

### BÀI 3. LẬP TRÌNH WINDOWS FORM CƠ BẢN

#### A. MỤC TIÊU:

○ Thiết kế được giao diện Windows Form từ các control trong C#; ○

Sử dụng được các Controls Textbox, Label, Button, Form.

#### B. NỘI DUNG THỰC HÀNH

##### 1. Cơ sở lý thuyết

##### 1.1. Kiến thức cần nhớ

Các thuộc tính và các sự kiện cơ bản của các Controls Textbox, Label, Button, Form.

**Các thuộc tính chung:** Text, TextAlign, TabIndex, Enable, Visible, Anchor, Size, BackColor, BackgroundImage, ForeColor,...

##### ○ Form:

✦ Phương thức: Close, Hide, Show, ShowDialog

✦ Sự kiện: Load, FormClosing, FormClosed ○ **TextBox:**

✦ Phương thức: AppendText, Clear, Select... ✦ Sự kiện:

TextChanged, KeyDown, KeyPress, KeyUp ○ **Button:**

✦ Sự kiện: Click

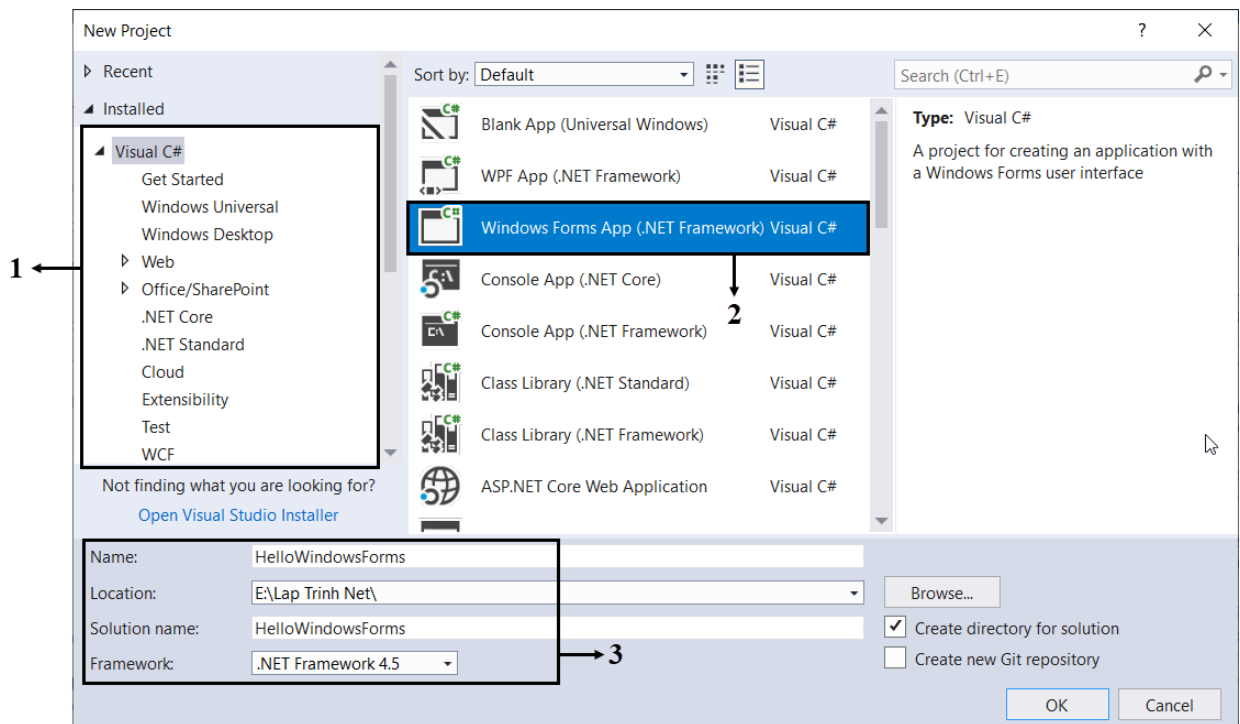
✦ Mỗi Solution hoặc Project có các loại tập tin sau:

File Name	Diễn giải
MyName.sln	Đây là file solution (file text) lưu trữ thông tin về solution và project. Nếu project đã có sẵn thì ta sẽ mở file này để tiếp tục project
frmMyName.cs	Lưu trữ code trong form
frmMyName.Designer.cs	Lưu trữ code design trong form
frmMyName.rex	Đây là file resource, định nghĩa tất cả system resources sử dụng trong form
MyName.csproj	File project này mô tả thông tin project và danh sách các file có trong project
Program.cs	Là tập tin tự phát sinh khi tạo project và cũng là tập tin bắt đầu một ứng dụng
Các files khác	Sau khi thực thi một project thì một số file khác sẽ tự động phát sinh trong thư mục solution

## Tạo một ứng dụng Windows Form

Khởi động chương trình: Microsoft Visual Studio 2017.

