

## Chương 1. GIỚI THIỆU MÔI TRƯỜNG LẬP TRÌNH

### Mục tiêu:

Sau khi học xong chương này, sinh viên có thể:

- + Hiểu được sự khác nhau giữa chương trình Windows và chương trình DOS.
- + Biết sự phát triển của lập trình Windows qua các thế hệ.
- + Hiểu biết tổng quan về .NET Framework với hai thành phần chính là Framework Class Library và Common Language Runtime.
- + Biết sử dụng các công cụ và chức năng cơ bản của môi trường phát triển tích hợp Microsoft Visual Studio.NET hay Microsoft Visual C# để viết chương trình Windows dùng ngôn ngữ C#.

### 1. Giới thiệu ngôn ngữ Visual Studio.Net

Visual Studio .NET (VS.NET) là ngôn ngữ lập trình hướng đối tượng (Object Oriented Programming Language) do Microsoft thiết kế, là một ngôn ngữ lập trình hoàn toàn mới trên nền Microsoft's .NET Framework.

.NET là tầng trung gian giữa các ứng dụng và hệ điều hành. Tầng .NET cung cấp mọi dịch vụ cơ bản giúp tạo ra các chức năng mà ứng dụng đòi hỏi.

Mục tiêu chính của .NET là giúp giảm thiểu tối đa công việc thiết kế hệ thống phân tán (distributed system). Bộ sưu tập *.NET Enterprise Servers* bao gồm:

- Server Operation Systems: Hệ điều hành Server như: MS Windows Server, Advanced Server và Data Center Server.
- Clustering và Load Balancing Systems: Hệ thống cân bằng tải như MS Application Center, MS Cluster Server.
- Database System: Hệ quản trị cơ sở dữ liệu như MS SQL Server, Access.
- E-Mail System: Hệ thống quản lý mail, như MS Exchange Server.
- Data-transformation engine trên cơ sở XML: truyền dữ liệu dạng XML như MS Biz lk Server.
- Accessing Legacy Systems: Hệ thống hỗ trợ truy xuất các phương thức, ứng dụng, kỹ thuật như Host Integration Server

### 2. Tổng quan .Net Framework

.NET Framework là một nền tảng lập trình và cũng là một nền tảng thực thi ứng dụng chủ yếu trên hệ điều hành Microsoft Windows được phát triển bởi Microsoft. Các chương trình được viết trên nền .NET Framework sẽ được triển khai trong môi trường phần mềm được biết đến với tên Common Language Runtime (CLR). Môi trường phần mềm này là một máy ảo trong đó

cung cấp các dịch vụ như an ninh phần mềm (*security*), quản lý bộ nhớ (*memory management*), và các xử lý lỗi ngoại lệ (*exception handling*).

.NET framework bao gồm tập các thư viện lập trình lớn, những thư viện này hỗ trợ việc xây dựng các chương trình phần mềm như lập trình giao diện; truy cập, kết nối cơ sở dữ liệu; ứng dụng web; các giải thuật, cấu trúc dữ liệu; giao tiếp mạng... CLR cùng với bộ thư viện này là 2 thành phần chính của .NET framework.

.NET framework đơn giản hóa việc viết ứng dụng bằng cách cung cấp nhiều thành phần được thiết kế sẵn, người lập trình chỉ cần học cách sử dụng và tùy theo sự sáng tạo mà gắn kết các thành phần đó lại với nhau. Nhiều công cụ được tạo ra để hỗ trợ xây dựng ứng dụng .NET, và IDE (*Integrated Development Environment*) được phát triển và hỗ trợ bởi chính Microsoft là Visual Studio

### **3. Môi trường phát triển tích hợp Microsoft Visual Studio.NET**

Là một môi trường lập trình dùng để tạo ra các ứng dụng giao diện người dùng (graphical user interface (GUI)) trong các hệ điều hành thuộc họ Windows. Nó bao gồm một nhóm các công cụ giúp người lập trình tạo ra các chương trình máy tính một cách thuận tiện, năng suất cao mà ít phạm các lỗi.

Trong học phần này sử dụng môi trường phát triển Microsoft Visual C#. Microsoft Visual C# là một phần của bộ Microsoft Visual Studio.NET hay tự hoạt động một mình (bộ Microsoft Visual C# Professional).

