

ỦY BAN NHÂN DÂN TỈNH BR – VT

**TRƯỜNG CAO ĐẲNG NGHỀ**

---



# **GIÁO TRÌNH**

## **LẬP TRÌNH WINDOWS 1**

**NGHỀ LẬP TRÌNH MÁY TÍNH**

**TRÌNH ĐỘ CAO ĐẲNG VÀ TRUNG CẤP**

*Ban hành kèm theo Quyết định số: 01/QĐ-CDN, ngày 04 tháng 01 năm 2016  
của Hiệu trưởng trường Cao đẳng nghề tỉnh BR - VT*



***Bà Rịa – Vũng Tàu, năm 2016***

## **TUYÊN BỐ BẢN QUYỀN**

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

TaiLieu.vn

## **LỜI GIỚI THIỆU**

Windows Forms là cách cơ bản để cung cấp các thành phần giao diện (GUI components) cho môi trường .NET Framework. Windows Forms được xây dựng trên thư viện Windows API. Windows Forms cơ bản bao gồm:

- Một Form là khung dùng hiển thị thông tin đến người dùng.
- Các Control được đặt trong form và được lập trình để đáp ứng sự kiện.

Tailieu.vn

## **MỤC LỤC**

### **TUYÊN BỐ BẢN QUYỀN 1**

**Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo. 1**

**Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm. 1**

**1**

### **LỜI GIỚI THIỆU 2**

**Windows Forms là cách cơ bản để cung cấp các thành phần giao diện (GUI components) cho môi trường .NET Framework. Windows Forms được xây dựng trên thư viện Windows API. Windows Forms cơ bản bao gồm: 2**

- Một Form là khung dùng hiển thị thông tin đến người dùng. 2**
- Các Control được đặt trong form và được lập trình để đáp ứng sự kiện. 2**

### **MỤC LỤC 3**

#### **MÔ ĐUN LẬP TRÌNH WINDOWS 1 9**

#### **VỊ TRÍ, TÍNH CHẤT CỦA MÔN HỌC 9**

- + Mô đun được bố trí sau khi học sinh học xong các môn học chung, các môn học, mô đun kỹ thuật cơ sở, cấu trúc dữ liệu và giải thuật, lập trình hướng đối tượng, cơ sở dữ liệu, lập trình quản lý 1. 9**
- + Là mô đun đầu tiên của môn học lập trình Windows (VB.NET). Đây là mô đun chứa đựng kiến thức nền tảng về ngôn ngữ lập trình VB.NET, là mô đun hỗ trợ cho hai mô đun lập trình Windows 2 và lập trình Windows 3. 9**

**MỤC TIÊU: Sau khi học xong mô đun này, học sinh – sinh viên có năng lực 9**

- + Làm quen ngôn ngữ lập trình. 9**
- + Làm quen và sử dụng các kiểu dữ liệu cơ bản. 9**
- + Trình bày và sử dụng các câu lệnh, cấu trúc điều khiển, rẽ nhánh. 9**
  - + Trình bày và sử dụng các control đơn giản. 9**
- + Phân tích chức năng của chương trình ứng dụng để lên kế hoạch xây dựng chương trình ứng dụng. 9**
  - + Thiết kế giao diện chương trình ứng dụng. 9**
  - + Sử dụng các tiện ích trợ giúp của ngôn ngữ lập trình. 9**

- + Hoàn thành đồ án chương trình ứng dụng đơn giản. 9
- + Phát triển các chương trình đồ án thành các phần mềm quản lý (Nhà hàng, khách sạn, bán hàng, giáo dục...) trong thực tế. 9

## **NỘI DUNG: 9**

### **BÀI 1. GIỚI THIỆU MICROSOFT VISUAL STUDIO .NET 10**

#### **1. Giới thiệu Microsoft .NET 2010 10**

- 1.1. Tình hình trước khi Visual Studio.NET ra đời 10
- 1.2. Sự ra đời của Visual Studio.NET 11
- 1.3 Tổng quan về Visual Studio.NET 11
- 1.4 Trình biên dịch và MSIL 12

#### **2. Khởi động Visual Basic.NET 2010 và giao diện 12**

#### **3. Tạo ứng dụng đầu tiên 21**

#### **4. Cấu trúc của ứng dụng Visual Basic.NET 23**

- 4.1 Namespaces là gì? 23
- 4.2 Tạo một Namespace 23

#### **5. Bài tập 26**

### **BÀI 2. NỀN TẢNG CỦA NGÔN NGỮ VB.NET 27**

#### **1. Các kiểu dữ liệu 27**

#### **2. Biến 28**

- 2.1 Khái niệm 28
- 2.2 Khai báo biến 28
- 2.3 Khởi tạo giá trị cho biến 29

#### **3 Mảng 29**

- 3.1 Khái niệm 29
- 3.2 Khai báo 30
  - 3.2.1 Mảng có chiều dài cố định: 30
  - 3.2.2 Mảng động 30
- 3.2 Một số thao tác trên mảng 31

#### **4. Toán tử 32**

- 4.1 Khái niệm 32
- 4.2 Các loại phép toán 32

#### **5. Câu lệnh điều khiển 33**

- 5.1 Câu lệnh gán 33
- 5.2 Câu lệnh rẽ nhánh If 34
- 5.3 Câu lệnh lựa chọn Select Case 35
- 5.4 Toán tử Is & To 36
- 5.5 Cấu trúc lặp 37