Tạo một project mới: File ➤ New ➤ Project:



**Hình 3. 1.** Giao diện tạo mới project Windows Forms

Trong đó:

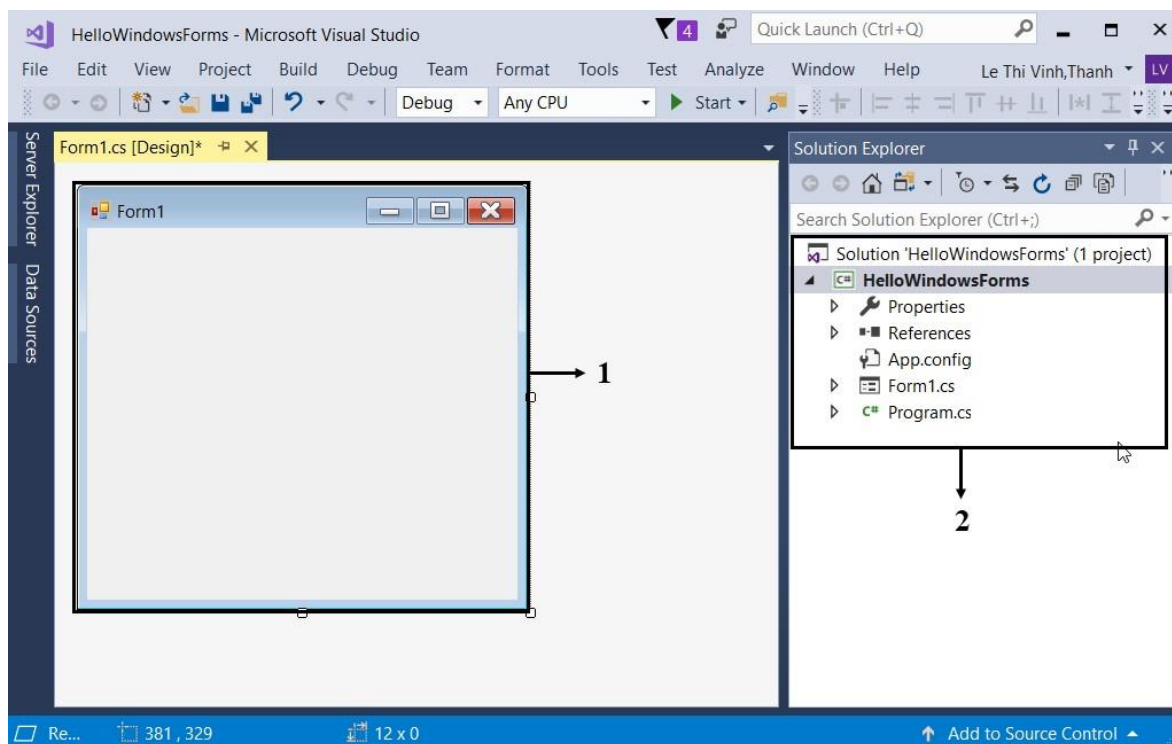
[1]: Ngôn ngữ viết code là C#

[2]: Chọn loại; Windows Form App (.Net Framework)

[3]:

- Name: Tên project đầu tiên
- Location: Đường dẫn lưu tập tin
- Solution name: Tên thư mục chứa project vừa tạo

Sau khi chọn ngôn ngữ, ứng dụng, đặt tên cho project, ... click OK, cửa sổ thiết kế giao diện (Form Design) sẽ hiện ra:

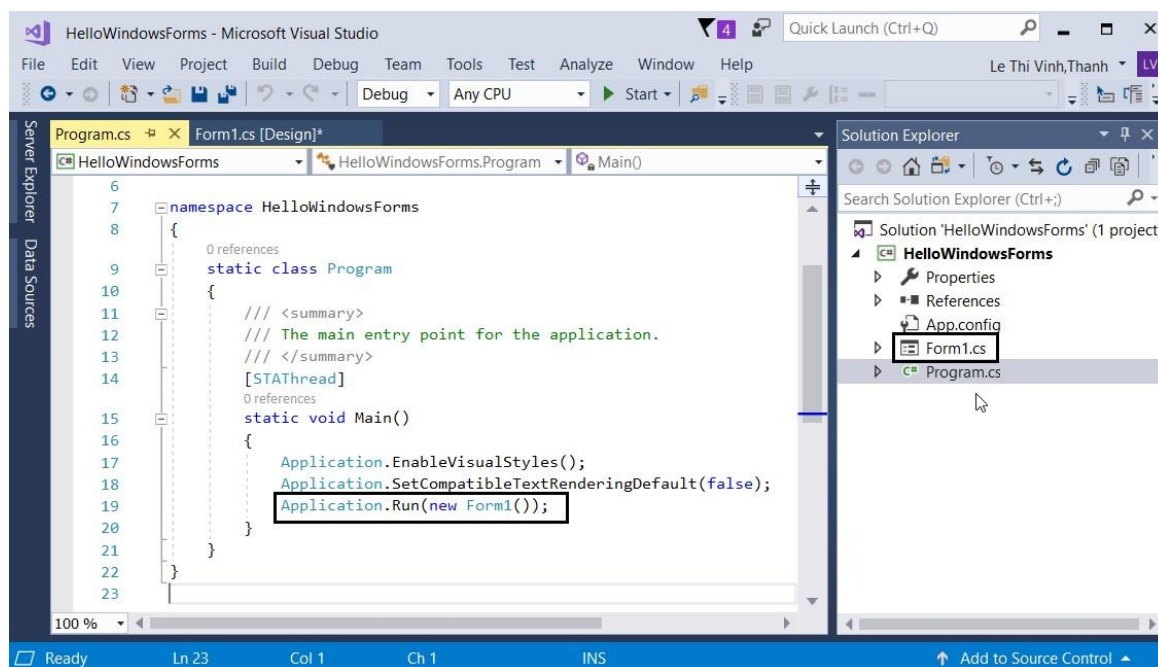


**Hình 3. 2.** Giao diện hiển thị thiết kế Windows Forms

[1]: Giao diện Form1.cs [Design] hiển thị lên màn hình

[2]: Solution chứa các project nhỏ

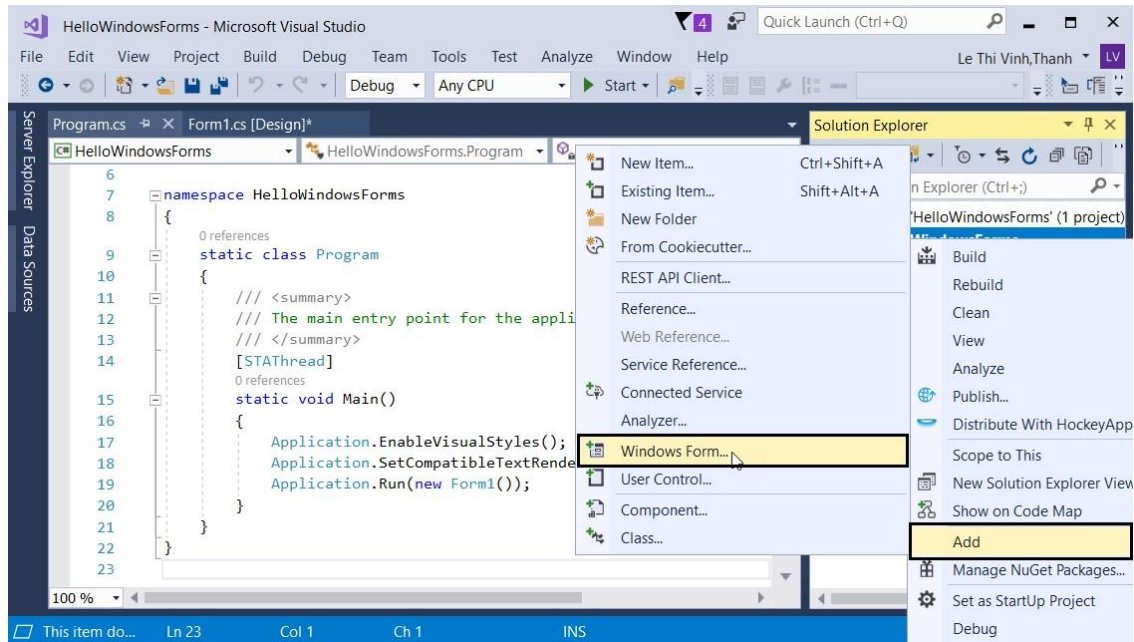
Lưu ý: Trong cùng một project có thể có nhiều form, muốn chọn form khởi động đầu tiên ta mở file Program.cs và thay tên form cần mở trong lệnh `Application.Run(new Ten_form());`;



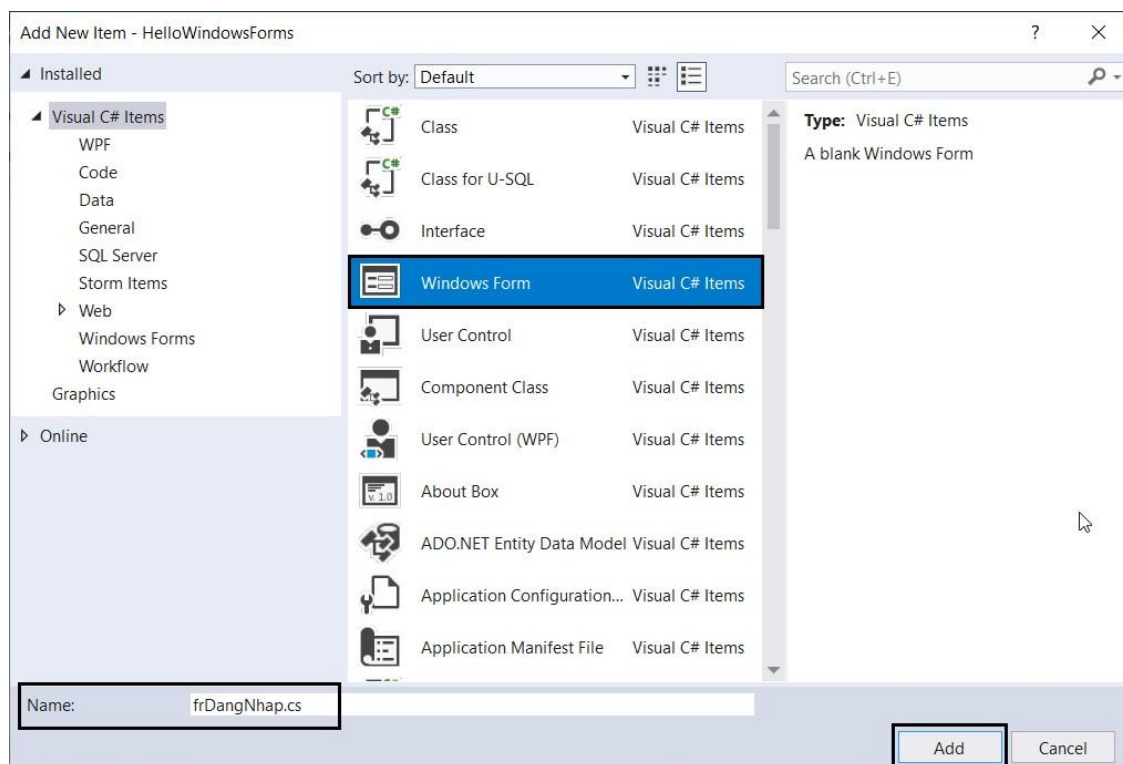
**Hình 3. 3.** Giao diện chạy chương trình Windows Forms

