
    ```

- Using Message Box
  example:

  ```
  MsgBox "the total charge of this purchase is $" & charge
  ```