

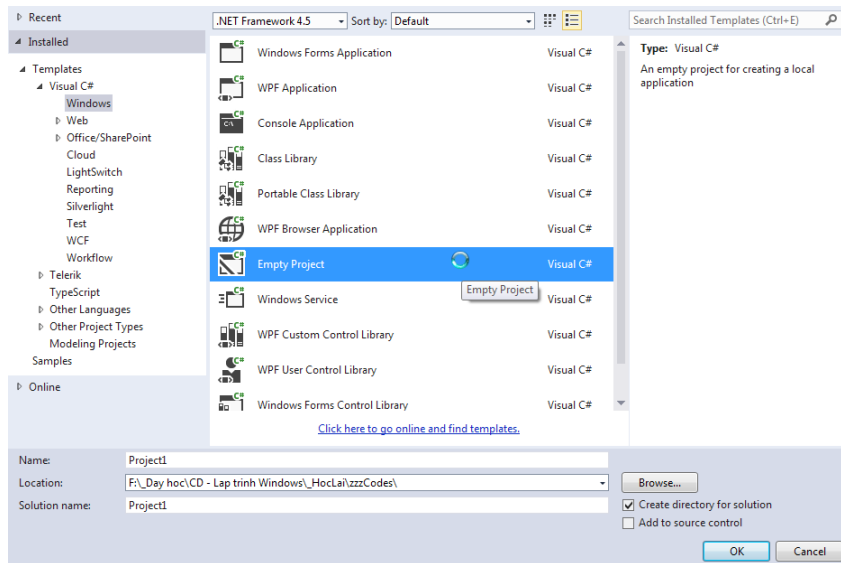
Mục tiêu

Luyện tập cơ bản C# với lập trình Winform.