Trước khi làm bài tập thì nên xóa Form1 (form được tạo mặc định), tạo Form mới và đặt tên theo yêu cầu. Cách thực hiện như sau:

Trong cửa sổ Solution Explorer, click chuột phải vào Form1 **7** Delete. Sau đó Click chuột phải trên vào tên project: HelloWorldForms **7** Add **7** Windows Form



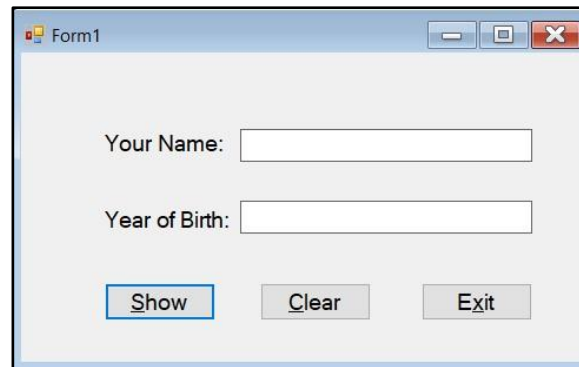
**Hình 3. 4.** Giao diện thêm mới một Form trong project (bước 1) Cửa sổ **Add New Item** xuất hiện, đặt tên cho form mới và chọn **Add**.



**Hình 3. 5.** Giao diện thêm mới một Form trong project (bước 2)

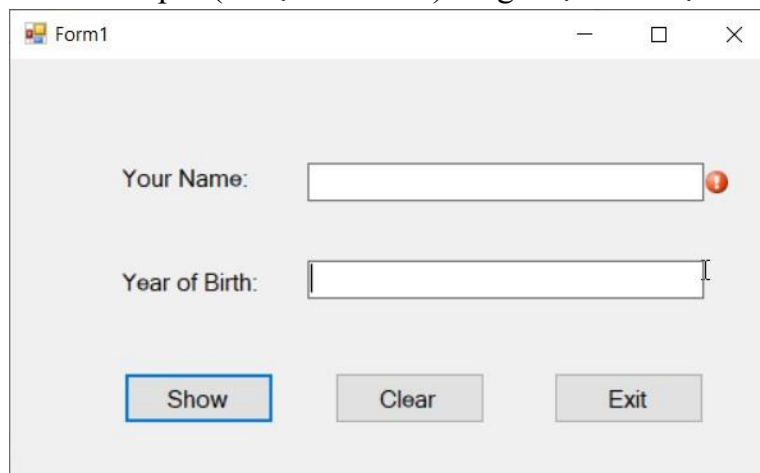
## 1.2. Giới thiệu bài tập mẫu

Thiết kế form sau:



**Yêu cầu:**

- Khi Form khởi động thực hiện nhập tên, năm sinh vào TextBox YourName và Year of Birth tương ứng. Nếu YourName không được nhập dữ liệu, năm sinh không phải là số thì thông báo lỗi (dùng **ErrorProvider**).
- **Errorpovider**: Cung cấp thông tin lỗi của điều khiển trên Form, thường được dùng với control input (ví dụ: TextBox) ràng buộc với một điều kiện nhập.



- Khi người dùng nhấn button Show sẽ hiển thị thông tin nhập vào MessageBox bao gồm: tên, tuổi (năm hiện tại – năm sinh).




Khi người dùng nhấn button Clear sẽ xóa thông tin đã nhập trên các TextBox, đặt con trỏ văn bản vào Textbox YourName.

