

BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔNG
TRƯỜNG CAO ĐẲNG CƠ GIỚI NINH BÌNH

GIÁO TRÌNH

MÔ ĐUN: LẬP TRÌNH C# 2

NGHỀ: LẬP TRÌNH MÁY TÍNH

TRÌNH ĐỘ: CAO ĐẲNG NGHỀ

Ban hành kèm theo Quyết định số: /QĐ-... ngày.....tháng....năm 2018
của Trường Cao đẳng Cơ giới Ninh Bình



Ninh Bình, năm 2018

TaiLieu.vn

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

Đây là tài liệu được biên soạn theo chương trình khung trình độ đào tạo Cao đẳng nghề Lập trình máy tính. Để học tốt môn học này, người học cần phải có kiến thức về lập trình trên Windows cơ bản, cụ thể là ngôn ngữ C#.NET. Lập trình Windows 2 là một mô đun lập trình nâng cao nhằm giúp người học có kiến thức và kỹ năng để xây dựng một ứng dụng chuyên nghiệp với cơ sở dữ liệu. Với phạm vi của tài liệu này, chúng tôi cung cấp cho người học các kiến thức và kỹ năng chính sau:

- Tạo các ứng dụng trên nền Windows.
- Tạo được các ứng dụng cơ sở dữ liệu trên nền Windows.
- Lập trình và sử dụng được các đối tượng của .NET.
- Tạo được ứng dụng cơ sở dữ liệu với các báo cáo bằng CrystalReport.
- Tạo ra các ứng dụng MDI.

Trong quá trình biên soạn, chúng tôi có tham khảo nhiều nguồn tài liệu trên Internet. Mặc dù rất cố gắng nhưng chắc chắn không tránh khỏi những thiếu sót, tác giả rất mong nhận được những ý kiến đóng góp để tài liệu ngày càng hoàn thiện hơn để cung cấp cho người học những kiến thức và kỹ năng thiết thực. Tài liệu này được thiết kế theo từng mô đun/ môn học thuộc hệ thống mô đun/ môn học của một chương trình để đào tạo hoàn chỉnh nghề Lập trình máy tính ở trình độ Cao đẳng. Tài liệu dùng làm giáo trình học tập cho sinh viên trong các khóa đào tạo và cũng có thể được sử dụng cho đào tạo ngắn hạn hoặc công nhân kỹ thuật đồng thời có thể làm tài liệu tham khảo cho các lập trình viên.

Tam Điệp, ngày.....tháng.... năm.....

Tham gia biên soạn

1. Phạm Thị Thoa
2. Nguyễn Xuân Khôi
3. Nguyễn Anh Văn

MỤC LỤC

LỜI GIỚI THIỆU.....	4
- Về năng lực tự chủ và trách nhiệm.....	7
Bài 1: Xử lý ngoại lệ.....	9
Mã bài: MD16 B01.....	9
Giới thiệu:.....	9
Bài học này bao gồm hai nội dung: Phát sinh và bắt giữ ngoại lệ và những lớp đối tượng ngoại lệ.....	9
Mục tiêu:.....	9
1. Phát sinh và bắt giữ ngoại lệ.....	9
1.1. Câu lệnh catch.....	9
2. Những đối tượng ngoại lệ.....	13
Bài 2: Lập trình Visual Studio.Net.....	15
1. Lập trình với Visual Studio.Net.....	15
2. Thêm và điều chỉnh các điều khiển.....	19
Bài 3: Tạo thực đơn.....	28
1. Tạo thực đơn chính.....	28
2. Tạo thực đơn ngữ cảnh.....	29
1. Tạo Project Menu chứa form sau.....	32
2. Tạo Project menu2 chứa Form có menu.....	32
.....	32
3. Tạo menu ngữ cảnh	32
.....	32
Bài 4: Tạo thanh trạng thái.....	33
1. Tạo thanh trạng thái.....	33
Bài tập.....	35
.....	35
Bài 5: Tạo hộp thoại.....	36
1. Tạo hộp thoại thông điệp.....	36
2. Sử dụng phương thức Form.Close.....	38
3. Tạo hộp thoại Modal.....	39
4. Tạo hộp thoại Modales.....	40
Bài tập.....	40
Tạo form khi nhấn vào mỗi nút sẽ hiển thị ra hộp thoại tương ứng.....	40
.....	41
Bài 6: Các hộp thoại tập tin.....	42
6.1. Tập tin và đường dẫn.....	42
Khái niệm về các loại tập tin.....	42
Xử lý hệ thống tập tin và thư mục.....	43
Xử lý đọc ghi tập tin.....	51
6.2. Tạo hộp thoại lưu tập tin.....	51
6. 3. Tạo hộp thoại mở tập in.....	55
Bài tập.....	59