#### **3.1. Project (Dự án) và Solution (Giải pháp)**

##### **3.1.1. Project**

Một ứng dụng .NET điển hình bao gồm nhiều file và folder liên quan nhau: file mã nguồn, file assembly (hay thư viện), file tài nguyên, ... Visual Studio tổ chức những file và folder này trong một bộ chứa gọi là project. Project được biên dịch thành một assembly, tức như một file thực thi (\*.EXE) hay một thư viện liên kết động (DLL).

Thông tin mô tả project lưu trong \*.Csproj.

##### **3.1.2. Solution**

Solution điển hình chứa một project hay nhiều project có liên quan nhau. Khi chúng ta tạo ra một project mới, Visual Studio tự động tạo ra solution mới chứa project, hoặc chúng ta có thể mở một solution đã có và add project mới vào. Thông tin mô tả solution lưu trong file \*.sln và file này được dùng để mở solution (nhấp kép lên tên \*.sln trong Windows Explorer).

Solution có thể không chứa project nào, chỉ chứa các nội dung như: lược đồ cơ sở dữ liệu, các file code mẫu, các ghi chú, ... Các solution loại này không được build nhưng có thể soạn thảo, hiệu chỉnh trong Solution Explorer.

## 3.2. Giao diện Microsoft Visual Studio.NET

### 3.2.1. Khởi động

Khởi động Microsoft Visual Studio.NET: Start → (All) Programs → Microsoft Visual Studio 2015 → Microsoft Visual Studio 2015.

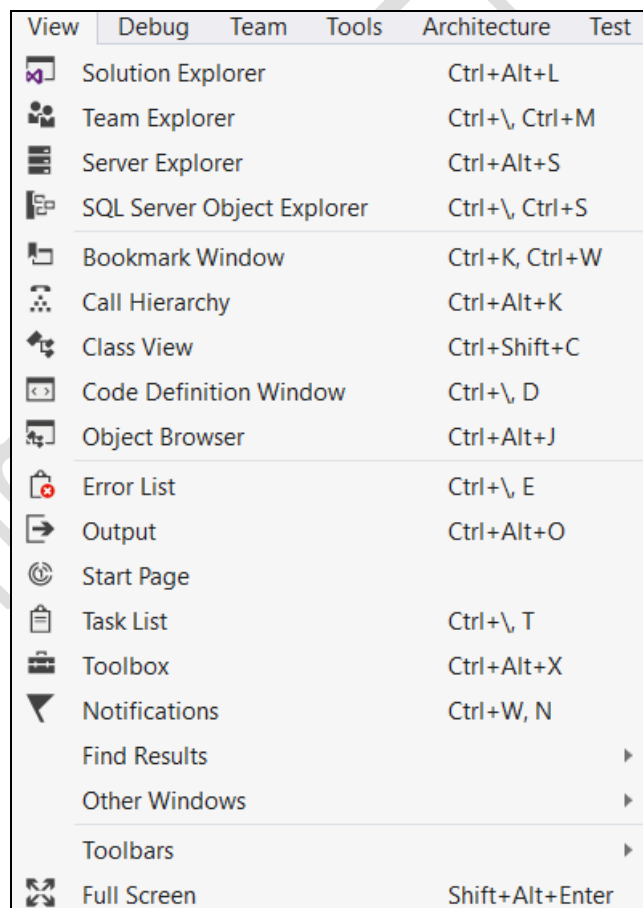
Start page: xuất hiện đầu tiên sau khi khởi động MS Visual C#. Khung Recent Projects cho phép mở project đã có hay tạo project mới.

### 3.2.2. Các cửa sổ

Visual Studio .NET (VS.NET) là một ứng dụng MDI (Multiple Document Interface). Nó bao gồm một cửa sổ cha (cửa sổ chính) duy nhất chứa menu, các toolbar và nhiều cửa sổ con khác. Một cửa sổ con có thể được ẩn hay hiện trên vùng làm việc của cửa sổ cha với kích thước hay vị trí được điều chỉnh khác nhau.