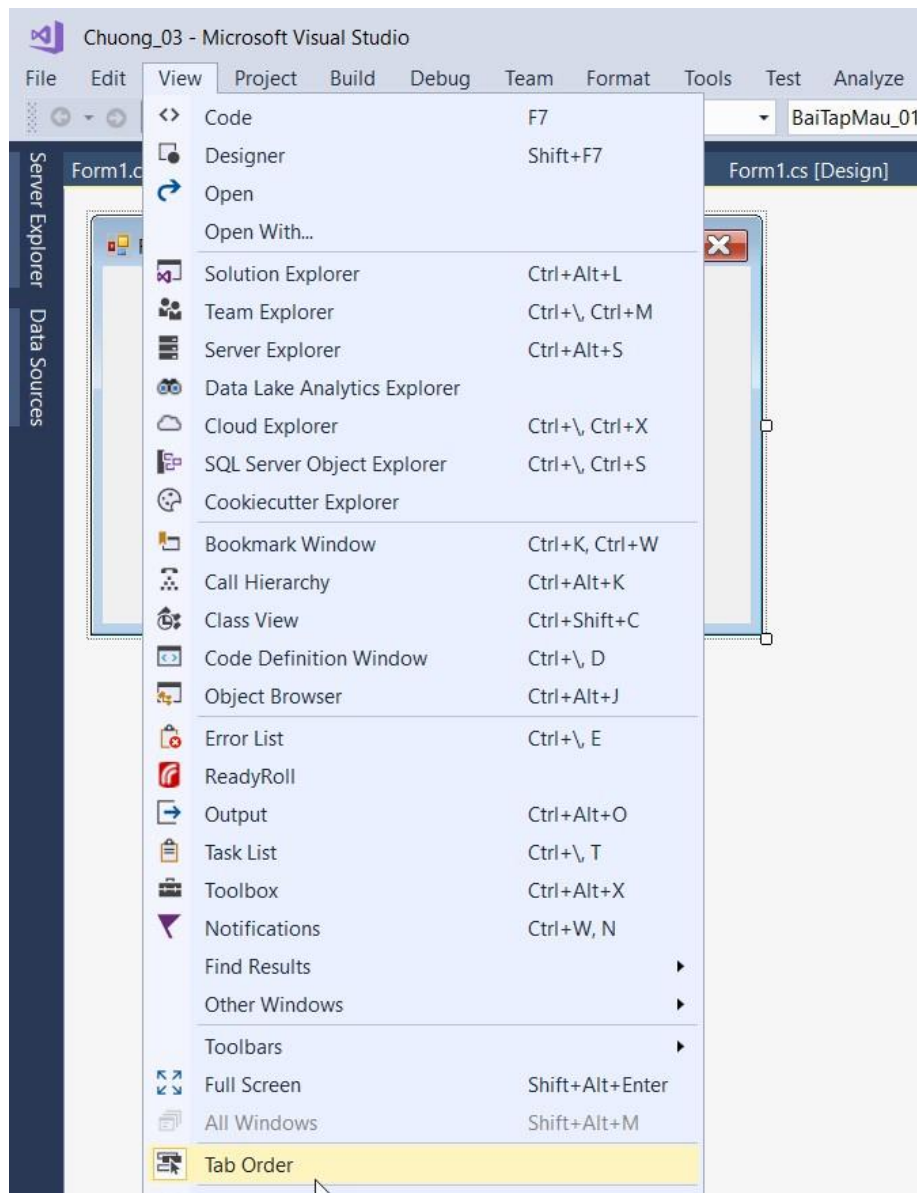
- Khi nhấn button Exit xác nhận người dùng có thực sự muốn thoát khỏi chương trình không? Yes: thoát, No: không

### Hướng dẫn:

Danh sách các thuộc tính của các đối tượng (Object) cần thiết:

Object	Properties	Events
frmBaiTap1	Name: frmBaiTap1 Text: My name Project FontName: Tahoma FontSize: 11 AcceptButton: btnShow (nhận sự kiện Click chuột khi nhấn Enter) CancelButton: btnExit (nhận sự kiện Click chuột khi nhấn Esc)	FormClosing
txtYourName	Name: txtYourName BorderStyle: FixSingle	Leave (mất tiêu điểm)
txtYear	Name: txtYear BorderStyle: FixSingle	TextChanged
btnShow	Name: btnShow Text: &Show	Click
btnClear	Name: btnClear Text: &Clear	Click
btnExit	Name: btnExit Text: E&xit	Click
errorProvider	Name: errorProvider1	

Thay đổi thứ tự nhận tiêu điểm trên form chọn menu View  Tab order:



Lần lượt thực hiện click chọn từng phần tử trên Form theo thứ tự nhận tiêu điểm:

Viết các sự kiện tương ứng với các yêu cầu:

Sự kiện đóng xác nhận khi form được đóng:

```
private void frmBaiTap1_FormClosing(object sender,
FormClosingEventArgs e)
{
    DialogResult r;
    r = MessageBox.Show("Bạn có muốn thoát?", "Thoát",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question,
        MessageBoxDefaultButton.Button1);    if (r == DialogResult.No)
e.Cancel = true; }

```

Kiểm tra TextBox Your Name đã được nhập nội dung chưa:

```
private void txtYourName_Leave(object sender, EventArgs e) {
    Control ctr = (Control)sender;
    if (ctr.Text.Trim().Length == 0)
        this.errorProvider1.SetError(ctr, "You must enter Your Name");
    else this.errorProvider1.Clear();
}

