**ITEC397 Chapter 02 LABWORK**

1. **Explicitly Defined Variables**
* Create a SUB program called Test\_explicit
* Define an Integer variable called MyInteger
* Define an String variable called MyString
* Define an Currency variable called MyCurrency
* Define an Long variable called MyLong
* Define an Double variable called MyDouble
* Use MsgBox and Vartype functions to display the datatypes of your variables

**Solution:**

Sub Test\_explicit()

 DIM MyInteger as Integer

 DIM MyString as String

 DIM MyCurrency as Currency

 DIM MyLong as Long

 DIM MyDouble as Double

 MsgBox VarType(MyInteger)

 MsgBox VarType(MyString)

 MsgBox VarType(MyCurrency)

 MsgBox VarType(MyLong)

 MsgBox VarType(MyDouble)

End Sub

1. **Implicitly Defined Variables**
* Create a SUB program called Test\_implicit
* Assign the value 100 to the variable vTest
* Display the data type of the variable vTest
* Assign the value “Hello World” to the variable vTest
* Display the data type of the variable vTest
* Assign the value 3.14 to the variable vTest
* Display the data type of the variable vTest
* Assign the value true to the variable vTest
* Display the data type of the variable vTest

***Solution:***

Sub Test\_implicit()

 vtest = 100

 MsgBox VarType(vtest)

 vtest = “Hello World”

 MsgBox VarType(vtest)

 vtest = 3.14

 MsgBox VarType(vtest)

 vtest = True

 MsgBox VarType(v\_test)

End Sub

1. **Fixed and Variable Length Strings**
* Create a SUB program called Test\_String
* Define a String variable sTest
* Assign the value “Hello World”to the variable sTest
* Display the data type of the variable sTest2
* Define a String variable sTest2 with 5 characters long
* Assign the value “Hello World”to the variable sTest2
* Display the data type of the variable sTest2

***Solution:***

Sub Test\_String()

 Dim sTest as String

 sTest = “Hello World”

 MsgBox sTest

 Dim sTest2 as String \* 5

 sTest2 = “Hello World”

 MsgBox sTest2

End Sub

1. **Definei initialize and print an Array**
* Create a SUB program called Test\_Array
* Define a one dimensional integer array with size 5 and name is arTest
* Display LBound, and UBound values of the array arTest
* Assign the values 0 to 6 to the elements 0, to 6 of the array arTest
* Display the contenent of the elements from 0 to 6
* See the error, you can’t use any element having larger index than Ubound
* Modify your program and re run it
* NOTE THAT, in VBA element size is defined always one larger then the given number.

***Solution***

Sub Test\_Arrays()

 Dim aTest(5) As Integer

 MsgBox LBound(aTest)

 MsgBox UBound(aTest)

 aTest(0) = 0

 aTest(1) = 1

 aTest(2) = 2

 aTest(3) = 3

 aTest(4) = 4

 aTest(5) = 5

 'aTest(6) = 6

 MsgBox aTest(0)

 MsgBox aTest(1)

 MsgBox aTest(2)

 MsgBox aTest(3)

 MsgBox aTest(4)

 MsgBox aTest(5)

 'MsgBox aTest(6)

End Sub

1. **Modify the previous example as the elements of the array definitely have to be from 0 to 4**
* Change the array definition as it will be Dim aTest(0 to 5) As Integer

***Solution***

Sub Test\_Arrays()

 Dim aTest(0 To 4) As Integer

 MsgBox LBound(aTest)

 MsgBox UBound(aTest)

 aTest(0) = 0

 aTest(1) = 1

 aTest(2) = 2

 aTest(3) = 3

 aTest(4) = 4

 'aTest(5) = 5

 'aTest(6) = 6

 MsgBox aTest(0)

 MsgBox aTest(1)

 MsgBox aTest(2)

 MsgBox aTest(3)

 MsgBox aTest(4)

 'MsgBox aTest(5)

 'MsgBox aTest(6)

End Sub

1. **Dynamic use of the arrays**
* Modify the previous example as its size have to be extent from 5 to 7
* Add values 5 and 6 to the last elements
* Display the content of the array

**Note that**, if you use redim without **Preserve** key word on the size change, all existing content of the array will be cleaned.

* Fixed your program to not loose the existing content of the array when the size of it will be changed.

***Solution-1:***

Sub Test\_Arrays()

 Dim aTest() As Integer

 ReDim aTest(5) As Integer

 MsgBox "LBound:" & LBound(aTest)

 MsgBox "UBound:" & UBound(aTest)

 aTest(0) = 0

 aTest(1) = 1

 aTest(2) = 2

 aTest(3) = 3

 aTest(4) = 4

 aTest(5) = 5

 ReDim aTest(7) As Integer

 MsgBox "LBound:" & LBound(aTest)

 MsgBox "UBound:" & UBound(aTest)

 aTest(6) = 6

 aTest(7) = 7

 MsgBox aTest(0)

 MsgBox aTest(1)

 MsgBox aTest(2)

 MsgBox aTest(3)

 MsgBox aTest(4)

 MsgBox aTest(5)

 MsgBox aTest(6)

 MsgBox aTest(7)

End Sub

***Solution-2:***

Sub Test\_Arrays()

 Dim aTest() As Integer

 ReDim aTest(5) As Integer

 MsgBox "LBound:" & LBound(aTest)

 MsgBox "UBound:" & UBound(aTest)

 aTest(0) = 0

 aTest(1) = 1

 aTest(2) = 2

 aTest(3) = 3

 aTest(4) = 4

 aTest(5) = 5

 ReDim Preserve aTest(7) As Integer

 MsgBox "LBound:" & LBound(aTest)

 MsgBox "UBound:" & UBound(aTest)

 aTest(6) = 6

 aTest(7) = 7

 MsgBox aTest(0)

 MsgBox aTest(1)

 MsgBox aTest(2)

 MsgBox aTest(3)

 MsgBox aTest(4)

 MsgBox aTest(5)

 MsgBox aTest(6)

 MsgBox aTest(7)

End Sub

1. **User Defined Data Types**
* Create a user defined data type as Student that will have two properties and add this definition to “module level”:
	+ Stdno String\*8
	+ Stdname String\*20
* Create a SUB program called Test\_UDV
* Create an array of Students with two elements
* Set the elements details as
	+ “20151836”, “Ayse Salih”
	+ “20162772”, ”Hasan Ali”
* Print the array elements

***Solution***

**In to the module-1**

 Type Student

 stdno As String

 stdname As String

 End Type

**In to the ThisWorkBook**

Sub Test\_UDV()

 Dim arstudents(0 To 1) As Student

 arstudents(0).stdno = "20151836"

 arstudents(0).stdname = "Ayse Salih"

 arstudents(1).stdno = "20162772"

 arstudents(1).stdname = "Hasan Ali"

 For n = 0 To 1

 MsgBox arstudents(n).stdno & " " & arstudents(n).stdname

 Next n

End Sub