CSE 123
Introduction to Computing

Lecture 7
Array Variables in VBA
Qucik look at VBA Functions

SPRING 2012

Assist. Prof. A. Evren Tugtas

Marmara Üniversitesi

Variable versus Array Variable

- Variable holds one bit of data under a name
- Array variable holds more bits of information under a name

```
Dim Vegetables (3) as String
Vegetables(0)="Carrot"
Vegetables(1)="Cauliflower"
Vegetables(2)="Celery"
Vegetables(3)="Mushroom"
```

Why would you use array variables?

Array variables are used to;

- Dynamically access the data

  MsgBox Vegetables(2) ---- Celery

- Data provided by the user can be stored
- Arrays can be used in loops

Array Variables in VBA

- Array is a variable that can contain number of values that have the same data type.
- Array is treated as a single value in VBA
- You can refer to array itself to work with all the values it contains.
- You can also refer to individual numbers stored within the array by using their index numbers

MsgBox Vegetables(2) ---- Celery

Array Variables in VBA

- An array is bounded by a lower and upper bound
- By default the lower bound is ZERO, therefore, the first item in an array is indexed as ZERO

```
Dim Vegetables (1) as String
Vegetables(0)="Carrot"
Vegetables(1)="Cauliflower"
```

- This could be confusing because the index number is always one lower than the items position in an array.

Array Variables in VBA

- VBA lets you change the default lower bound.
- Using **Option Base 1** statement at the begining of your code makes the default index number of the first item in an array 1.
- **Option Base 1** statement makes the index number for each item in an array the same as the item's position in that array.
- Other programming languages do not have this option, by default their arrays are zero based.

Declaring an Array

- Array is a variable, therefore, regular keywords are used to declare it
  - Dim
  - Private
  - Public
  - Static
- Array variables are indicated by a pair of parenthesis

Dim A as Single **Dim A( ) as Single**

Declaring an Array

- Number of items in an array are declared by an array subscript.
- Following statement declares that the array named A assigns the Single data type and contains **6** items. A is a one dimensional array.

Dim A(5) as Single

Be carefull, it is a ZERO BASED array

Declaring an Array

Dim A(5) as Single

Element Number	Name	Content
0	A(0)	12
1	A(1)	54
2	A(2)	67
3	A(3)	81
4	A(4)	3
5	A(5)	98

Declaring an Array

- If you want numbering to start at 1,

Option Base 1

Dim A(6) as Single

Element Number	Name	Content
1	A(1)	12
2	A(2)	54
3	A(3)	67
4	A(4)	81
5	A(5)	3
6	A(6)	98

Storing Values in an Array

```
Option Base 1
Dim Cities (6) as String
Cities (1)="Istanbul"
Cities(2)="Ankara"
Cities(3)="Bursa"
```

You have created a String Array but only three variables have been assigned

Storing Values in an Array

Element Number	Name	Content
1	Cities(1)	Istanbul
2	Cities(2)	Izmir
3	Cities(3)	Bursa
4	Cities(4)	-
5	Cities(5)	-
6	Cities(6)	-

Multidimensional Arrays

- VBA supports arrays up to 60 dimensions

Option Base 1

Dim Myarray(3,3)

Resulting two dimensional array would be;

Column1	Column2	Column3
1,1	2,1	3,1
1,2	2,2	3,2
1,3	2,3	3,3

Multidimensional Arrays

- In multidimensional arrays the information in any series does not have to be related to eachother.
  - You can assign 10 folder names to first dimension as string
  - 10 filenames to second dimension as string
  - Names pf 10 cities to the third dimension
- Later on you can access the any information you like by specifying the location.
- Excel workbook of worksheets, rows and columns is a three-dimensional array.

Declaring a Dynamic Array

- Arrays can be declared as ***fixed-size arrays*** or ***dynamic arrays***.
- Dim A(6) → ***Fixed-size*** array
- ***Dynamic arrays*** are used when you are storing changing number of arrays
- For example windows in Microsoft is a ***dynamic array*** because the programmer does not know the number of windows that will be opened by the user

Declaring a Dynamic Array

- You should not specify the item number when you are declaring a dynamic array.
- **Dim Power( )** → dynamic array, size is not declared

Redimensioning an Array
ReDim Statement

- You can change the size of an Array by using ***ReDim*** Statement

ReDim Countries(5)

- When you redimension the arrays using ReDim statement, you lose the values currently in the array.

Redimensioning an Array
ReDim Statement

- If you want the preserve the contents of an array while redimensioning it, you need to use the ***ReDim Preserve*** statement

ReDim Preserve Countries(5)

- You would still lose the information stored in any subscripts that is not included in the array.
- ReDim Preserve only works for the last dimension array, you cannot preserve the data in other dimensions

Returning Information from an Array

- You need to use the index number to specify the position of the data you want to return

```
Option Base 1
MsgBox Cities(3)
```

```
Option Base 1
MsgBox Myarray (2,3)
```

Returns third item of the second dimension of an array

Determining array vaeiables

- You may want to check whether a variable is array or scalar variable.
- IsArray function with the variable name is used.

```
If IsArray(A)=True Then
Msg="A" & " is an array"
Else
Msg="A" & " is not an array"
End If
```

Finding boundaries of an Array

Lbound(array, [, dimension])

Ubound(array, [, dimension])

Array – Name of your array

Dimension – optional variant specifying the dimension whose bound you want to return

MsgBox Ubound(Myarray, 2) – returns upper bound of the second dimension of array ***Myarray***

Sorting an Array

- You may need to sort an array especially if you ara loading an information into the array from an external source.
- Sorting will be revisited in the loops lecture

Erasing an Array

- Contents of an array can be erased by using the ***Erase*** statement with the name of the array.
- This statement frees the memory in the arrays

- Erase Myarray

Functions

Functions

- Function is a procedure
- Function always returns a value
- Subroutine does not return a value
- You feed information into a built-in prewritten function that's built into VBA
- The built in function processes the info you feed in and returns a value

Using Functions

- To use a function→ you need to call it from a subprocedure or from another function.
  - Use the Call statement (rarely used)
  - Use the name of the function itself

Passing Arguments to a Function

- Supplying the argument values, without their names, in the order in which the function expects them
- Supplying the arguments, with their names, in the order which the function expects them
- Supplying the arguments, with their names, in any order you choose.

Function with one argument

```
Function viscosity(T As Double) As Double
viscosity = 0.001 / (0.021482 * (T - 8.435 + Sqr(T ^ 2 - 16.87 * T +
    8149.5492)) - 1.2)
End Function
```

```
Function density(T As Double) As Double
 density = 999.84 + 0.066436 * T - 0.0087983 * T ^ 2 + 7.8934 * 10 ^ (-5)
    * T ^ 3 - 4.9115 * 10 ^ (-7) * T ^ 4
End Function
```

Function with three arguments

```
Option Explicit
Public Function gasvol(P As Single, T As Single, n As
Single) As Single
Dim R As Single
R = 0.0082 ' Ideal gas constant L atm /K mol
gasvol = n * R * T / P
End Function
```

```
Sub idealgas()
Dim Ans As Single
Ans = gasvol(Range("B3"), Range("B4"), Range("B5"))
MsgBox ("Volume of the gas is " & Ans  & "L")
```