LA IMMACULADA CONCEPCION SCHOOL
SENIOR HIGHSCHOOL
GRADE 11 – ICT: COMPUTER PROGRAMMING

Topic: LESSON 15: ERRORS HANDLING

Introduction

Error handling is an essential procedure in Visual Basic 2008 programming because it can help make the program error-free. An error-free program can run smoothly and efficiently, otherwise all sorts of problems occur such as program crash or system hang.

Errors often occur due to incorrect input from the user. For example, the user might make the mistake of attempting to enter a text (string) to a box that is designed to handle only numeric values such as the weight of a person, the computer will not be able to perform arithmetic calculation for text therefore will create an error. These errors are known as synchronous errors.

Therefore, a good programmer should be more alert to the parts of program that could trigger errors and should write errors handling code to help the user in managing the errors. Writing errors handling code should be considered a good practice for Visual Basic programmers, so do try to finish a program fast by omitting the errors handling code. However, there should not be too many errors handling code in the program as it creates problems for the programmer to maintain and troubleshoot the program later.

Visual Basic 2008 has improved a lot in built-in errors handling compared to Visual Basic 6. For example, when the user attempts to divide a number by zero, Vb2008 will not return an error message but gives the **'infinity'** as the answer (although this is mathematically incorrect, because it should be undefined)

Using On Error GoTo Syantax

Visual Basic 2008 still supports the VB6 errors handling syntax, that is the On **Error GoTo** ***program_label*** structure. Although it has a more advanced error handling method, we shall deal with that later. We shall now learn how to write errors handling code in VB2008. The syntax for errors handling is

```
On Error GoTo   program_label
```

where program_label is the section of code that is designed by the programmer to handle the error committed by the user. Once an error is detected, the program will jump to the ***program_label*** section for error handling.

Example: Division by Zero

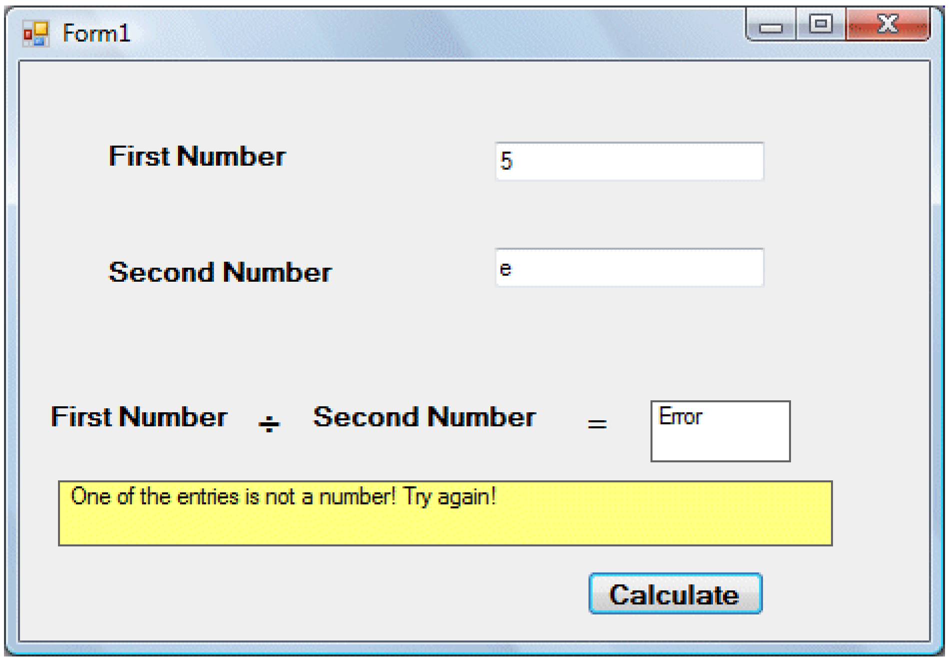
In this example, we will deal with the error of entering non-numeric data into the textboxes that supposed to hold numeric values. The program_label here is error_hanldler. when the user enters a non-numeric values into the textboxes, the error message will display the the text"One of the entries is not a number! Try again!". If no error occur, it will display the correct answer. Try it out yourself.

The Code

```
Public Class Form1
      Private Sub CmdCalculate_Click(ByVal As System.Object,ByVal e As
      System.EventArgs) Handles CmdCalculate.Click
            Lbl_ErrorMsg.Visible = False
            Dim firstNum, secondNum As Double
            On Error GoTo error_handler
            firstNum =Txt_FirstNumber.Text
            secondNum = Txt_SecondNumber.Text
            Lbl_Answer.Text = firstNum / secondNum
            Exit Sub 'To prevent error handling even the inputs are valid
            error_handler:
```

```
            Lbl_Answer.Text = "Error"
            Lbl_ErrorMsg.Visible = True
            Lbl_ErrorMsg.Text = "One of the entries is not a number! Try again!"
      End Sub
End Class
```

The Output



One of the entries is not a number! Try again!

Errors Handling Using Try.....Catch....End Try

VB2008 has adopted a new approach in handling errors, or rather exceptions handling. It is supposed to be more efficient than the old On Error Goto method, where it can handles various types of errors within the **Try...Catch...End Try** structure.

The structure looks like this

```
Try
      statements
Catch exception_variable as Exception
      statements to deal with exceptions
End Try
```

To Do:

Create your own sample program using try catch and try. Monitor and try to analyze what errors can be found within execution.