1. Tạo phần nền Notepad.....	59
.....	59
Khi nhấn vào	59
New tạo ra một file mới.....	59
Bài 7: Sử dụng các đối tượng.....	61
1. Sử dụng TextBox, Button, Label.....	61
2. Sử dụng ListBox.....	68
3. Sử dụng ComboBox.....	74
4. Sử dụng ListView.....	78
ListView.....	78
5. Sử dụng TreeView.....	85
Bài tập.....	94
2. Viết chương trình gồm 1 form trong đó có 1 button và 1 label hiển thị số lần nhấn button đó.....	94
3. Thiết kế giao diện và cài đặt chương trình như sau:.....	94
Bài 8: Một số tính năng khác.....	100
Bài tập.....	103
1. Tạo project quanlydiem và tạo Form frmMain là Form cha chứa các form con.....	103
.....	104
- Khi chọn Quản lý điểm trong thực đơn Quản lý sẽ hiển thị Form con frmdiem.....	104
.....	104
.....	104
Bài 9: Truy cập và xử lý cơ sở dữ liệu.....	105
1. Kết nối dữ liệu.....	105
2. Thao tác với dữ liệu: Thêm, sửa, xóa, lọc dữ liệu.....	112
tblSinhVien(MaSV,Hoten,Ngaysinh, Noisinh, Gioitinh Diachi, Makhoa).	118
tblKhoa(Makhoa, Tenkhoa)	118
tblMonHoc(Mamon, Tenmon, Makhoa, Sohocphan, Giaovien)	118
tblDiem(Mamon, MaSV, Diem)	118
Bài 10: Lập báo cáo với CrystalReport.....	121
1. Lập báo cáo bằng Winzard.....	121
2. Lập báo cáo không sử dụng Winzard.....	127
Madg.....	131
Manv.....	132
Maloai.....	132
Masach.....	132
Manv.....	133
Madg.....	133

MÔ ĐUN: LẬP TRÌNH C#2

Mã mô đun: MĐ 16

Vị trí, tính chất, ý nghĩa và vai trò của mô đun:

Vị trí: Học sau mô đun Lập trình C# 1.

Tính chất: Là mô đun chuyên môn ngành lập trình máy tính.

Ý nghĩa và vai trò: Đây là mô đun cung cấp các kiến thức chuyên sâu về lập trình Window Form.

Mục tiêu của mô đun:

- Kiến thức:

+ Trình bày được các lỗi phát sinh ngoại lệ và tầm ảnh hưởng của nó đến chương trình;

+ Trình bày được khái quát các thành phần của Visual Studio.NET;

+ Mô tả được các bước tạo thực đơn chính và thực đơn ngữ cảnh;

+ Trình bày được chức năng, các thuộc tính và trình tự sử dụng của các đối tượng để tạo Form;

+ Trình bày được các bước kết nối, truy cập và xử lý dữ liệu;

+ Mô tả được các bước lập báo cáo;

- Kỹ năng

+ Tạo các ứng dụng trên nền Windows.

+ Tạo được các ứng dụng MDI.

+ Lập trình và sử dụng được các đối tượng của .NET.

+ Tạo được ứng dụng cơ sở dữ liệu với các chức năng thêm, sửa, xóa, thông kê, tìm kiếm thông tin.