\* Hiện thị cửa sổ (con)

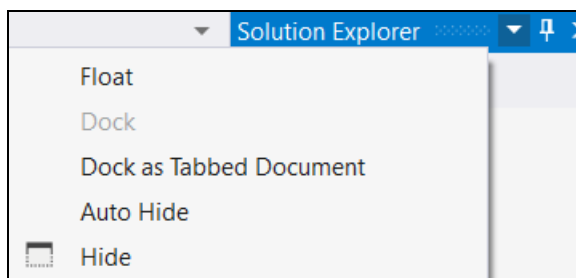
Vào menu View → Chọn tên cửa sổ muốn hiện thị trong danh sách của sổ như hình bên dưới:



**Hình 1.** Danh sách cửa sổ của VS.NET

### \* Định vị trí cửa sổ

Nhấp phải trên thanh tiêu đề của cửa sổ (hay nhấp lên nút Window position trên thanh tiêu đề) sẽ xuất hiện menu như hình bên dưới: chọn một kiểu định vị trong menu.



**Hình 2.** Kiểu định vị cửa sổ

- + Float: cửa sổ có thể kéo và thả “nổi” tại vị trí tùy ý trên desktop.

- + Dock: cửa sổ có thể được kéo và “neo” áp vào một cạnh của một cửa sổ (cửa sổ chính hay một cửa sổ con nào đó). Khi kéo cửa sổ Dock sẽ xuất hiện các nút định vị chỉ các phía của biên màn hình (phải hay trái, trên, dưới, giữa), rê trỏ chuột (trong lúc vẫn đang kéo cửa sổ) đến chọn một trong các nút này, cửa sổ Dock sẽ được neo áp với phía tương ứng với nút chọn.

- + Dock as Tabbed Document: cửa sổ sẽ thành một tab trên vùng làm việc của cửa sổ chính.

- + Auto Hide: cửa sổ thu nhỏ thành một tab khi rê chuột ra ngoài cửa sổ. Khi rê trỏ trên tab này thì cửa sổ hiện ra lại. Nút Pushpin thẳng đứng: đang tắt Auto Hide (Auto Hide không hiệu lực). Nút Pushpin nằm ngang: đang bật Auto Hide.

- + Hide: cửa sổ được ẩn.

### 3.2.3. Menu và Toolbar

\* Menu và Toolbar là theo ngữ cảnh, tức các mục của menu và các nút của toolbar sẽ xuất hiện phụ thuộc vào phần nào của IDE đang được chọn và hoạt động nào đang được phép.

\* Cách hiện và ẩn các toolbar như thông thường (View → Toolbars → ... hay nhấp phải một toolbar nào đó).

\* Một số menu giúp gỡ mã nguồn chính xác:

- + Edit → IntelliSense

- + Tools → Options → Environment. Tools → Options → Text Editor

- + Help → Search

- + Help → Dynamic Help

### 3.3. Biên dịch (Build) và Chạy (Run) chương trình

#### 3.3.1. Biên dịch chương trình

\* Build solution: menu Build → Build Solution (hay nhấn F6)

\* Build project: menu Build → Chọn Build [tên project] (hay nhấn Shift + F6)

Báo cáo về quá trình build hiển thị trong cửa sổ Output. Sau khi build mã nguồn, ta thu được file EXE (hay file DLL).

\* Quản lý cấu hình (Configuration) khi build: cấu hình qui định cách thức biên dịch (và liên kết) mã nguồn. Project có sẵn hai cấu hình: Debug và Release, mặc định là Debug.

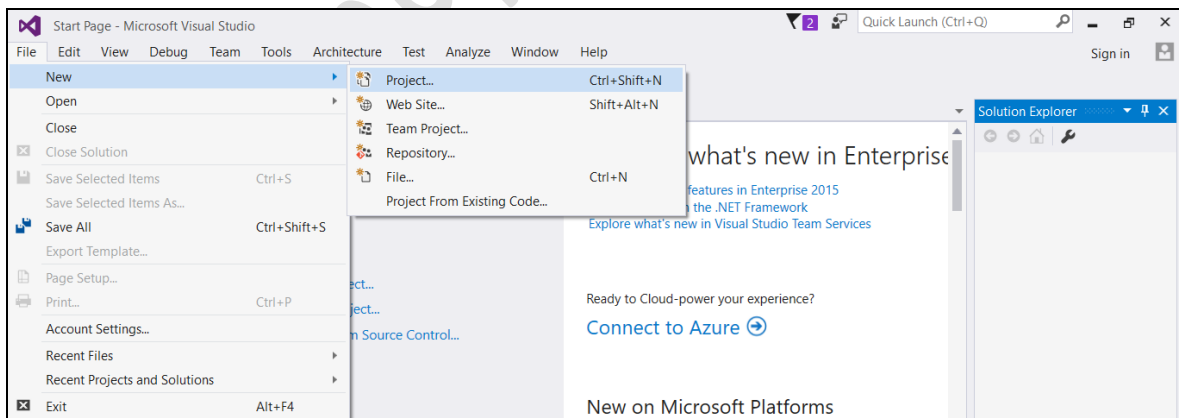
+ Cấu hình Debug: quá trình biên dịch được gắn thêm các thông tin debug, nhưng kết quả biên dịch sản sinh nhiều file hơn và file có dung lượng lớn hơn.