| STT | Tên bài                                                            | Thời gian | Hình thức giảng dạy |
|-----|--------------------------------------------------------------------|-----------|---------------------|
| 1   | Giới thiệu tổng quan về Visual Studio .NET                         | 1         | Lý thuyết           |
| 2   | Nền tảng cơ bản ngôn ngữ lập trình                                 | 1         | Lý thuyết           |
| 3   | Sử dụng câu lệnh nhập xuất cơ bản, cấu trúc điều khiển If          | 3         | Tích hợp            |
| 4   | Sử dụng cấu trúc lựa chọn select..case, for, for Each              | 5         | Tích hợp            |
| 5   | Sử dụng cấu trúc vòng lặp While, Do...While                        | 5         | Tích hợp            |
| 6   | Sử dụng câu lệnh Break, Continue, GoTo                             | 4         | Tích hợp            |
|     | Kiểm tra bài 6                                                     | 1         | TH                  |
| 7   | Hàm                                                                | 4         | Tích hợp            |
|     | Kiểm tra bài 7                                                     | 1         | TH                  |
| 8   | Đăng ký đồ án                                                      | 1         | Lý thuyết           |
| 9   | Phân tích yêu cầu đồ án theo hướng đối tượng                       | 4         | Tích hợp            |
| 10  | Thiết kế các lớp đối tượng cho đồ án                               | 4         | Tích hợp            |
| 11  | Định nghĩa các thuộc tính, phương thức hành động cho lớp đối tượng | 4         | Tích hợp            |
| 12  | Sử dụng Form, MenuStrip                                            | 5         | Tích hợp            |
| 13  | Sử dụng Button, Label, TextBox                                     | 5         | Tích hợp            |
| 14  | Sử dụng Checkbox, RadioButton, GroupBox                            | 5         | Tích hợp            |
| 15  | Sử dụng ComboBox, ListBox                                          | 5         | Tích hợp            |
|     | Kiểm tra bài 15                                                    | 1         | TH                  |
| 16  | Sử dụng DateTimePicker, MonthCalendar                              | 5         | Tích hợp            |
| 17  | Sử dụng ListView, TreeView                                         | 5         | Tích hợp            |
| 18  | Sử dụng PictureBox, ToolTip                                        | 5         | Tích hợp            |
| 19  | Sử dụng CheckListBox, NumericUpDown                                | 5         | Tích hợp            |
| 20  | Sử dụng Panel, SplitContainer, TabControl                          | 5         | Tích hợp            |
| 21  | Truy xuất cơ sở dữ liệu bằng Dataset                               | 5         | Tích hợp            |
| 22  | Truy xuất cơ sở dữ liệu bằng DataGridView                          | 5         | Tích hợp            |
| 23  | Truy xuất cơ sở dữ liệu bằng BindingSource                         | 5         | Tích hợp            |
| 24  | Truy xuất cơ sở dữ liệu bằng BindingNavigator                      | 5         | Tích hợp            |

- 5.5.1 Lặp không biết trước số lần lặp 37
  - 5.5.1.1 Câu lệnh Do ... Loop 37
  - 5.5.1.2 Câu lệnh While ... End While 38
- 5.5.2 Lặp biết trước số lần lặp với câu lệnh For...Next 38

## **6. Xử lý lỗi 39**

- 6.1 Cú pháp Try...Catch 40
- 6.2 Sử dụng mệnh đề Finally 42
- 6.3 Cài đặt Try...Catch phức tạp hơn 42
- 6.4 Tự mình phát sinh lỗi 45
- 6.5 Sử dụng các khối Try...Catch lồng nhau 46
- 6.6 So sánh cơ chế xử lý lỗi với các kỹ thuật phòng vệ lỗi 47
- 6.7 Sử dụng phát biểu thoát Exit Try 47

## **7. Bài tập 48**

### **BÀI 3. LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG VISUAL BASIC .NET 50**

#### **1. Khái niệm hướng đối tượng 50**

- 1.1 Định nghĩa 50
- 1.2 Đặt điểm 50
  - 1.2.1 Tính trừu tượng 50
  - 1.2.2 Tính đóng gói 50
  - 1.2.3 Tính thừa kế 52
  - 1.2.4 Tính đa hình 52

#### **2. Lập trình hướng đối tượng trong VB.NET 52**

- 2.1 Tạo một class 52
- 2.2 Tạo class kế thừa 54
  - 2.2.1 Tính thừa kế (Inherits) 54
- 2.3 Constructor (Thủ tục khởi tạo) 57
- 2.4 Destructors (Thủ tục khởi hủy) 58
- 2.5 Phương thức (Methods) 59
- 2.6 Trường (Fields) và thuộc tính (Properties) 59
- 2.7 Khai báo sự kiện (Event) 60
- 2.8 Từ khóa Me, MyBase, MyClass 61
  - 2.8.1 Từ khóa Me 61
  - 2.8.2 Từ khóa MyBase 62
  - 2.8.3 Từ khóa MyClass 63
- 2.9 Giao diện (Interface) 64

#### **3. Xây dựng các lớp xử lý 66**

- 3.1 Mô hình đa tầng 66
  - 3.1.1 Presentation Layer 66
  - 3.1.2 Business Logic Layer 67
  - 3.1.3 Data Access Layer 68
- 3.2 Phân tích và thiết kế 69
  - 3.2.1 Business Entities 70
  - 3.2.2 Lớp CategoryService 71
  - 3.2.3 Data Access Components 72
- 3.3 Thiết kế cơ sở dữ liệu 72
  - 3.3.1 Hiện thực lớp Business Logic & Data Access 72
  - 3.3.2 Hiện thực Data Access Components 72
  - 3.3.3 Hiện thực lớp Business Logic 74

## **4. Bài tập 74**

### **BÀI 4. LÀM VIỆC VỚI DỰ ÁN CÓ NHIỀU FORM 76**

#### **1. Thiết kế thực đơn bằng MenuStrip 76**

- 1.1 Tạo Menu 76
- 1.2 Một số tùy biến cho Menu 77
  - 1.2.1 Thêm phím truy cập vào các mục chọn lệnh trên menu 77
  - 1.2.2 Thay đổi thứ tự các mục chọn 77
  - 1.2.3 Đặt tên và thuộc tính cho menu 78
- 1.3 Viết lệnh cho sự kiện của menu 79

#### **2. Thiết kế các dạng form 80**

- 2.1 Form cha 80
- 2.2 Form con 81

#### **3. Sử dụng các điều khiển cơ bản 83**

- 3.1 Mối quan hệ giữa thuộc tính, phương thức và sự kiện 83
- 3.2 Thuộc tính, phương thức, sự kiện của một số điều khiển cơ bản 84
  - 3.2.1 Form 84
  - 3.2.2 Hộp văn bản - TextBox 87
  - 3.2.3 Nút lệnh – Button 89
  - 3.2.4 Nhãn – Label 90
  - 3.2.5 Dòng mạch nước - ToolTip 91
- 3.3 Các hộp thoại thông dụng 92
  - 3.3.1 Hộp thoại mở tập tin (OpenFileDialog) 92
  - 3.3.2 Hộp thoại lưu tập tin (SaveFileDialog) 93
  - 3.3.3 Hộp thoại font 94
  - 3.3.4 Hộp thoại màu 94

