**ITEC397 Chapter 07 LABWORK**

1. **An example for a Compiler Error**

* Open a macro-enabled Excel sheet, go to developer Tab, open the Visual Basic editor, if it is not created, create a module (module1)
* Create a sub routine named example\_07\_01
* Write the code (infinite loop) and run it to see the “Compiler Error” you have done.

*Sub example\_07\_01*

*X=2*

*Do Until x=9*

*X = X +1*

*MsgBox “X=” & X*

*End Sub*

* Here, we used Do Until statement but missed to end it with Loop statement.
* Modify the code as it is given below and re-run it

*Sub example\_07\_01*

*X=2*

*Do Until x=9*

*X = X +1*

*Loop*

*MsgBox “X=” & X*

*End Sub*

1. **An example for a Runtime Error**

* Create a sub routine named example\_07\_02
* Write the code (infinite loop) and run it to see the “Runtime error” you have done.

*Sub example\_07\_02*

*X=2*

*MsgBox X / (X - 2)*

*End Sub*

* Modify the code as it is given below and re-run it

*Sub example\_07\_02*

*X=2*

*MsgBox X / (X + 2)*

*End Sub*

1. **An example for a Logical Error and debugging of it**

* Create a sub routine named example\_07\_03
* Write the code (infinite loop) and run it to see the system will never finish it is a Logic error.

*Sub example\_07\_03*

*X=2*

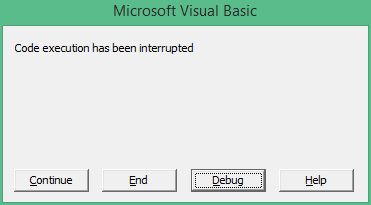
*Do Until x=1*

*X = X +1*

*Loop*

*End Sub*

* Break the program with pressing Ctrl-Break buttons. When the Code execution has been interrupted window appear, click the Debug button (Instant watch Mode).



* With pressing F8, statement by statement you can run your code, and if you bring the cursor on any variable, you can learn the current value of it.
* Drag the yellow arrow on to the statement as x=2, check the current value of the x, then execute that statement (F8) and re-check the current value of the X.
* You can stop the execution completely by clicking Reset button, or restart your code with Run (F5) button.

1. **An example for using Stop Statement for debugging**

* Copy-Past the content of the example\_07\_03 as a new subroutine example\_07\_04
* Modify it as it will be as the given code below.

Sub example\_07\_04()

x = 2

Do Until x = 1

x = x + 1

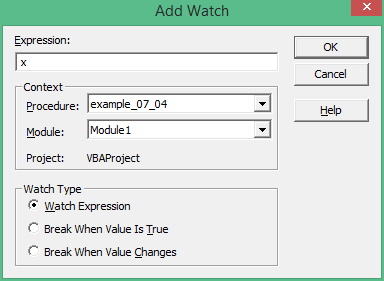
Stop

Loop

End Sub

* Run your code. With clicking F8 execute the statements step by step.

1. **Using the Watch Window in debugging**
   * With Calling Debug 🡪Add watch bring the following configuration window, and enter the variable name x into the Expression field and click OK.



* + On the bottom most part of the screen Watch panel have to be appear.
  + Run the code of example\_07\_04 and follow the variable X.

1. **Example for Single Stepping**
   * Open the exercise example\_07\_03
   * Click the statement “Loop”, and then press CTRL+F8, that will run all statements above until the statement “Loop” (where the cursor was on)
2. **Example of using Call Stack Dialog in debugging**

* Create a sub routines example\_07\_05, f\_assign(), and f\_add() with the code given below

*Sub t\_assign()*

*x = 2*

*End Sub*

*Sub t\_add()*

*x = x + 1*

*End Sub*

*Sub example\_07\_05()*

*t\_assign*

*Do Until x = 1*

*t\_add*

*Loop*

*End Sub*

* + Put a break on the statement Loop, run the code in stepwise until it will come on the subroutine t\_add, then press Ctrl-L to open the Call-Stack window to see, t\_add is the current subroutine which is called by example\_07\_05 (It is still uncompleted task).