+ Cấu hình Release: quá trình biên dịch không gắn thêm thông tin debug. Kết quả biên dịch được tối ưu hóa về hiệu năng thực thi và kích thước file.

+ Chọn cấu hình hiện hành (trong số các cấu hình đã thiết lập): menu Build → Configuration Manager... → Active Solution Configuration → Chọn một cấu hình. Hay có thể chọn trong combo box Solution Configuration trên thanh Standard.

#### 3.3.2. Tạo project mới

Để tạo một Project mới vào File → New Project hoặc nhấn Ctrl + N

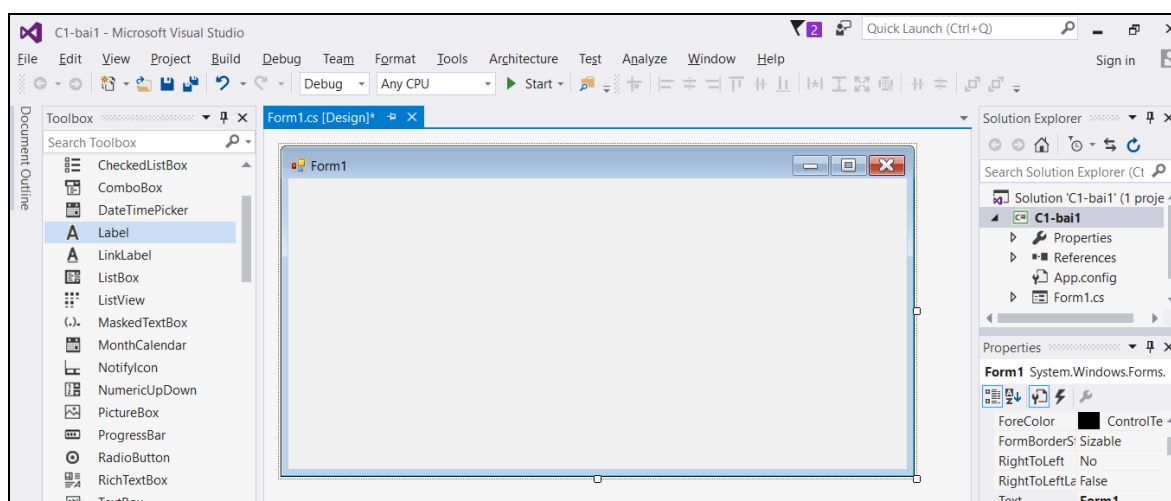


**Hình 3.** Minh họa cách tạo một Project

#### 3.3.3. Mở một project

Tại Start Page, có thể chọn một project của phần Recent Projects, vào mục *Open Project* hoặc vào *File → Open Project* sau đó chọn đường dẫn đến Project cần mở.

### 3.3.4. Môi trường làm việc của Visual Studio



**Hình 4.** Môi trường làm việc của VS.NET

Mô tả ý nghĩa các menu chính của VS.NET:

**Bảng 1.1.** Mô tả ý nghĩa các menu chính của Visual Studio

Tên	Ý nghĩa
File	Tiêu chuẩn chung cho mọi ứng dụng trong nền Windows. File dùng để mở hay đóng các tập tin hay dự án (project).
Edit	Cung cấp các chọn lựa khi soạn mã nguồn và dùng công cụ lập trình: Undo, Redo, Cut, Copy, Paste, Delete.
View	Bật/tắt các công cụ Windows tạo môi trường của IDE như: Solution Explorer, Properties, Output, ToolBox, Server Explorer,...
Project	Dùng để quản lý dự án (Project) bằng cách thêm vào hay xóa bỏ các tập tin liên hệ.
Build	Cho phép xây dựng và thực thi ứng dụng một cách độc lập bên ngoài IDE.
Debug	Giúp tìm ra các lỗi lập trình trong môi trường IDE và giúp kiểm tra từng bước một các mã nguồn trong dự án.
Data	Giúp kết nối để sử dụng thông tin trong cơ sở dữ liệu (database).
Tools	Chứa các công cụ bố trí Microsoft Visual Studio.Net IDE.
Windows	Tiêu chuẩn chung dùng quản lý mọi Windows trong IDE.
Help	Cung cấp các yêu cầu giúp đỡ với Microsoft Visual Studio.Net documentation từ mạng Internet.
Solution Explorer	Quản lý các tập tin đang có trong ứng dụng.