- Về năng lực tự chủ và trách nhiệm

Thực hiện được các biện pháp an toàn cho người và máy tính

Nội dung của mô đun:

Số TT	Tên các bài trong mô đun	Thời gian			
		Tổng g số	Lý thuyết t	Thực hành	Kiểm tra
1	Bài 1: Xử lý ngoại lệ	5	2	3	
2	Bài 2: Lập trình với Visual Studio .NET	5	3	2	
3	Bài 3: Tạo thực đơn	8	2	6	1
4	Bài 4: Tạo thanh trạng thái	7	2	5	
5	Bài 5: Tạo các hộp thoại	10	2	8	
6	Bài 6: Tạo các hộp thoại tập tin	15	4	9	1
7	Bài 7: Sử dụng các đối tượng	30	9	20	1
8	Bài 8: Tạo ứng dụng MDI	5	1	4	
9	Bài 9: Truy cập và xử lý cơ sở dữ liệu	60	12	58	2

10	Bài 10: Lập báo cáo với CrystalReport	15	3	10	1
	Cộng	160	40	114	6

TaiLieu.vn

Bài 1: Xử lý ngoại lệ

Mã bài: MD16_B01

Giới thiệu:

Bài học này bao gồm hai nội dung: Phát sinh và bắt giữ ngoại lệ và những lớp đối tượng ngoại lệ.

Mục tiêu:

Trình bày được các lỗi phát sinh ngoại lệ và tầm ảnh hưởng của nó đến chương trình.

Sử dụng được các lớp đối tượng ngoại lệ.

Xây dựng được phần xử lý ngoại lệ trong chương trình.

Thực hiện được các biện pháp an toàn cho máy tính

1. Phát sinh và bắt giữ ngoại lệ

Một Exception (ngoại lệ) là một vấn đề xuất hiện trong khi thực thi một chương trình. Một Exception trong C# là một phản hồi về một tình huống ngoại lệ mà xuất hiện trong khi một chương trình đang chạy, ví dụ như chia cho số 0.

Exception cung cấp một cách để truyền điều khiển từ một phần của một chương trình tới phần khác. Exception Handling (Xử lý ngoại lệ) trong C# được xây dựng dựa trên 4 từ khóa là: **try**, **catch**, **finally**, và **throw**.

- ❖ **try**: Một try xác định một khối mã mà trường hợp ngoại lệ đặc biệt được kích hoạt. Nó được theo sau bởi một hoặc nhiều hơn các khối catch.
- ❖ **catch**: Một chương trình bắt một ngoại lệ với một ngoại lệ được xử lý tại chỗ. Từ khóa catch chỉ bắt một ngoại lệ.
- ❖ **finally**: Khóa finally được sử dụng để thực hiện một tập hợp các báo cáo, cho dù một ngoại lệ được ném hoặc không ném. Ví dụ, nếu bạn mở một tập tin, nó phải được đóng cho dù một ngoại lệ xảy ra hay không.
- ❖ **throw**: Một chương trình ném một ngoại lệ khi một vấn đề xuất hiện. Điều này được thực hiện bằng cách sử dụng từ khóa throw.