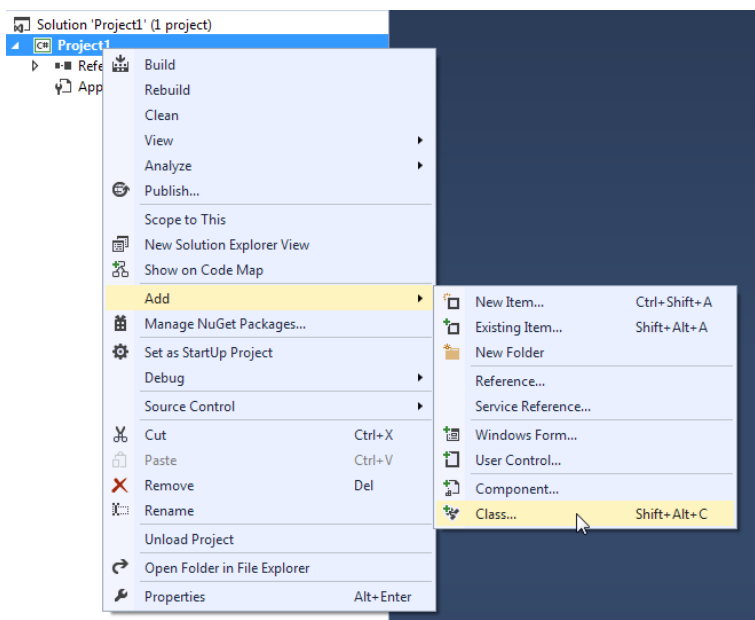
Nội dung

Chương trình Winform đơn giản

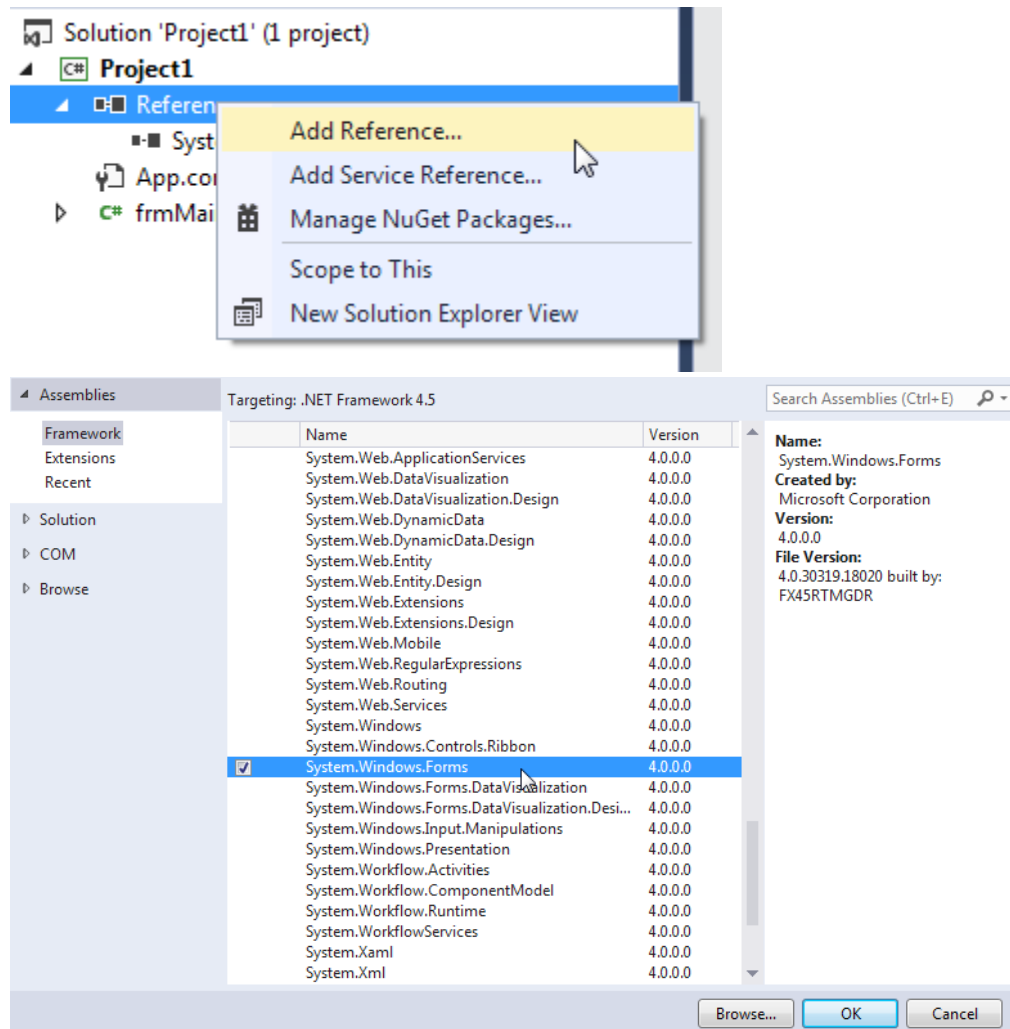
- ✓ Bước 1: mở Visual Studio và tạo Project với ngôn ngữ C# kiểu “Empty Project”



- ✓ Bước 2: chọn Add New Class để tạo class frmMain



- ✓ Bước 3: Chọn Add Reference để thêm các thư viện cần thiết cho Winform của .NET (System.Drawing và System.Windows.Forms)



- ✓ Bước 4: Code frmMain như sau

```
public class frmMain : Form
{
    //khai báo đối tượng Label để thể hiện text
    Label lbl;

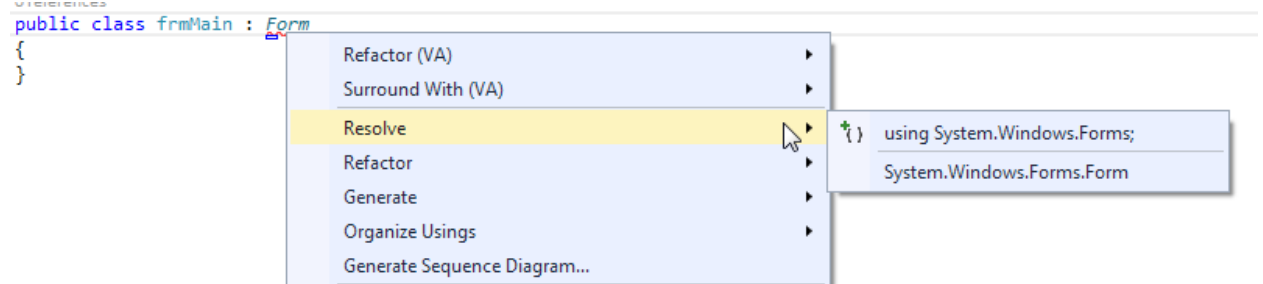
    public frmMain()
    {
        //luôn phải new đối tượng trước khi sử dụng
        lbl = new Label();
        //gán thuộc tính chính của Label
        lbl.Text = "Hello World!";
        //set giá trị các thuộc tính thể hiện
        lbl.Font = new Font("tahoma", 20f);
        lbl.AutoSize = true;
        lbl.Left = 50;
        lbl.Top = 100;
        //thành phần chỉ thực sự là thuộc Form
    }
}
```

```

        //sau khi được add vào mảng Controls
        this.Controls.Add(lbl);
    }
}