Tên	Ý nghĩa
Class View	Quản lý các lớp điều khiển đang có trong ứng dụng.
Properties Window	Màn hình quản lý các thuộc tính của điều khiển được chọn trên Form.
Toolbar	Chứa các điều khiển dùng để thiết kế Form, chia làm nhiều nhóm: <ul style="list-style-type: none"> <li>▪ Windows Forms: tập hợp các điều khiển dùng để thiết kế giao diện cho ứng dụng Windows.</li> <li>▪ WebForms: tập hợp các điều khiển để thiết kế giao diện cho ứng dụng Web.</li> <li>▪ Data : tập hợp các điều khiển dùng để kết nối với cơ sở dữ liệu.</li> </ul>

### 3.3.5. Chạy chương trình

Dùng một trong các lệnh sau:

+ Menu Debug → Start Debugging (hay nhấn F5, hay nhấp nút Start Debugging ()) trên thanh công cụ Standard).

+ Menu Debug → Start Without Debugging (hay nhấn Ctrl + F5).

### 3.3.6. Các loại lỗi

Có ba loại lỗi xảy ra bao gồm:

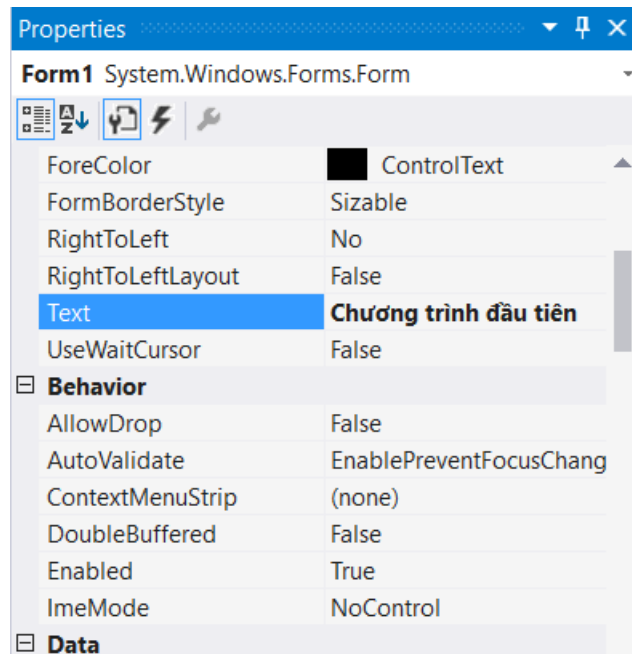
- Lỗi cú pháp (Syntax Error): Còn được gọi là lỗi biên dịch – Compiler error. Lỗi này do gõ sai cấu trúc ngôn ngữ. Lỗi này bộ soạn thảo mã có thể bắt được (xuất hiện dòng gạch chân màu xanh dưới dòng mã gây lỗi).

- Lỗi thực thi (Runtime error): Xảy ra bất ngờ khi chương trình đang thực thi.

Lỗi logic (Logic error): Lỗi do tư duy sai dẫn đến kết quả sai so với dự kiến.

### 3.3.7. Cửa sổ thuộc tính Properties

Cho phép thay đổi thông số của đối tượng thiết kế Form sau này. Để mở cửa sổ Properties, nhấp phải vào đối tượng chọn Properties:



**Hình 5.** Minh họa cửa sổ Properties của Form

### 3.3.8. Thoát khỏi Visual Studio

Chọn *File* → *Exit*

Lưu lại những việc đã làm bằng cách chọn *File* → *Save All*.

## BÀI TẬP

**Bài 1.** Cài đặt VS.NET và thiết lập môi trường làm việc của VS.Net.

**Bài 2.** Làm quen với ứng dụng đầu tiên trên VS.NET