#### **4. Làm việc với Module 95**



|                                                        |     |
|--------------------------------------------------------|-----|
| 4.1 Tạo và lưu module chuẩn                            | 95  |
| 4.2 Sử dụng các biến Public                            | 98  |
| 4.2.1 Làm việc với các biến Public (biến toàn cục)     | 98  |
| 4.2.2 Biến Public ở phạm vi form                       | 100 |
| 4.3 Tạo thủ tục (Procedure)                            | 101 |
| 4.3.1. Khai báo thủ tục                                | 101 |
| 4.3.2 Sử dụng các thủ tục - Sub                        | 101 |
| 4.3.3 Truyền đối số theo tham trị và tham biến         | 102 |
| 4.4 Khai báo hàm (Function)                            | 103 |
| 4.4.1 Cú pháp khai báo hàm                             | 103 |
| 4.4.2. Gọi hàm                                         | 104 |
| 4.4.3. Sử dụng hàm thực hiện tác vụ tính toán          | 104 |
| 4.4.5 Chạy chương trình:                               | 105 |
| <b>5. Làm quen với ADO.NET 106</b>                     |     |
| 5.1 Lập trình với ADO.NET                              | 106 |
| 5.1.1 Thuật ngữ về cơ sở dữ liệu (CSDL)                | 106 |
| 5.1.2 Làm việc với cơ sở dữ liệu Access                | 107 |
| 5.1.3 Tạo bộ điều phối dữ liệu Data Adapter            | 110 |
| 5.1.4 Sử dụng đối tượng điều khiển OleDbDataAdapter    | 110 |
| 5.1.5 Làm việc với DataSet                             | 114 |
| 5.2 Sử dụng các điều khiển ràng buộc dữ liệu           | 116 |
| 5.3 Tạo các điều khiển duyệt xem dữ liệu               | 119 |
| 5.4. Hiển thị vị trí của bản ghi hiện hành             | 121 |
| 5.5 Trình diễn dữ liệu sử dụng điều khiển DataGrid     | 123 |
| 5.5.1 Sử dụng DataGrid để hiển thị dữ liệu trong bảng: | 123 |
| 5.5.2 Định dạng các ô lưới trong DataGrid              | 130 |
| 5.5.3 Cập nhật cơ sở dữ liệu trở lại bảng              | 131 |
| <b>6. Bài tập 132</b>                                  |     |
| <b>Tài liệu tham khảo: 135</b>                         |     |

## **MÔ ĐUN LẬP TRÌNH WINDOWS 1**

### **VỊ TRÍ, TÍNH CHẤT CỦA MÔN HỌC**

- + Mô đun được bố trí sau khi học sinh học xong các môn học chung, các môn học, mô đun kỹ thuật cơ sở, cấu trúc dữ liệu và giải thuật, lập trình hướng đối tượng, cơ sở dữ liệu, lập trình quản lý 1.
- + Là mô đun đầu tiên của môn học lập trình Windows (VB.NET). Đây là mô đun chứa đựng kiến thức nền tảng về ngôn ngữ lập trình VB.NET, là mô đun hỗ trợ cho hai mô đun lập trình Windows 2 và lập trình Windows 3.

**MỤC TIÊU:** Sau khi học xong mô đun này, học sinh – sinh viên có năng lực

- + Làm quen ngôn ngữ lập trình.
- + Làm quen và sử dụng các kiểu dữ liệu cơ bản.
- + Trình bày và sử dụng các câu lệnh, cấu trúc điều khiển, rẽ nhánh.
- + Trình bày và sử dụng các control đơn giản.
- + Phân tích chức năng của chương trình ứng dụng để lên kế hoạch xây dựng chương trình ứng dụng.
- + Thiết kế giao diện chương trình ứng dụng.
- + Sử dụng các tiện ích trợ giúp của ngôn ngữ lập trình.
- + Hoàn thành đồ án chương trình ứng dụng đơn giản.
- + Phát triển các chương trình đồ án thành các phần mềm quản lý (Nhà hàng, khách sạn, bán hàng, giáo dục...) trong thực tế.

**NỘI DUNG:**

# BÀI 1. GIỚI THIỆU MICROSOFT VISUAL STUDIO .NET

*Mục tiêu của bài:*

Trình bày được cấu trúc Net Framework;  
Hiểu các tính năng của Visual Studio.Net 2010;  
Làm quen với giao diện của C#, VB.Net;  
Viết ứng dụng nhỏ trên C#, VB.net;

## 1. Giới thiệu Microsoft .NET 2010

### 1.1. Tình hình trước khi Visual Studio.NET ra đời

Với sự phát triển liên tục và đa dạng của thế giới công nghệ thông tin ngày nay, các phần mềm, các hệ điều hành, các môi trường phát triển và các ứng dụng liên tục ra đời. Tuy nhiên, đôi khi việc phát triển không đồng nhất và nhất là do không tương thích về mặt lợi ích của các công ty phần mềm lớn đã làm ảnh hưởng đến công việc của những kỹ sư xây dựng phần mềm.

Trong giới phát triển ứng dụng trên Internet ta có thể sử dụng các ngôn ngữ Java, PHP, ASP... Khi Java mới được Sun Corporation giới thiệu nó đã có một sức mạnh đáng kể và hướng tới việc chạy trên nhiều hệ điều hành khác nhau, độc lập với các bộ xử lý. Đặc biệt Java rất thích hợp cho việc viết các ứng dụng trên Internet. Tuy nhiên, Java lại có hạn chế về mặt tốc độ và trên thực tế vẫn chưa thịnh hành.