```

Yêu cầu TextBox Year phải nhập số, nhập sai dùng ErrorProvider thông báo lỗi:

```
private void txtYear_TextChanged(object sender, EventArgs e)
{
    Control ctr = (Control)sender;    if (ctr.Text.Length > 0 &&
!Char.IsDigit(ctr.Text[ctr.Text.Length
- 1])) this.errorProvider1.SetError(ctr, "This is not a valid
number");    else this.errorProvider1.Clear();
}

```

Hiển thị thông tin theo yêu cầu khi click vào Button Show:

```
private void btnShow_Click(object sender, EventArgs e)
{
    int age; string s;
    s = "My name is: " + txtYourName.Text + "\n";    age =
DateTime.Now.Year - Convert.ToInt32(txtYear.Text);    s = s +
"Age: " + age.ToString();
    MessageBox.Show(s);
}

```

Button Clear thực hiện lệnh xóa các TextBox chuẩn bị cho lần nhập tiếp theo:

```
private void btnClear_Click(object sender, EventArgs e) {
    //clear Text
    txtYourName.Clear();

    txtYear.Clear();
    //Textbox YourName reset focus    txtYourName.Focus();
}

```

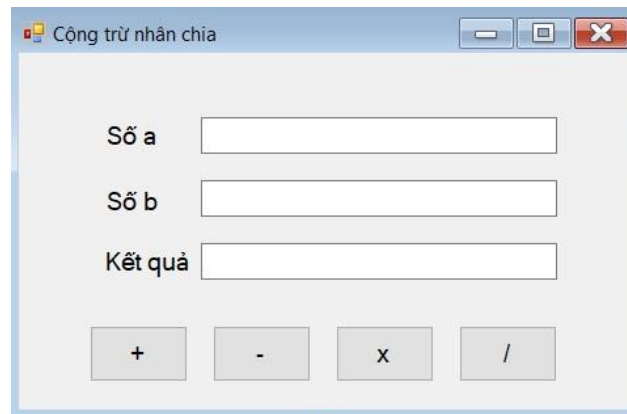


Thoát chương trình khi click vào Button Exit:

```
private void btnExit_Click(object sender, EventArgs e)
{
    this.Close();
}
```

## 2. Bài tập tại lớp

**Bài tập 1.** Thiết kế form thực hiện phép tính +, -, x, / của hai số a, b với giao diện như sau:



**Yêu cầu:** Khi chọn các phép toán +, -, \*, / tương ứng trên các button thì hiển thị kết quả lên textbox kết quả.

- Kiểm tra tính hợp lệ của các textbox theo hai mức độ:
  - Mức 1: Nếu dữ liệu nhập vào textbox a, b không hợp lệ thì dùng ErrorProvider thông báo lỗi.
  - Mức 2: Chặn không cho người dùng nhập giá trị khác số vào textbox a, b.
- Thông báo lỗi bằng MessageBox nếu dữ liệu nhập không phù hợp. Hỏi xác nhận trước khi đóng form

### Hướng dẫn:

Bước 1: Tạo một project mới và xây dựng các controls theo bảng sau:

Control	Thuộc tính	Sự kiện
TextBox	Name: txtNumberA	TextChanged
TextBox	Name: txtNumberB	TextChanged
TextBox	Name: txtResult	
Button	Name: btnAdd	Click
Button	Name: btnSubtract	Click
Button	Name: btnMultiply	Click
Button	Name: btnDivide	Click
errorProvider	Name: errorProvider1	

Bước 2: Xây dựng các sự kiện cho các Controls.

Kiểm tra text có phải là số hay không trong hai TextBox: số A, số B:

```
if (txtNumberA.Text.Length > 0 &&  
    !Char.IsDigit(txtNumberA.Text[txtNumberA.Text.Length - 1]))  
this.errorProvider1.SetError(txtNumberA, "This is not a valid  
number"); else  
    this.errorProvider1.Clear();
```

Ở trên kiểm tra cho textBox NumberA, NumberB tương tự.

Đầu tiên kiểm tra dữ liệu đầu vào:

```
private bool isNumber(string str)  
{  
    for (int i = 0; i < str.Length;  
i++)  
        {  
            if  
(!Char.IsDigit(str[i]))  
            {  
                return false;  
            }  
        }  
    return true;  
}
```

Xây dựng các sự kiện Click cho các Button Add:

```
private void btnAdd_Click(object sender, EventArgs e)  
{  
    //Kiểm tra xem số A có phải là số không?  
    if (!isNumber(txtNumberA.Text) || txtNumberA.Text.Equals(""))  
    {  
        MessageBox.Show("This is not a valid number", "Error",  
            MessageBoxButtons.OK,  
            MessageBoxIcon.Error);  
        txtNumberA.Focus();  
    }  
    else if (!isNumber(txtNumberB.Text) || txtNumberB.Text.Equals(""))  
    {  
        MessageBox.Show("This is not a valid number", "Error",  
            MessageBoxButtons.OK,  
            MessageBoxIcon.Error);  
        txtNumberB.Focus();  
    }  
    else { double numberA =  
double.Parse(txtNumberA.Text); double numberB =  
double.Parse(txtNumberB.Text); double sum = numberA + numberB;  
txtResult.Text = sum.ToString();  
    }  
}
```

Các sự kiện cho các nút khác tương tự, tuy nhiên trong phép chia thêm một dòng *if else* nếu người dùng nhập số 0 sẽ thông báo lỗi:

```
else if (txtNumberB.Text.Equals("0"))
{
    MessageBox.Show("Number B does not contain zero", "Error",
        MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    txtNumberB.Focus();
}
```

Xử lý đóng cửa sổ đã trình bày bài mẫu.