1.1. Câu lệnh catch

◆ Dùng Try-catch để xử lý ngoại lệ

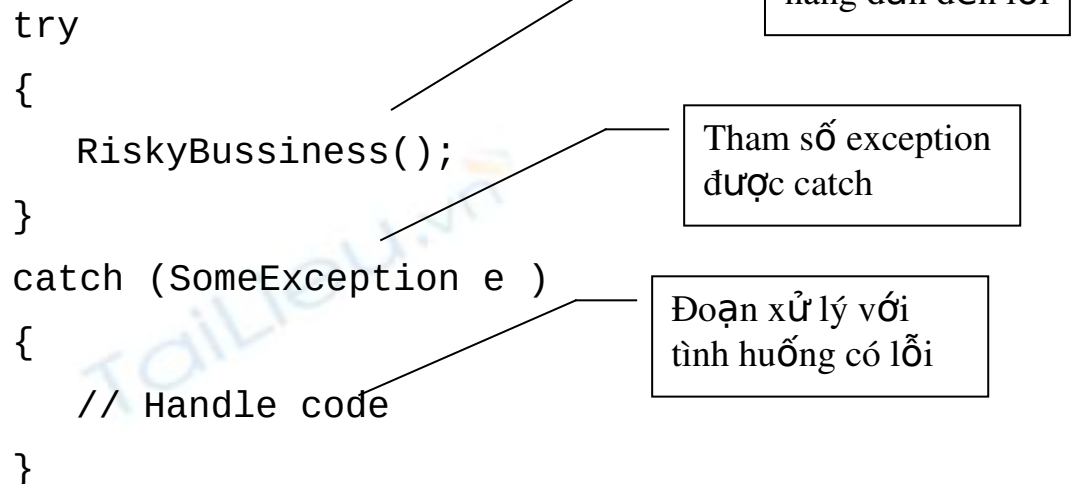
- ❖ Đặt code có khả năng dẫn đến ngoại lệ vào khối “try”

❖ Cung cấp các khối “catch” theo sau “try”

❖ Có thể cung cấp tất cả catch cho các lỗi nếu muốn xử lý, bằng cách sử dụng các lớp exception thích hợp

❖ Nếu không cung cấp catch cho một ngoại lệ, thì exception này được lan truyền lên trên.

Cú pháp:



Khối try:

◆ Bao gồm các phần

❖ Từ khóa try

❖ Theo sau khối “{ ... }”

❖ Khối “{ ... }” bắt buộc phải có, khác với “{ ... }” trong if hay for

◆ Bên trong khối try

❖ Đặt bất cứ câu lệnh nào có khả năng phát sinh ra ngoại lệ

Khối catch:

◆ Đặt một hay nhiều ngay sau khối try

❖ Không có lệnh nào chen giữa hai khối catch của một try

◆ Cú pháp khối catch như sau

```
catch (Exception-class [var1])
{
    // xử lý ngoại lệ 1
}
```

```

    }
    catch (Exception-class [var2])
    {
        // xử lý ngoại lệ 2
    }

```

Ví dụ:

```

namespace NguyenHaGiang
{
    class Program
    {
        static void Main(string[] args)
        {
            int x=0;
            int div=0;
            try
            {
                div = 100 / x;
                Console.WriteLine("This line is not executed!");
            }
            catch (DivideByZeroException e)
            {
                Console.WriteLine("Exception occurred");
            }
            Console.WriteLine("Result is {0}",div);
        }
    }
}

```

1.2. Câu lệnh Finally

◆ Khi một exception được ném ra

- ❖ Luồng thực thi sẽ nhảy vào khối catch xử lý nó.
- ❖ Một số đoạn code giải phóng tài nguyên có thể bị bỏ qua
 - Open File
 - Read Data // ngoại lệ được phát sinh
 - Close File // đoạn code này bị bỏ qua, dù file chưa đóng
- ❖ Khối try-catch có phần option là finally
 - ❖ Luôn luôn được gọi
 - ❖ Sử dụng để dọn dẹp các tài nguyên đang nắm giữ

```

namespace NguyenHaGiang
{
    class Program
    {
        static void Main(string[] args)
        {
            int x=0;
            int div=0;
            try
            {
                div = 100 / x;
                Console.Write("This line is not executed!");
            }
            catch (DivideByZeroException e)
            {
                Console.WriteLine("Exception occurred");
            }
            finally // alway executed
            {
                Console.WriteLine("Finally block!");
            }
            Console.WriteLine("Result is {0}",div);
        }
    }
}

```

1.3. Câu lệnh throw

Phát sinh một ngoại lệ: sử dụng từ khóa throw

throw new System.Exception();

Hoạt động: Khi phát sinh ngoại lệ thì ngay tức khắc sẽ làm ngừng việc thực thi trong khi CLR sẽ tìm kiếm một trình xử lý ngoại lệ. Nếu một trình xử lý ngoại lệ không được tìm thấy trong phương thức hiện thời, thì CLR tiếp tục tìm trong phương thức gọi cho đến khi nào tìm thấy. Nếu CLR trả về lớp Main() mà không tìm thấy bất cứ trình xử lý ngoại lệ nào, thì nó sẽ kết thúc chương trình.