Để làm giảm khả năng ảnh hưởng của Java, bên hãng Microsoft cũng cung cấp ngôn ngữ ASP - chuyên dùng để viết các ứng dụng trên Web. Trong các trang ASP vừa chứa thẻ HTML vừa chứa các đoạn script (VBScript, JavaScript). Trong quá trình xử lý một trang ASP, nếu là thẻ HTML thì sẽ được gửi thẳng tới trình duyệt, còn nếu là các đoạn script thì sẽ được chuyển thành các dòng HTML rồi gửi đi. Khi nhà lập trình muốn đóng gói và sử dụng lại một số chức năng nào đó, thì họ dịch các đoạn chương trình thành ActiveX và đưa nó vào Web Server. Tuy nhiên, vì lý do bảo mật nên các Admin của các trang Web thường rất dè dặt khi cài ActiveX lạ trên máy của họ, ngoài ra việc tháo gỡ các phiên bản của ActiveX này cũng là công việc rất khó khăn.

Còn trong giới phát triển ứng dụng trên Windows ta có thể viết ứng dụng bằng Visual C++, Delphi, Visual Basic... đây là một số công cụ phổ biến và mạnh. Trong đó Visual C++ là một ngôn ngữ rất mạnh nhưng cũng rất khó sử dụng. Visual Basic thì đơn giản dễ học, dễ dùng nhất nên rất thông dụng nhưng hạn

chế là Visual Basic không phải ngôn ngữ hướng đối tượng và không hỗ trợ khả năng phát triển thuật toán.

Tóm lại trong giới lập trình theo Microsoft thì việc lập trình trên desktop cho đến lập trình hệ phân tán hay trên web là những mảng độc lập.

## ***1.2. Sự ra đời của Visual Studio.NET***

Đầu năm 1998, sau khi hoàn tất phiên bản Version 4 của Internet Information Server -IIS, đội ngũ lập trình của Microsoft nhận thấy họ còn có rất nhiều sáng kiến để có thể kiến tạo toàn IIS, và họ bắt đầu xây dựng một kiến trúc mới trên nền tảng ý tưởng đó và đặt tên là Next Generation Windows Services - NGWS. Tham vọng của họ là cung cấp một môi trường có thể dùng chung cho tất cả ngôn ngữ lập trình trong bộ Visual Studio cũng như cho các ngôn ngữ lập trình của các công ty khác.

Kết quả là năm 2001 Visual Studio.Net 2001 ra đời đánh dấu cho một môi trường lập trình trên nền .NET Framework 1.0 tiên tiến mới.

Năm 2003, sau 2 năm .NET Framework nâng cấp thêm một bậc với phiên bản 1.1 với đặc điểm ngoài các chương trình Windows truyền thống – là các tệp tin .exe giờ đây Windows còn tồn tại những chương trình khác – những chương trình chạy trên nền .NET. Muốn chạy chương trình .NET ta chỉ cần cài .NET Framework là đủ. Một điểm lý thú và cũng là điều mong đợi của tất cả lập trình viên, từ phiên bản Windows 2003 .NET Framework được cài đặt như một phần mặc định của Windows. Song song đó, môi trường phát triển Visual Studio .NET 2001 được nâng cấp thành Visual Studio .NET 2003 cho phép viết và chạy các ứng dụng trên nền .NET Framework 1.1

Cuối năm 2005, Visual Studio 2005 với nền .NET Framework 2.0 mạnh mẽ và vượt trội hơn so với nền .NET Framework 1.1 trước đó. Ngay sau đó Microsoft công bố phiên bản Windows Vista, và toàn bộ Windows là .NET, tất cả các hàm API lỗi trong những phiên bản Windows trước đây đều đã được thay thế bằng các hàm hay thư viện .NET. Microsoft đã viết lại hoàn toàn lõi API, không còn một lớp API nào nữa.

## ***1.3 Tổng quan về Visual Studio.NET***

Visual Studio.NET gồm 2 phần: Framework và Integrated Development Environment– IDE, cho phép lập trình viên khi xây dựng các ứng dụng có thể lựa chọn sử dụng nhiều ngôn ngữ lập trình khác nhau như Visual C++.NET, Visual C#.NET, Visual J#.NET, Visual Basic.NET... trong cùng một môi trường phát triển IDE thống nhất trên kiến trúc .NET Framework.

Framework là thành phần quan trọng nhất, là cốt lõi và tinh hoa của môi trường .NET, Framework giúp chúng ta biên dịch và thực thi các ứng dụng .NET (cấu trúc của Framework chúng ta sẽ tìm hiểu ở các chương sau của giáo trình).

IDE cung cấp một môi trường phát triển trực quan, giúp các lập trình viên có thể dễ dàng và nhanh chóng xây dựng giao diện cũng như viết mã lệnh cho các ứng dụng dựa trên nền tảng .NET. Nếu không có IDE chúng ta cũng có thể dùng một trình soạn thảo văn bản bất kỳ, ví dụ như Notepad để viết mã lệnh và sử dụng command line để biên dịch và thực thi ứng dụng. Tuy nhiên việc này mất rất nhiều thời gian, tốt nhất là chúng ta nên dùng IDE để phát triển các ứng dụng, và đó cũng là cách dễ sử dụng nhất.

Ngoài ra trong Visual Studio.NET thì lập trình Winform và Webform là tương tự, ví dụ cả Visual C#.NET lẫn Visual Basic.NET đều hỗ trợ khả năng lập trình trên Win và Web...