## Bài tập 2. Thiết kế form đăng ký tài khoản:

### Yêu cầu:

- Kiểm tra định dạng của email sau khi nhập và ra khỏi textbox Địa chỉ email.
- Bắt buộc nhập dữ liệu trên những textbox có (\*).
- Khi chọn button Đăng ký hoặc nhấn phím Enter sau khi nhập xong dữ liệu vào textbox Xác nhận mật khẩu thì hiển thị thông tin của tất cả các textbox lên MessageBox.
- Hỏi xác nhận trước khi đóng Form.

### Hướng dẫn:

Bước 1: Xây dựng thiết kế như hình và tinh chỉnh các thuộc tính cho các Controls như sau:

Control	Thuộc tính	Sự kiện
Label1	ForeColor: MenuHighlight Font: 20, Bold	
TextBox	Name: txtUserName	
TextBox	Name: txtEmail	
TextBox	Name: txtPassword	
TextBox	Name: txtCfPassword	

Button	Name: btnSignUp	Click
Form	Font: size: 11	FormClosing

Bước 2: Xây dựng hàm kiểm tra Email hợp lệ:

```
private bool checkEmail(string email)
{
    //Kiểm tra có chứa @ không?
    if (!email.Contains("@"))
    {
        return false;
    }
    //Nếu không chứa ".com" -> Sai
    if (!email.Contains(".com"))
    {
        return false;
    }
    //Tìm vị trí @ trong chuỗi email    int index1 =
    email.IndexOf("@");    int index2 = email.IndexOf(".com");    //Lấy
    ra tên miền nếu có của email    string domain =
    email.Substring(index1 + 1, index2 - index1 - 1);
    //Tại đây bạn có thể thêm miền nào muốn vào đây
    //Nếu không nằm trong đây trả về false
    if (domain != "gmail"&& domain != "hotmail")
    {
        return false;
    }
    return true;
}
```

Bước 3: Sự kiện Click của Button Đăng ký.

Đầu tiên kiểm tra các ô trong textbox có tồn tại kí tự (có rỗng không) không:

```
if (txtUserName.Text.Equals(""))
{
    MessageBox.Show("Tên đăng nhập không được để trống!!!", "Thông
    báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    txtUserName.Focus();
} else if
(txtEmail.Text.Equals(""))
{
    MessageBox.Show("Email không được để trống!!!", "Thông báo",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
    txtEmail.Focus();
}
```

Hai ô TextBox còn lại làm tương tự như trên. Tiếp tục gọi đến phương thức kiểm tra email hợp lệ nếu sai thì thông báo người dùng:

```
//Nếu Email không hợp lệ (dấu "!") thì thông báo lỗi if
(!checkEmail(txtEmail.Text))
{
    MessageBox.Show("Email vừa nhập không hợp lệ!!!", "Thông báo",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    txtEmail.Focus();
}
```

Tiếp tục kiểm tra mật khẩu và xác nhận mật khẩu có trùng khớp với nhau không:

```
if (txtCfPassword.Text!=txtPassword.Text)
{
    MessageBox.Show("Mật khẩu xác nhận không trùng khớp!!!", "Thông
báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    txtCfPassword.Focus();
}
```

Nếu kiểm tra tất cả đều hợp lệ thực hiện in thông báo như sau:

```
string mess = "Tên đăng nhập: " + txtUserName.Text + "\n";
mess += "Email: " + txtEmail.Text + "\n"; mess +=
"Password: " + txtPassword.Text + "\n";
MessageBox.Show(mess, "Thông báo", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
```

**Bài tập 3.** Thiết kế giao diện thực hiện yêu cầu nhập vào 2 số nguyên dương a và b. Xuất ra ước chung lớn nhất và bội chung nhỏ nhất của 2 số nguyên dương a và b (kiểm tra dữ liệu nhập và xuất các thông báo lỗi nếu có).

**Hướng dẫn:**

Bước 1: Thiết kế form như hình trên và thông số như bảng sau:

Control	Thuộc tính	Sự kiện
Label1	ForeColor: Red Font: 20, Bold	

TextBox	Name: txtNumberA	
TextBox	Name: txtNumberB	
TextBox	Name: txtDivisor Enable: false	
TextBox	Name: txtMultiples Enable: false	
Button	Name: btnComfilm	Click
Button	Name: btnContinue	Click
Button	Name: btnExit	Click
Form	Font: size: 11	FormClosing

Bước 2: Bắt sự kiện không cho người dùng nhập kí tự khác số vào 2 ô TextBox đầu tiên (nhập số a, nhập số b).