Ví dụ:

```

namespace NguyenHaGiang
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                throw new DivideByZeroException("Invalid Division");
            }
            catch (DivideByZeroException e)
            {
                Console.WriteLine("Exception!");
            }
            Console.WriteLine("Last statement");
        }
    }
}

```

2. Những đối tượng ngoại lệ

Lớp Exception

- ◆ Có 2 loại ngoại lệ
 - ❖ Ngoại lệ phát sinh bởi chương trình
 - ❖ Ngoại lệ được tạo bởi CLR
- ◆ Lớp System.Exception là cơ sở cho tất cả lớp trong C#
- ◆ 2 lớp kế thừa từ lớp này:
 - ❖ ApplicationException: thường làm lớp cơ bản cho lớp ngoại lệ phát sinh từ ứng dụng
 - ❖ SystemException: do CLR phát sinh

Một số lớp Exception thường dùng

Lớp ngoại lệ	Mô tả
System.IO.IOException	Xử lý lỗi I/O
System.IndexOutOfRangeException	Xử lý các lỗi phát sinh khi một phương pháp để cập đến một chỉ số mảng nằm ngoài phạm vi.
System.ArrayTypeMismatchException	Xử lý các lỗi phát sinh khi loại chưa phù hợp với các kiểu mảng.
System.NullReferenceException	Xử lý các lỗi phát sinh từ một đối tượng null.
System.DivideByZeroException	Xử lý các lỗi phát sinh từ việc chia cho số không.

System.InvalidCastException	Xử lý các lỗi phát sinh trong quá trình phân loại.
System.OutOfMemoryException	Xử lý các lỗi được tạo ra từ bộ nhớ không đủ.
System.StackOverflowException	Xử lý các lỗi phát sinh từ tràn stack.

Bài tập

TaiLieu.vn

Bài 2: Lập trình Visual Studio.Net

Mã bài: MD16_B02

Giới thiệu:

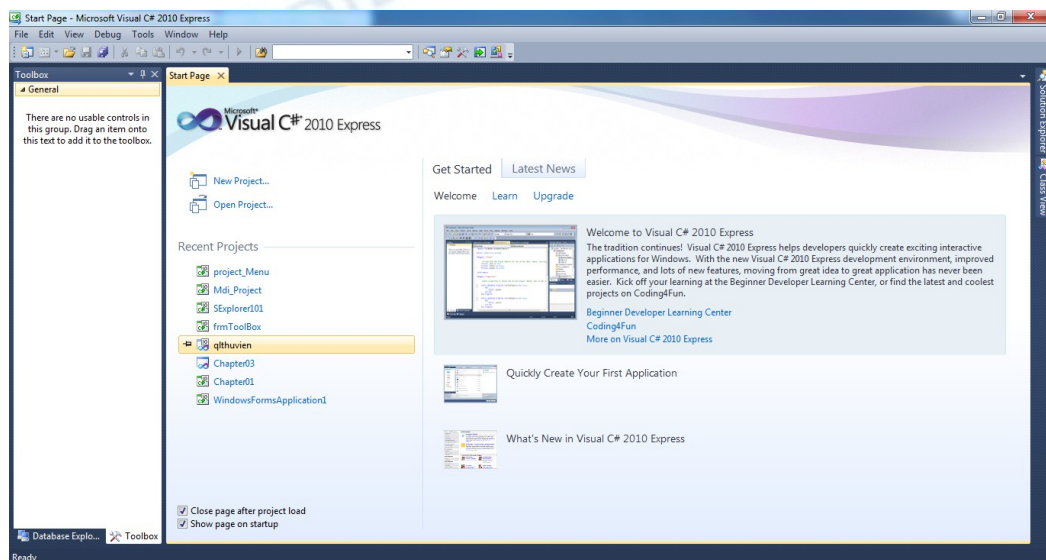
Trong bài học này chúng ta sẽ làm quen với giao diện tạo ứng dụng Windows Form Application trong Visual Studio C# 2010 Express.