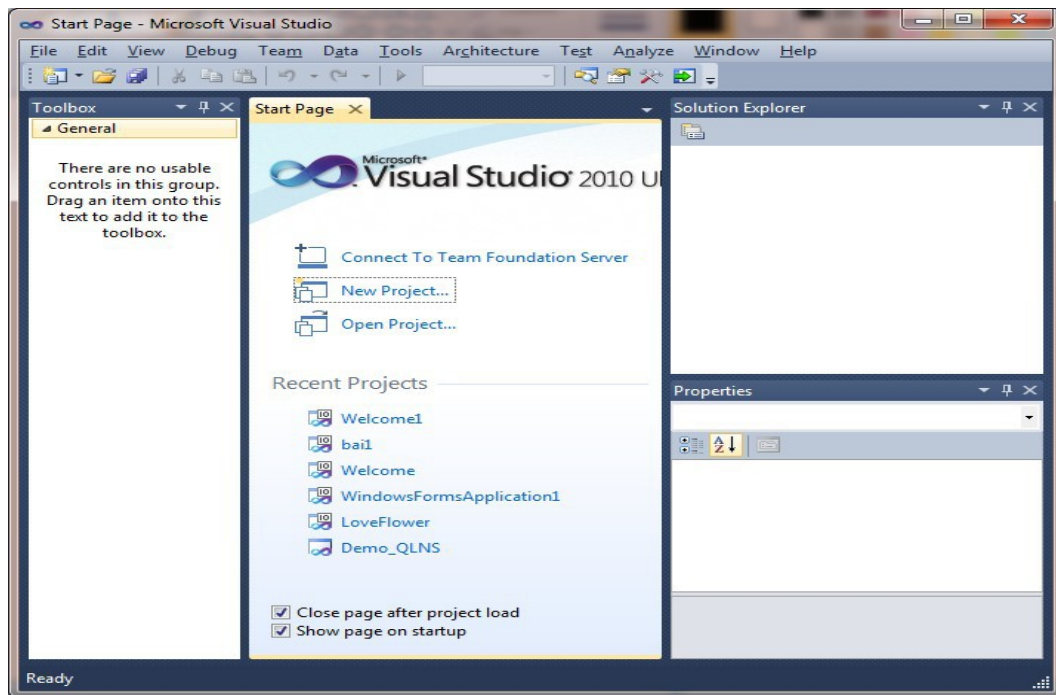
#### ***1.4 Trình biên dịch và MSIL***

Microsoft Intermediate Language (MSIL) hay Common Intermediate Language (CIL) là một ngôn ngữ trung gian được tạo ra sau quá trình biên dịch từ các loại ngôn ngữ khác trong .Net như C#, C++, VB.Net, J#, ...

Tất cả mã nguồn .NET đều được biên dịch thành MSIL. Sau đó MSIL sẽ được chuyển thành mã máy khi phần mềm được cài đặt hoặc khi chạy (run-time) bởi trình biên dịch JIT (Just-In-Time).

### **2. Khởi động Visual Basic.NET 2010 và giao diện**

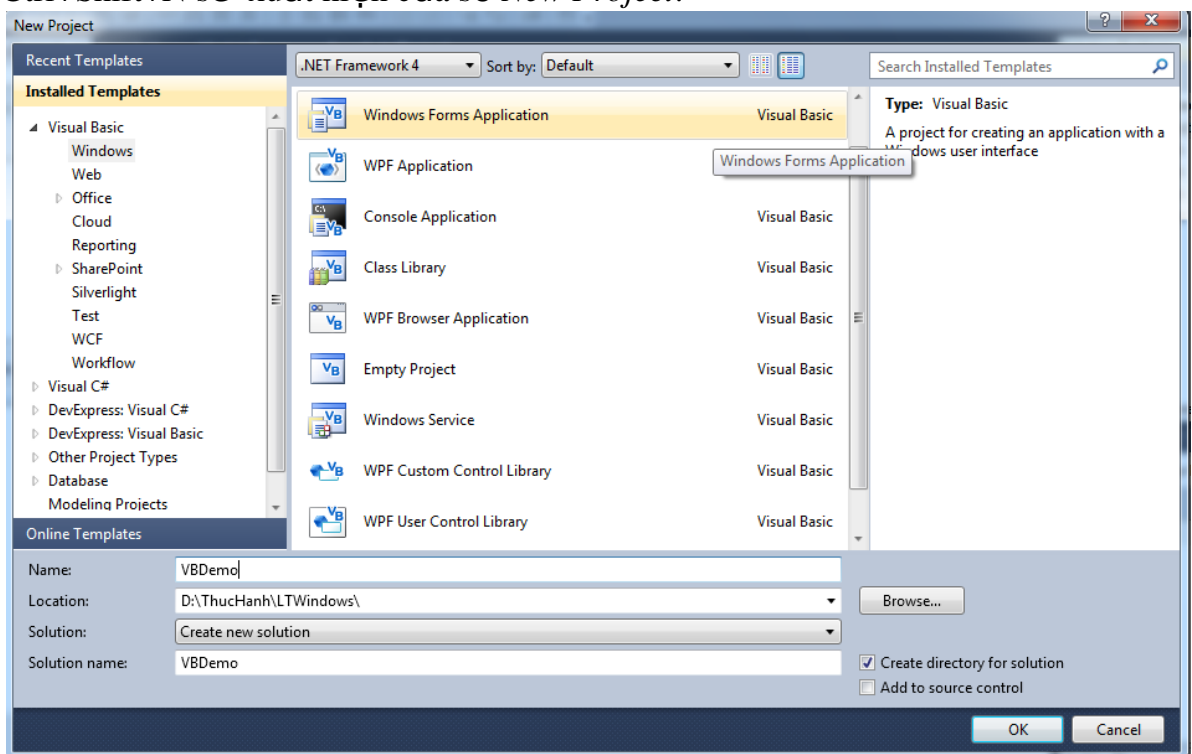
Để khởi động Visual C# 2010 và giao diện: Vào Start/Programs/Microsoft Visual Studio 2010/Microsoft Visual Studio 2010, xuất hiện cửa sổ *Start Page*.



Hình 1. Cửa sổ Start Page

- + New Project: Tạo đồ án mới.
- + Open Project: Mở các đồ án có sẵn.
- + Recent Projects: Danh sách các đồ án gần đây nhất.

Sau đó kích chọn mục New Project hoặc vào File/New/Project hoặc bấm phím tắt Ctrl+Shift+N sẽ xuất hiện cửa sổ New Project.



Hình 2. Cửa sổ New Project

Chọn ngôn ngữ *Visual Basic* và ứng dụng *Windows*.

Đặt tên cho đồ án tại mục *Name*.