Sử dụng sự kiện KeyPress và xử lý như sau:

```
private void txtNumberA_KeyPress(object sender, KeyPressEventArgs e)
{
    //Kiểm tra nếu nó là kí tự và khác phím xóa (Back) trên bàn phím
    thì không hiển thị lên màn hình    if (!Char.IsDigit(e.KeyChar) &&
    (Keys)e.KeyChar!=Keys.Back)
    {
        e.Handled = true;
    }
}

// Ô textbox thứ hai tương tự
```

Bước 3: Tạo các sự kiện Click cho 3 Button lần lượt như sau:

Khi nhấn Thực hiện:

+ Hệ thống kiểm tra 2 ô TextBox chứa 2 số có rỗng không.

+ Xây dựng hàm tìm ước chung lớn nhất. +

Xây dựng hàm tìm bội chung nhỏ nhất:

$$UCLN(a, b) = \frac{a \times b}{BCNN(a, b)} \quad \text{Vì}$$

chúng ta đã có hàm xử lý UCLN nên chỉ cần gọi lại sử dụng:

```
private int BCNN(int numberA, int numberB)
{
    int NCLN = NCLN(numberA,
    numberB);    return numberA *
    numberB / NCLN;
}
```

Thực hiện tính toán và hiển thị kết quả lên 2 ô TextBox tiếp theo:

```
//Vì hai hàm trả về kiểu số nên khi gán xảy ra lỗi --> +"" sẽ giải quyết được
//Hoặc chấm tời ToString() chuyển số thành chuỗi
txtDivisor.Text = UCLN(int.Parse(txtNumberA.Text),
    int.Parse(txtNumberB.Text)).ToString(); txtMultiples.Text
= BCNN(int.Parse(txtNumberA.Text),
    int.Parse(txtNumberB.Text))+"";
```

Khi nhấn Tiếp tục:

- + Thực hiện xóa nội dung trong 4 ô TextBox; +
- Cho con trỏ chuột về ô TextBox đầu tiên.

Khi nhấn Thoát:

- + Hiện MessageBox cảnh báo thoát cho người dùng.

**Bài tập 4.** Thiết kế giao diện sau:

**Yêu cầu:**

- Nhập vào một dãy số nguyên bất kỳ (nhập số nguyên vào textbox Nhập số, nhấn button Nhập), xuất các số vừa nhập ra textbox dãy vừa nhập.
- Bổ sung thêm button Tính tổng, Tổng chẵn, Tổng lẻ các số vừa nhập và xuất ra các textbox tương ứng.
- Tính tổng các số chẵn và tổng các số lẻ.
- Button Tiếp tục trả lại trạng thái ban đầu của form. ● Button Thoát: xác nhận và đóng form.

**Hướng dẫn:**

Bước 1: Xây dựng giao diện như hình trên.

Bước 2: Đặt tên và tạo các sự kiện như bảng sau:

Controls	Thuộc tính	Sự kiện
Label1	Font: size: 20-Bold, color: Red	
TextBox1	Name: txtNumber	KeyPress
TextBox2	Name: txtArrayNumber Enable: false	
TextBox3	Name: txtSumArrayNumber Enable: false	
TextBox4	Name: txtSumEvenNum Enable: false	
TextBox5	Name: txtSumOddNum Enable: false	
Button1	Name btnEnter	Click
Button2	Name btnContinue	Click
Button3	Name btnExit	Click
Form1	AcceptButton: btnEnter CancelButton: btnExit	Form Closing

Bước 3: Viết sự kiện cho các hàm vừa tạo:

Sự kiện cảnh báo đóng cửa sổ - Form Closing thực hiện tương tự như bài tập mẫu.

Sự kiện ngăn người dùng nhấn các kí tự khác số - Xử lý trong sự kiện KeyPress tương tự như bài tập 3.

Sự kiện cho button Tiếp tục:

```
private void btnContinue_Click(object sender, EventArgs e)
{
    arr.Clear();
    txtArrayNumber.Clear();
    txtSumArrayNumber.Clear();
    txtSumEvenNum.Clear();
    txtSumOddNum.Clear();
    txtNumber.Clear();
    txtNumber.Focus();
}
```

Sự kiện cho button Nhập:



Bên ngoài tạo một biến lưu trữ mảng vừa nhập:

```
List<long> arr=new List<long>();
```

Trong sự kiện Nhập viết như sau:

```
private void btnEnter_Click(object sender, EventArgs e)
{
    if (txtNumber.Text.ToString().Equals(""))
    {
        MessageBox.Show("Số không được để trống!!", "Lỗi",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        txtNumber.Focus();
    }
    else
    {
        long number = long.Parse(txtNumber.Text); arr.Add(number);
        txtArrayNumber.AppendText(number.ToString()+" ");
        txtSumArrayNumber.Text = SumArrayNumber(arr);

        txtSumEvenNum.Text = SumEvenNum(arr); txtSumOddNum.Text
            = SumOddNum(arr);

    }
}
```

Trong đó có các hàm hỗ trợ như sau: