

# Chapter 1

## Introduction to Visual Basic 2010

---

- ❖ A brief description of Visual Basic 2010
  - ❖ Getting to know the Visual Basic 2010 Integrated Development Environment
- 

### 1.1 A Brief Description of Visual Basic 2010

Visual Basic 2010 is the latest version of Visual Basic launched by Microsoft in 2010. It is almost similar to Visual Basic 2008 but it has added many new features. Visual Basic has gone through many phases of development since the days of BASIC that was built for DOS. BASIC stands for **B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. The program code in Visual Basic resembles the English language. Different software companies had produced many different versions of BASIC for DOS, such as Microsoft QBASIC, QUICKBASIC, GWBASIC, and IBM BASICA and more. Then, Microsoft launched the first graphical BASIC, Visual Basic Version 1 in 1991. It is GUI based and especially developed for MS window. Since then Microsoft slowly phased out the DOS versions of BASIC and completely replaced them by Visual Basic.

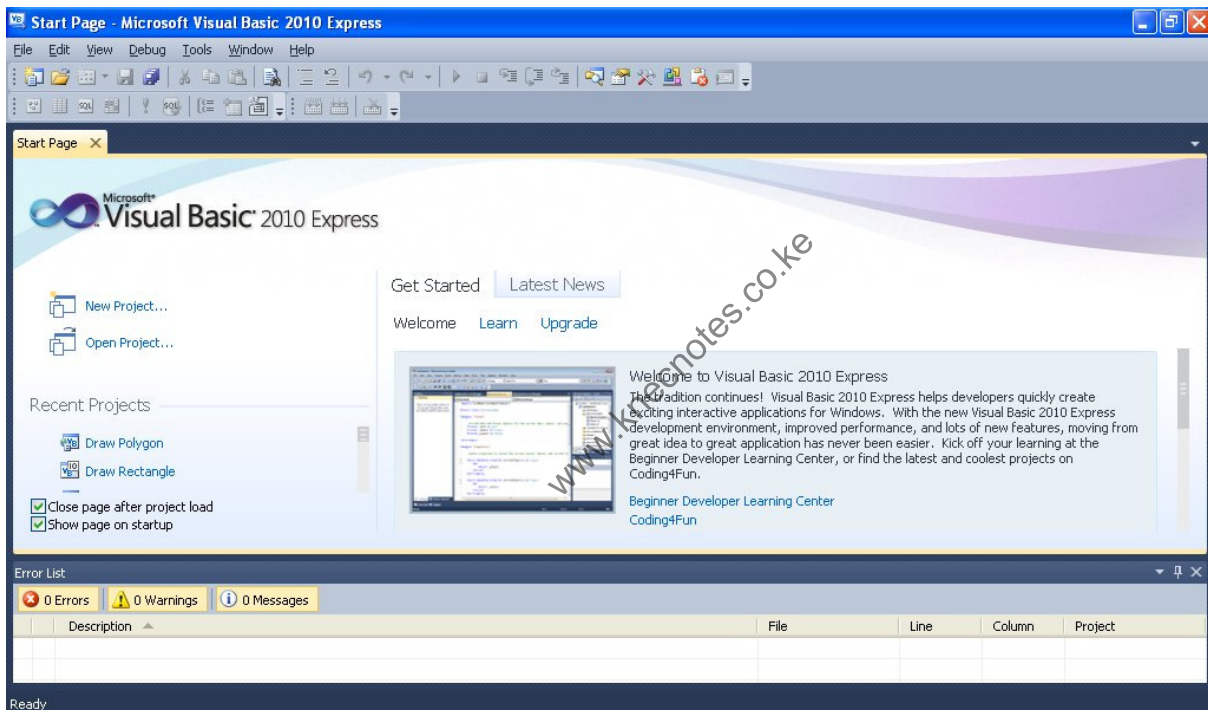
Visual Basic was initially a functional or procedural programming language until the popular Visual Basic 6. Then, Microsoft transformed Visual Basic into a more powerful object oriented programming language by launching Visual Basic.Net, Visual Basic 2005, Visual Basic 2008 and the latest Visual Basic 2010. Visual Basic 2010 is a full-fledged Object-Oriented Programming (OOP) Language; it has caught up with other OOP languages such as C++, Java, C# and others. However, you do not have to know OOP to learn VB2010. In fact, if you are familiar with Visual Basic 6, you can learn VB2010 effortlessly because the syntax and interface are almost similar. Visual Basic 2010 Express Edition is available for free download from the Microsoft site as shown below:

<http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express>

## 1.2 Navigating the Visual Basic 2010 Integrated Development Environment

### 1.2.1 The Start Page

When you launch Visual Basic 2010 Express, you can see the start page of the Integrated Development Environment, as shown in Figure 1.1.



**Figure 1.1: The VB2010 IDE Start Page**

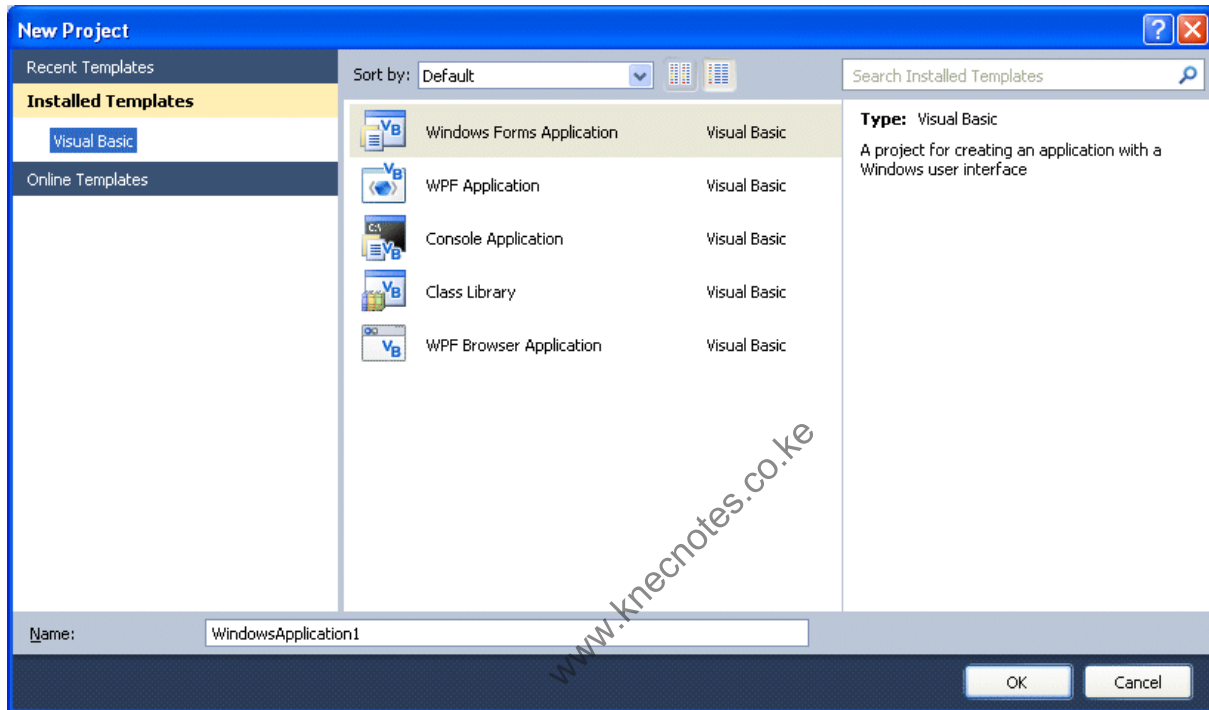
The IDE consists of a few panes, namely:

- The Recent Projects Pane- it shows the list of projects that you have created recently.
- The Get Started Pane- It provides some helpful tips so that you can quickly develop your new application.
- The Latest News pane- It provides latest online news about Visual Basic 2010 Express. It will announce new releases and updates.

Besides that, it also shows two icons, New Project and Open Project.

## 1.2.2 The New Project Dialog

When you click on the **New Project** icon, the Visual Basic 2010 New Project dialog will appear, as shown in Figure 1.2



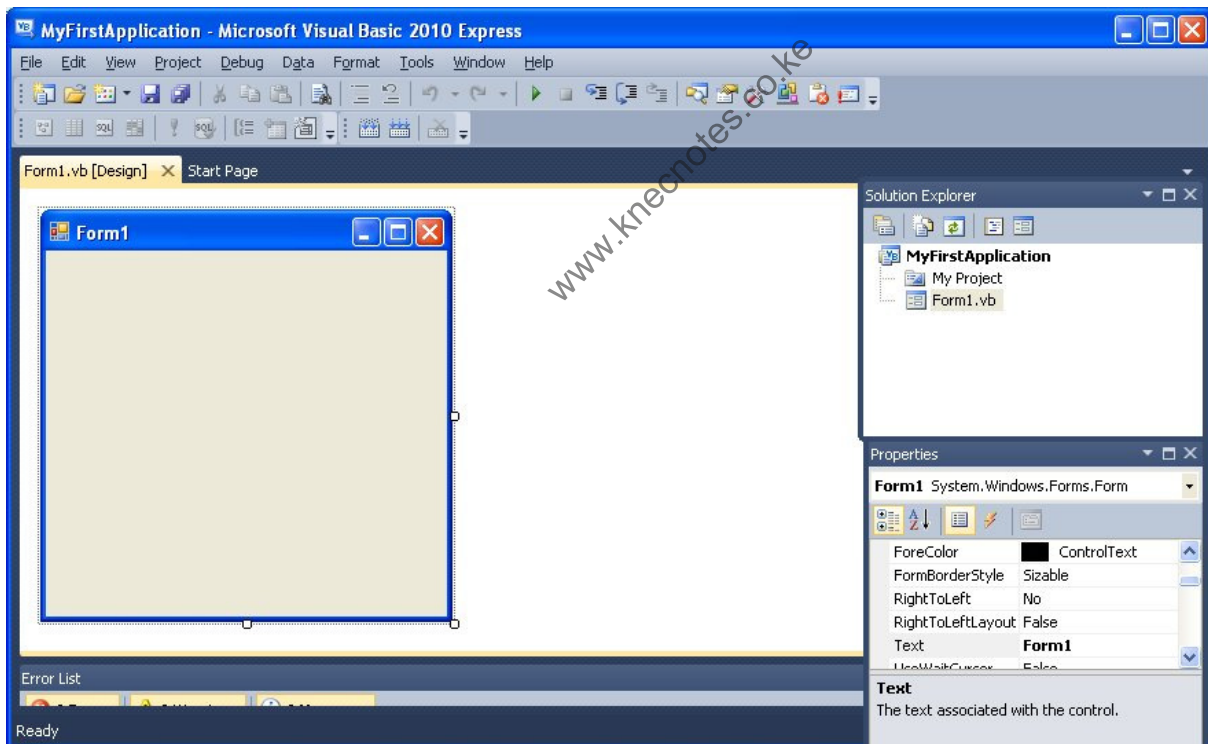
**Figure 1.2: VB2010 New Project Dialog Box**

The dialog box offers you five types of projects that you can create. They are Windows Form Application, WPF Application, Console Application, Class Library and WPF Browser Application. As we are going to create a standard Windows application, we will select Windows Forms Application. At the bottom of this dialog box, you can change the default project name **WindowsApplication1** to some other name you like, for example, **MyFirstApplication**. After you have renamed the project, click OK to go into the Designer interface.

## 1.2.3 The Designer Interface

The VB2010 IDE Designer interface is shown in Figure 1.3. The Designer consists of the **Menu bar**, the **Toolbars**, an empty **Form**, the **Solution Explorer** and the **Properties Window**.

The VB2010 Designer environment that appears on your PC or laptop might not be the same here, depending how you customize it. You can customize your interface by dragging the windows and dock them or let them float. You can also hide them. To dock a window, you drag its title bar and drag it to the side, top or bottom of the workspace or another window. In Figure 1.3, we have dragged the Solution Explorer and the Properties Window to the side and docked them. You can also resize the docked window by dragging the side of the window. To free up and float the docked window, you just drag its title bar and move it away from the edge of the workspace. If you do not see a particular window such as the properties window, you can click on the View menu and click the name of the window, that particular window will appear.



**Figure 1.3: VB2010 IDE with A New Form**

- **Form**-The Form is the first place to build your application. It is the place to design the user interface.
- **Solution Explorer** -The solution explorer displays a list of projects, files and other components that you can easily browse and access. For example, it displays My Project and Form1.vb in Figure 1.3

- **Properties Window-** This is the place to set the properties of the objects in your application. The objects include the default form and the controls you place in the form. We will learn more about setting properties later.

### 1.3 Understanding the Concept of Object Oriented Programming

The main difference between VB2010 and Visual Basic 6 is that it is a full Object Oriented Programming Language while VB6 may have OOP capabilities, it is not fully object oriented. In order to qualify as a fully object oriented programming language, it must have three core technologies namely **encapsulation, inheritance** and **polymorphism**. Read more about the three terms in the box below:

**Encapsulation** refers to the creation of self-contained modules that bind processing functions to the data. These user-defined data types are called classes. Each class contains data as well as a set of methods, which manipulate the data. The data components of a class are called instance variables and one instance of a class is an object. For example, in a library system, a class could be member, and John and Sharon could be two instances (two objects) of the library class.

#### **Inheritance**

Classes are created according to hierarchies, and inheritance allows the structure and methods in one class to be passed down the hierarchy. That means less programming is required when adding functions to complex systems. If a step is added at the bottom of a hierarchy, then only the processing and data associated with that unique step needs to be added. Everything else about that step is inherited. The ability to reuse existing objects is a major advantage of object technology.

#### **Polymorphism**

Object-oriented programming allows procedures about objects to be created whose exact type is not known until runtime. For example, a screen cursor may change its shape from an arrow to a line depending on the program mode. The routine to move the cursor on screen in response to mouse movement would be written for "cursor," and polymorphism allows that cursor to take on whatever shape is required at run time. It also allows new shapes to be integrated easily.

VB2010 is a fully Object Oriented Programming Language, just like other OOP such as C++ and Java. It is different from the earlier versions of VB because it focuses more on the data itself while the previous versions focus more on the actions. Previous versions of VB are **procedural** or **functional** programming language. Some other procedural programming languages are C, Pascal and Fortran.

VB2010 allows users to write programs that break down into modules. These modules represent the real-world objects; we also call them classes or types. An object can be created out of a class, it is an instance of the class. A class can also comprise subclass. For example, apple tree is a subclass of the plant class and the apple in your backyard is an instance of the apple tree class. Another example is student class is a subclass of the population class while a student with the name John is an instance of the student class. A class consists of data members as well as methods. In VB2010, the program structure to define a population class can be written as follows:

```
Public Class Population
    'Data Members
    Private Name As String
    Private Birthdate As String
    Private Gender As String
    Private Age As Integer
    'Methods
    Overridable Sub ShowInfo( )
        MessageBox.Show(Name)
        MessageBox.Show(Birthdate)
        MessageBox.Show(Gender)
        MessageBox.Show(Age)
    End Sub
End Class
```

After you have created the population class, you can create a subclass that inherits the attributes or data from the population class. For example, you can create a student

class that is a subclass of the population class. Under the student class, you do not have to define any data fields that were already defined under the population class; you only have to define the data fields that are different from an instance of the population class. For example, you may want to include StudentID and Address in the student class. The program code for the StudentClass is as follows:

Public Class Student

Inherits Population

Public StudentID as String

Public Address As String

Overrides Sub ShowInfo( )

MessageBox.Show(Name)

MessageBox.Show(StudentID)

MessageBox.Show(Birthdate)

MessageBox.Show(Gender)

MessageBox.Show(Age)

MessageBox.Show(Address)

End Sub

### Summary

- In section 1.1, you learned about the evolution of Visual Basic.
- In section 1.2, you have learned how to launch the start page, the new project dialog and the designer interface. You have also learned that the designer interface consists of the Form, the Solution Explorer and the Properties window.
- You have also learned some basic concepts of object oriented programming, which comprises encapsulation, polymorphism and inheritance.