Mục tiêu:

Trình bày được khái quát các thành phần của Visual Studio.NET.
Tạo được Project, tạo và sử dụng được form đơn giản
Thực hiện được các biện pháp an toàn cho máy tính

1. Lập trình với Visual Studio.Net

- Giao diện của Visual Studio C# 2010 Express



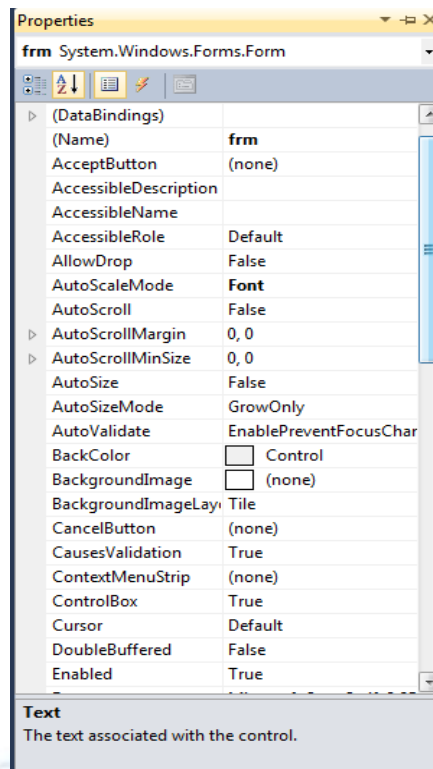
+New Project: Tạo Project mới

+ Open Project: Mở Project đã tạo

+ Recent Projects: Các Project mới tạo gần đây nhất

- Form

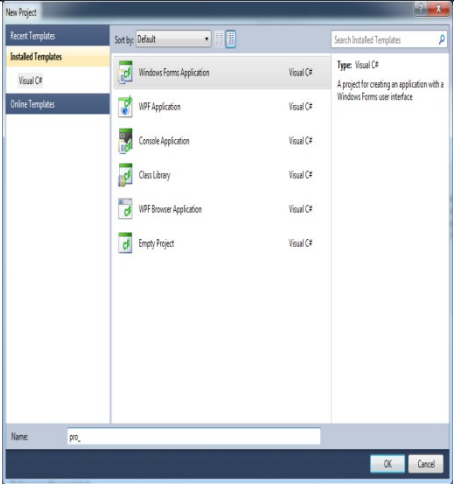
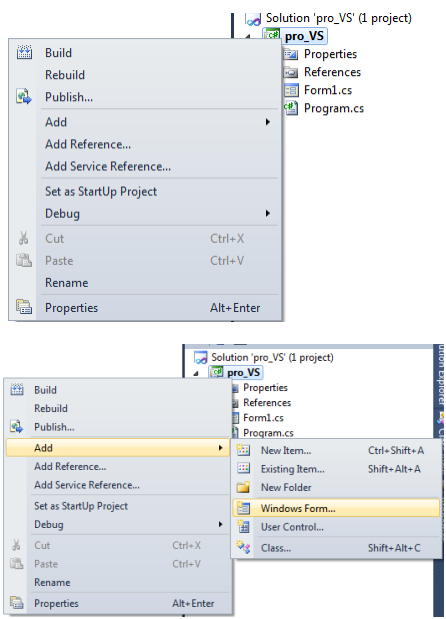
+ Các thuộc tính của Form: Chứa trong cửa sổ Properties

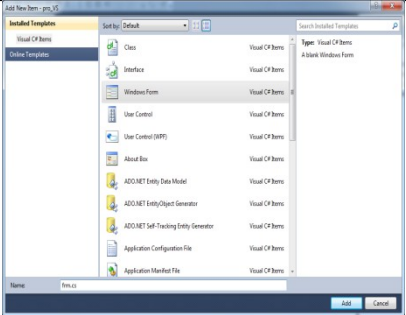
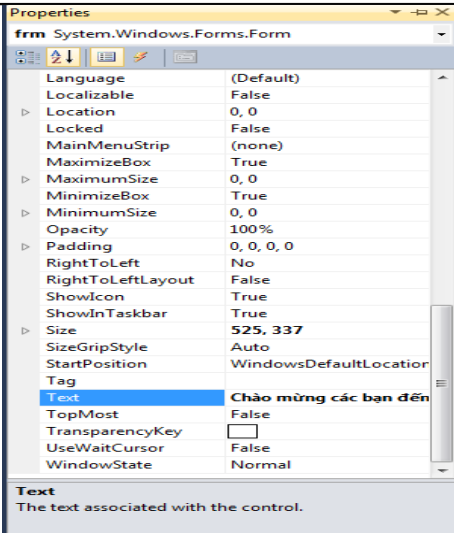
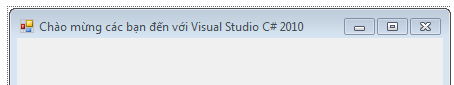
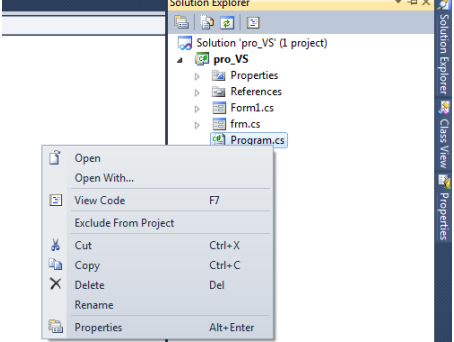
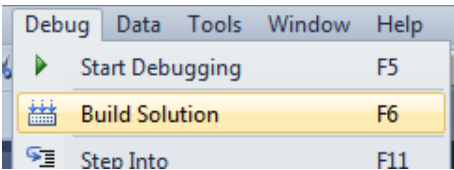


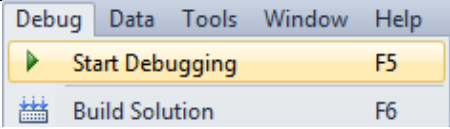
Một số thuộc tính thường dùng

Thuộc tính	Chức năng
(name)	Tên Form. Nó như một biến vậy. Quan trọng
AllowDrop	Cho phép kéo thả
AutoSizeMode	+GrowOnly: cho phép thay đổi kích thước form +GrowAndShrink: không cho phép thay đổi kích thước form
BackColor	Màu nền Form
ContextMenuStrip	Menu xuất hiện khi chuột phải lên Form
Cursor	Con trỏ chuột khi rê trên form
KeyPreview	Cho phép nhận phím hay không (cái này kết hợp với sự kiện nhận phím)
Opacity	Độ trong suốt của form
ShowInTaskbar	Hiện thị ở thanh taskbar khi chạy form (true/false)
StartPosition	Vị trí xuất hiện của form +CenterScreen: Giữa màn hình +CenterParent: Giữa Form gọi ra nó (form cha)
TopMost	Hiện thị đè lên các form khác
WindowState	Trạng thái của form
Text	Chữ hiển thị khi form chạy

Trình tự thực hiện

TT	Nội dung	Yêu cầu kỹ thuật	Hình ảnh
Bước 1	Tạo mới Project + Khởi động Visual Studio C# 2010 Express + Chọn New Project + Chọn Windows Forms Application + Gõ tên dự án vào mục Name + Nhấn OK	Tạo Project với tên chính xác	
Bước 2	Thêm Form mới vào Project + Chọn cửa sổ Reluion Explorer + Chuột phải vào tên Dự án + Chọn Add + Chọn Windows Form + Gõ tên Form trong mục Name + Chọn ADD		

			
Bước 3	Tùy chỉnh Form - Chọn Form - Chọn cửa sổ Properties Thay đổi thuộc tính Text: Chào mừng các bạn đến với Visual Studio C# 2010	Thiết lập chính xác thuộc tính	 
Bước 4	Chạy Form - Chọn cửa sổ Solution Explorer - Chuột phải vào lớp Program.cs - Chọn View Code (F7) - Thay thế tên form cần chạy vào mục Application.Run(new Ten_form()); - Biên dịch Debug\ Build Solution (Nhấn F6)		 <pre> namespace pro_VS { static class Program { /// <summary> /// The main entry point for the application. /// </summary> [STAThread] static void Main() { Application.EnableVisualStyles(); Application.SetCompatibleTextRenderingDefault(false); Application.Run(new frm()); } } } </pre> 

	- Chạy Form Debug\ Start Debugging (Nhấn F5)		
--	---	--	---

Sai hỏng thường gặp, nguyên nhân và biện pháp phòng tránh/ khắc phục

- Sai hỏng thường gặp

Không ra ứng dụng Window Form

Nguyên nhân

Ở bước 1, chọn sai loại Project

Biện pháp khắc phục

Làm lại từ bước 1:

+ Chọn New Project

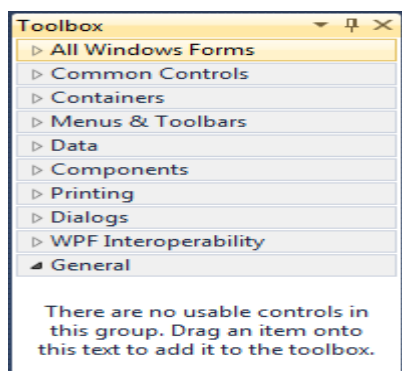
+ Chọn Windows Forms Application

+ Gõ tên dự án vào mục Name

+ Nhấn OK

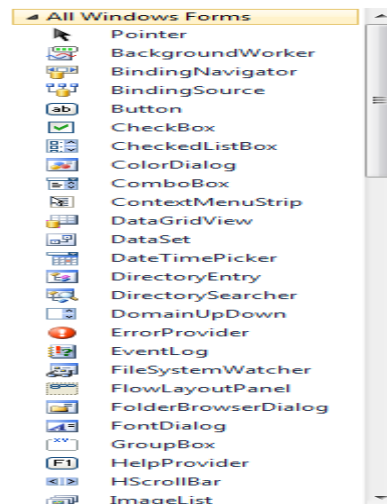
2. Thêm và điều chỉnh các điều khiển

* Các điều khiển

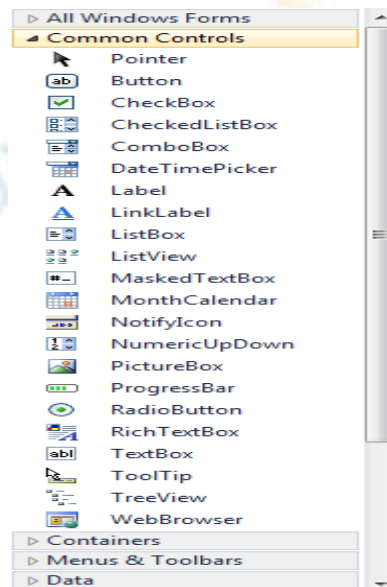


Toolbox bao gồm các nhóm điều khiển

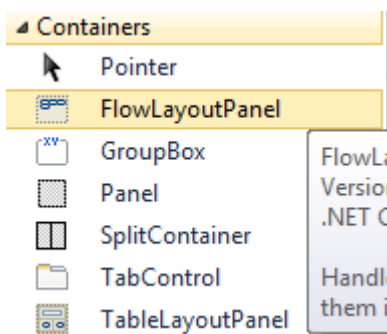
- All Windows Form: bao gồm tất cả các điều khiển dùng để tạo windows form



- Common Controls: Các điều khiển thông thường



- Container bao gồm các điều khiển có thể chứa các điều khiển khác



- Menu & Toolbars: Các điều khiển dùng để tạo thanh thực đơn và thanh trạng thái.