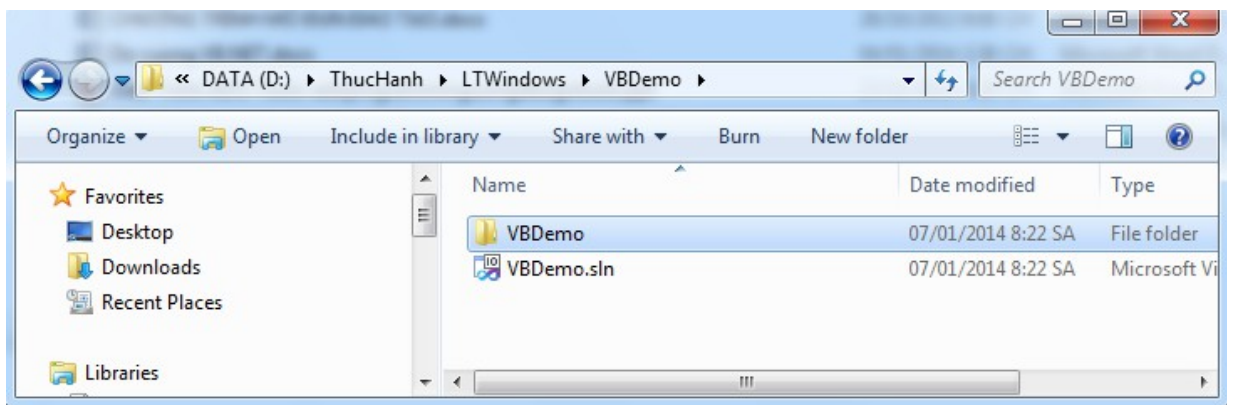
Chọn đường dẫn lưu đồ án tại mục *Location*.

Chọn *OK* để tạo một đồ án mới.

\* Lưu ý:

Mục *Create directory for solution* cho phép tạo một thư mục tại *Location* chứa tất cả

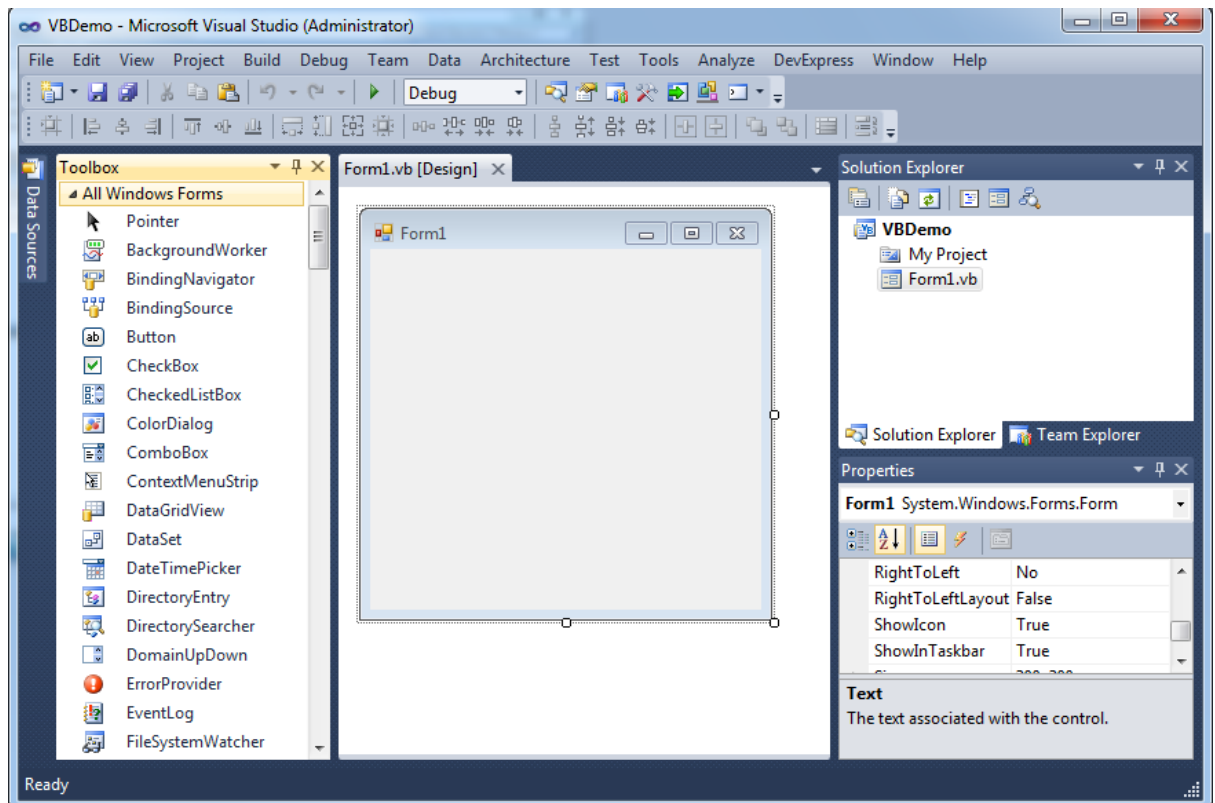
các tệp phát sinh của đồ án (nếu không các tệp của đồ án sẽ được lưu tại *Location*).



Hình 3. Nơi lưu trữ đồ án

Kết quả xuất hiện cửa sổ môi trường phát triển tích hợp IDE, với giao diện và các thành phần cơ bản như sau:





Hình 4. Môi trường phát triển tích hợp IDE

**Title Bar:** Thanh tiêu đề chứa tên đồ án.

**Menu Bar:** Thanh Menu chứa đầy đủ các công cụ cần để phát triển, thực thi và cài đặt

Ứng dụng...

*File:* cho phép mở, thêm mới và lưu trữ đồ án...

*Edit:* gồm các thao tác hỗ trợ việc soạn thảo mã lệnh như: copy, cắt, dán...

*View:* cho phép hiển thị các công cụ hỗ trợ người dùng trong quá trình xây dựng đồ án như:

- Cửa sổ viết mã lệnh - Code
- Form thiết kế - Designer
- Hộp công cụ - Toolbox
- Thanh công cụ - Toolbars
- Cửa sổ thuộc tính - Properties Window...

*Project:* cho phép bổ sung các đối tượng khác nhau vào đồ án như: các form, các

component, các modul, các lớp...

*Built:* cho phép biên dịch đồ án.

*Debug:* cho phép chạy và gỡ rối chương trình.

*Data:* cho phép thêm mới và hiển thị cơ sở dữ liệu của đồ án.



*Tools*: cung cấp các công cụ cho phép kết nối tới các thiết bị ngoại vi như Pocket PC, Smartphone... hoặc kết nối tới các hệ quản trị cơ sở dữ liệu cũng như kết nối tới máy chủ server...

