Object Oriented Programming with Visual Basic .Net

By:

Dr. Hossein Hakimzadeh
Computer Science and Informatics
IU South Bend

(c) Copyright 2007 to 2015 H. Hakimzadeh
What is Visual Basic .Net

Visual Basic .Net is the latest reincarnation of Basic language.
What is Visual Basic .Net

Programmers can use Visual basic to create “Windows” as well as “Console” applications.
What is Visual Basic .Net

VB .Net programmers can rapidly create GUI enabled applications that are event driven and object oriented.
What is Structured Procedural Programming

- In traditional or procedural applications, application execution starts with the first line of code, and follows a predefined path through the application, calling procedures as needed.

- Sequential flow of control.

- Code and data are distinct from each other.
What is Object Oriented Programming?

- Object-oriented programming (OOP) is a programming paradigm that uses "objects" and their interactions to design applications.

- It is based on several techniques, including encapsulation, polymorphism, and inheritance.
What is Object Oriented Programming?

- Object-oriented history goes back to the 1960s, however it did not popular until the late 1980’s or early 1990s.

- The primary need for OOP’s arose as the result of software becoming increasingly complex and unmanageable.

- Object Oriented Programming languages were designed to improve software re-use, quality and maintenance.

- Object-oriented programming can be viewed as a collection of cooperating objects, as opposed to a traditional view in which a program may be seen as a group of tasks.

- In OOP, each object is capable of receiving messages, processing data, and sending messages to other objects.
What is Object Oriented Programming?

- OOP’s integrate Code and Data together to form an ‘Object’.

- OO Programming Languages provide safer and more robust programming facilities, and they encourage code reuse through the use of inheritance.
Main Features of Object Oriented Programming?

- **Encapsulation**
  - (Protect data from unauthorized use)
  - (prevent inadvertent programming errors)

- **Inheritance**
  - (inherit and extend the capabilities of preexisting code)
  - (allow for reusable code)

- **Polymorphism**
  - (Ability to overload and override methods)
  - (makes code more readable and intuitive)
What is Event Driven Programming?

- In an event-driven application, execution does not follow a predetermined path. Instead, it runs different code sections in response to events.

- Events can be triggered by the user's actions, by messages from the system or other applications. The sequence of events determines the sequence in which the code runs. Therefore, the path through the application's code differs each time the program runs.
What is Event Driven Programming?

- An essential part of event-driven programming is to write code that responds to the possible events that may occur in an application.

- The figure shows some actions that generate events to which you can respond by writing code. These events can occur in any order.
What is Visual Basic Project?

- When using Visual Studio, a project is a collection of files and directories that get created as when you create a Windows or Console application.
Developing a VB Project

- **Planning:**
  - Design the GUI
  - Plan the properties
  - Plan the Code (pseudo-code to solve the problem)
  - Desk check your pseudo-code.

- **Coding:**
  - Convert the “GUI Design” to Forms and Controls.
  - Set the properties of the Forms and Controls.
  - Convert the pseudo-code to Visual Basic code.
  - Compile, Test and Debug the code.
Let’s look the steps for developing a simple VB Project
Step 1: Create a window (or FORM)

- A form or window for developing an application
Step 2: Using the **Tool Box**, select elements called **Controls** and place them on the window.

Controls:
- Button
- Text box
- Check Box
- Label
- Radio Button
- etc.
Let’s look the steps for developing a simple VB Project.

3) Set the properties of each ‘control’.
Step 4: **Program** the ‘controls’ to react to events.

- By double clicking a control, (e.g. the “Calculate” button) you can write the code behind that button.
Let’s Implement a Windows Application....
Problem 1:

Given the radius of a circle, compute and display the Area and the Circumference of the circle.
Problem 1: Given the radius of a circle, compute and display the Area and the Circumference.

Analysis:

- Determine the Input and Output of the program:
  - Radius of a circle

- Determine the formulas, fact, etc. needed:
  - Area = PI * (Radius)^2
  - Circumference = 2 * PI * Radius
  - PI = 3.14159
Problem 1: Given the radius of a circle, compute and display the Area and the Circumference.

Design:

- Design the **GUI** (windows and controls)
- Set the **properties** of the windows and controls.
- Determine the **events** that could occur, and write code to handle those events:
Problem 1: Given the radius of a circle, compute and display the Area and the Circumference.

**Design:**
- Determine the *events* that could occur, and write code to handle those events:

  - When the user presses the **EXIT** button: Close() the application
  - When the user presses the **CALCULATE** button: Get the radius from the radius textbox, compute the Area
    - \( \text{Area} = \pi \times \text{Radius}^2 \)
    - computer the circumference
      - \( \text{Circumference} = 2 \times \pi \times \text{Radius} \)
  - Display the area and circumference in appropriate text boxes.
Implementation:

- Similarly Double Click the **CALCULATE** Button. You will be placed in the CODE window, and the following event handler code (event procedure) will be automatically created for you.

```vbnet
Private Sub BtnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnExit.Click
    Me.Close()
End Sub
```

- Type the following code within the event procedure block:

```vbnet
Me.Close()
```
Implementation:

Double Click the EXIT Button. You will be placed in the CODE window, and the following event handler code (event procedure) will be automatically created for you.

```vbnet
Private Sub BtnCalculate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnCalculate.Click
End Sub
```

Type the following code within the event procedure block:

```vbnet
Dim Radius, Area, Circumference As Double
Const PI As Double = 3.14159
Radius = CDbl(TxtRadius.Text) 'Convert the string to double
Area = PI * Radius * Radius
Circumference = 2 * PI * Radius
TxtArea.Text = CStr(Area) 'Convert to string and put in the textbox
TxtCircumference.Text = CStr(Circumference)
```
Final Results:

“Windows” Application
Let’s implement the same problem as a “Console” application.
Problem 1: Given the radius of a circle, compute and display the Area and the Circumference.

Implementation:

Module Module1

Sub Main()

    Dim radius, area, circumference As Double
    Const PI = 3.14159

    Console.Write("Please enter the radius of the circle? ")
    radius = Console.ReadLine()
    area = PI * radius * radius
    circumference = 2 * PI * radius
    Console.WriteLine("Area = " & area)
    Console.WriteLine("Circumference = " & circumference)

    Console.ReadLine() 'just to pause the program.

End Sub
End Module
Final Results:

“Console” Application

Please enter the radius of the circle? 10
Area = 314.159
Circumference = 62.8318
Different Errors Encountered by Programmers

- Syntax Errors:
  - When VB’s rules for punctuation, format, or spelling is violated.
  - Most Syntax errors are detected by the editor in the IDE.

- Runtime Errors
  - If your program halts or crashes during execution.
  - Example:
    - Divide by zero.
    - Finding the square root of a negative number.
    - Trying to read from a non-existing file.
  - Runtime errors are also known as ‘EXCEPTIONS’

- Logic Errors
  - The program runs but produces incorrect results.