**Toolbar**: thanh công cụ gồm một tập hợp các nút lệnh, mỗi nút lệnh chứa một biểu tượng icons và có chức năng tương đương với chức năng của một mục lựa chọn trong thanh menu. Thanh công cụ rất hữu ích và trực quan, giúp người dùng dễ dàng và nhanh chóng thực hiện một chức năng mong muốn chỉ thông qua một cái kích chuột.

Visual Basic 2010 có tới 39 thanh công cụ khác nhau như: Standard, Formatting, Debug, Build... Ví dụ hình ảnh thanh công cụ Standard:



Hình 5. Thanh công cụ Standard

Để gọi các thanh công cụ ta vào *View/Toolbars* khi đó sẽ xuất hiện danh sách tất cả các thanh công cụ. Muốn ẩn/hiện thanh công cụ nào ta kích chọn tại dòng chứa tên thanh công cụ đó.

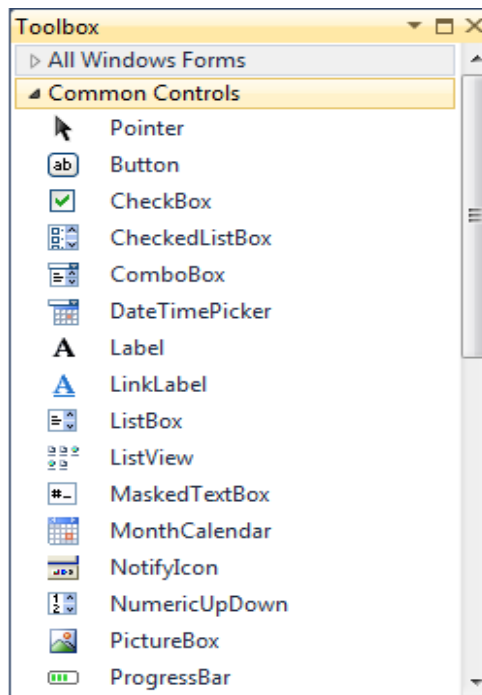
**Toolbox**: là hộp công cụ chứa các điều khiển – controls được đặt lên Form khi thiết kế giao diện người dùng.

Để hiển thị hộp công cụ ta thực hiện một trong các cách sau:

Vào *View/Toolbox*

Bấm tổ hợp phím *Ctrl+W+X*

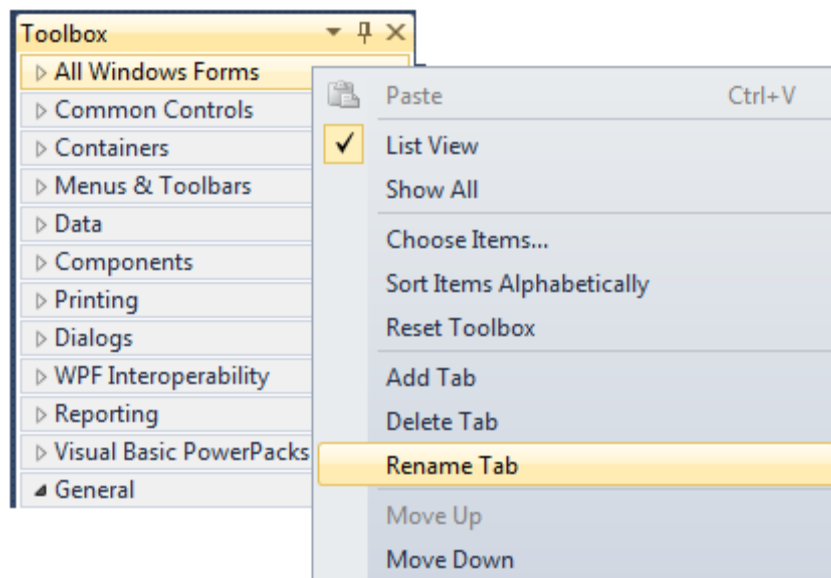
Kích chuột tại biểu tượng Toolbox  trên thanh công cụ Standard.



Hình 6. Hộp công cụ Toolbox

Mặc định hộp công cụ được chia thành 11 tab khác nhau như: *All Windows Forms*, *Common Controls*...

Ta có thể thêm mới, loại bỏ, đổi tên... các tab bằng cách kích chuột phải tại vị trí bất kỳ trên tab, xuất hiện một menu ngữ cảnh cho phép lựa chọn các thao tác cần thực hiện.

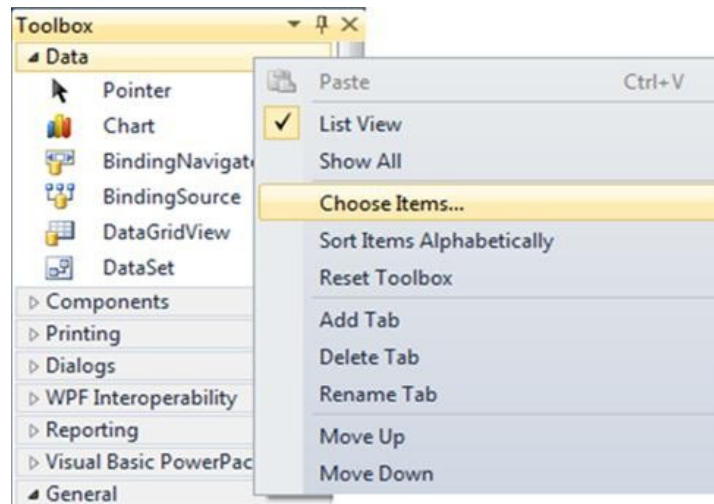


Hình 7. Các chức năng làm việc với tab trong Toolbox

Trong mỗi tab của hộp Toolbox chứa danh sách các loại điều khiển khác nhau, các điều khiển này có thể thêm mới, loại bỏ, thay đổi vị trí... Kích chuột

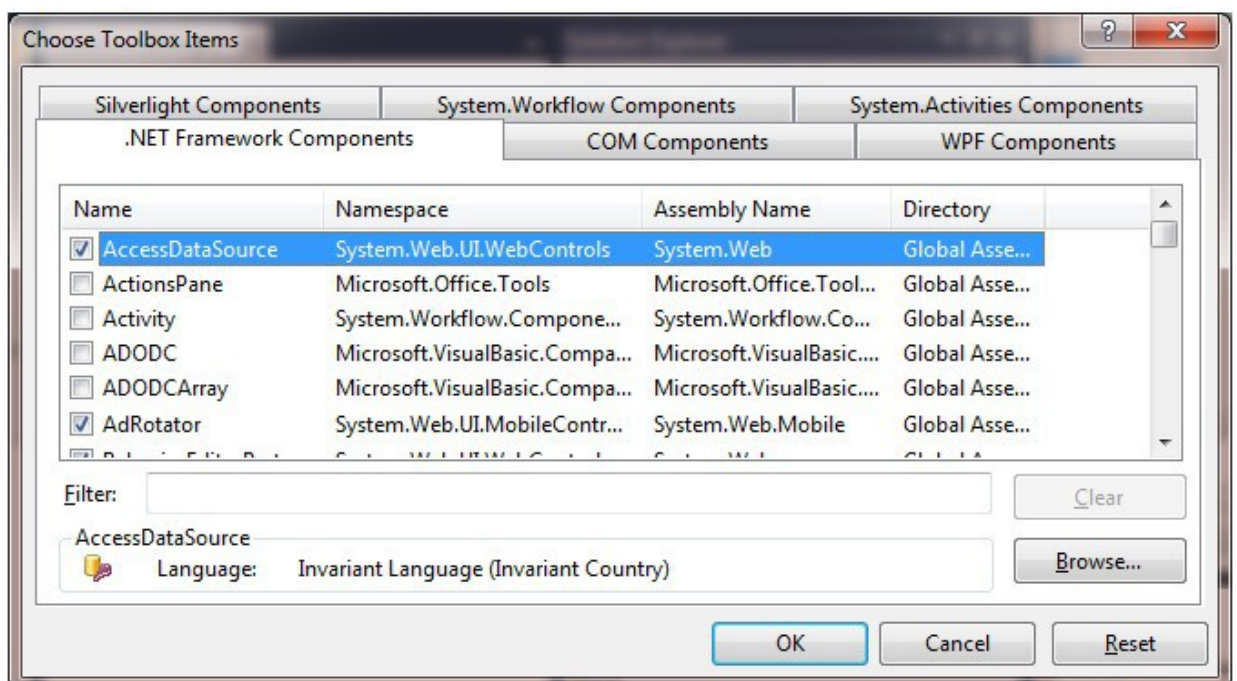
phải tại một điều khiển bất kỳ trên tab, xuất hiện một menu ngữ cảnh cho phép lựa chọn các thao tác cần thực hiện.

Ví dụ để thêm mới một điều khiển vào trong tab Data, ta kích chuột phải tại vị trí bất kỳ trên tab Data, chọn *Choose Items...*



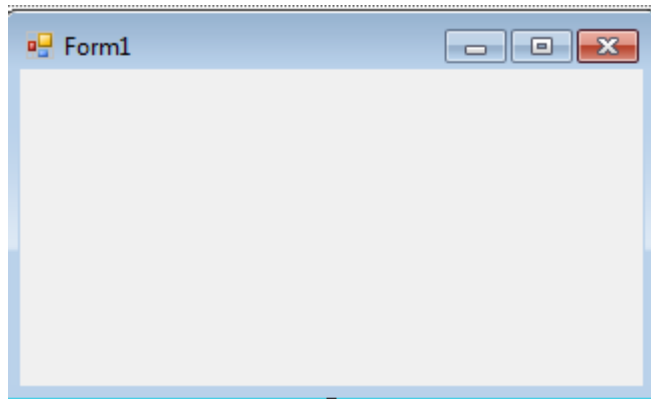
Hình 8. Các chức năng làm việc với từng điều khiển trong tab

Kết quả sẽ xuất hiện cửa sổ *Choose Toolbox Items*, kích chọn các điều khiển mong muốn rồi bấm OK để kết thúc.



Hình 9. Cửa sổ Choose Toolbox Items

**Form Designer:** cửa sổ thiết kế dùng để thiết kế giao diện cho chương trình, mỗi dự án có thể có một hoặc nhiều Form.



Hình 10. Cửa sổ Form Designer

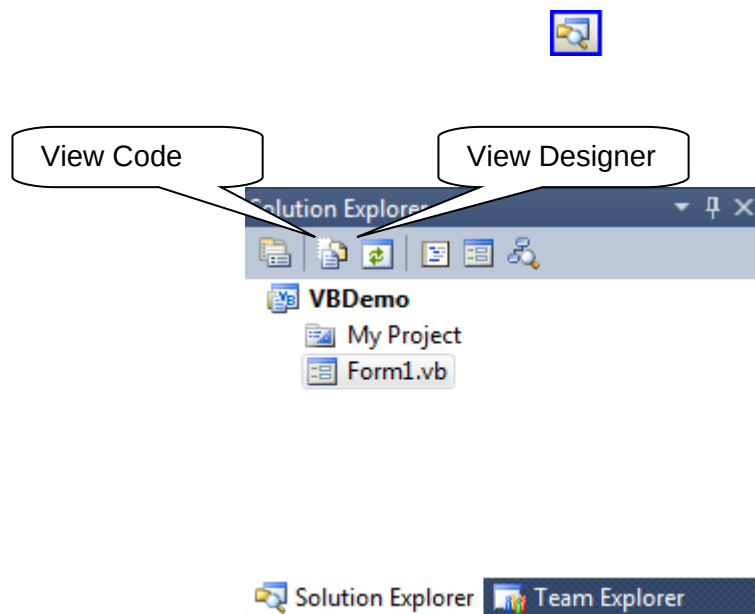
**Solution Explorer:** cửa sổ giải pháp - đây là phần cửa sổ giúp ta quản lý tất cả các tài nguyên và tập tin dự án.

Solution Explorer được tổ chức thành một cấu trúc cây bao gồm những mục khác nhau, như: danh sách các Form của đồ án, danh sách các lớp Class, danh sách các tài nguyên cũng như danh sách cơ sở dữ liệu...

Để hiển thị cửa sổ Solution Explorer ta thực hiện một trong các cách sau:

Vào *View/Solution Explorer*

Bấm tổ hợp phím *Ctrl+W+S*



Hình 11. Cửa sổ Solution Explorer