```

Cách thêm namespace nhanh chóng nhất



✓ Bước 5: Thêm class Program và code như sau

```

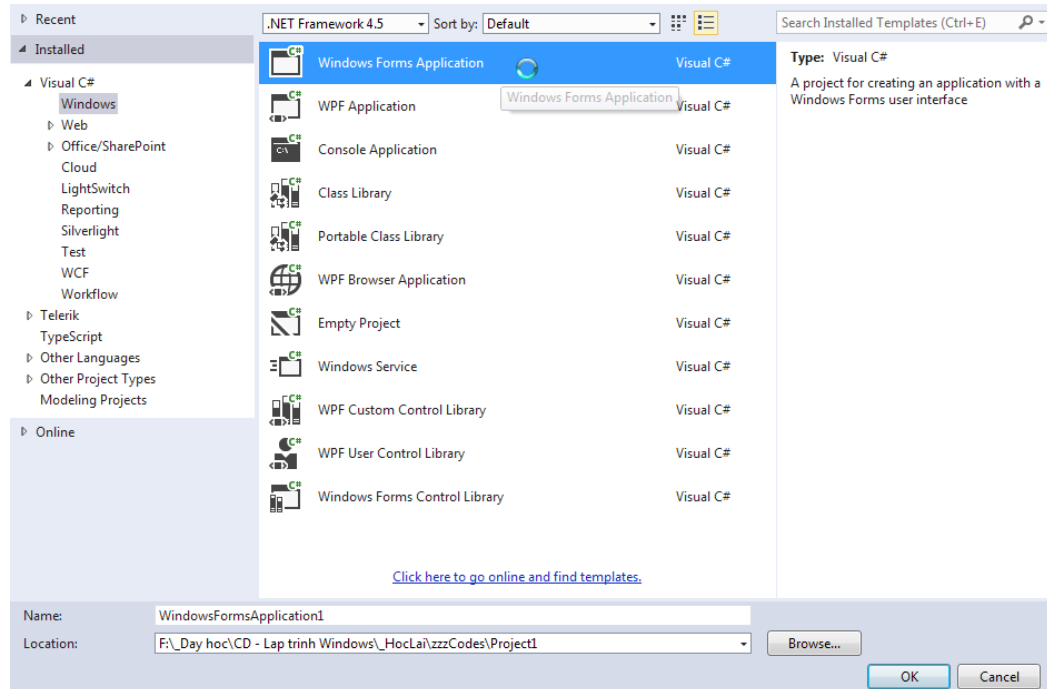
static class Program
{
    static void Main()
    {
        Application.Run(new frmMain());
    }
}

```

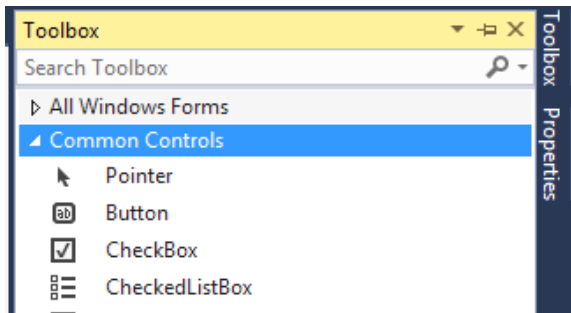
✓ Bước 6: chạy chương trình.

Chương trình tính toán phép tính đơn giản

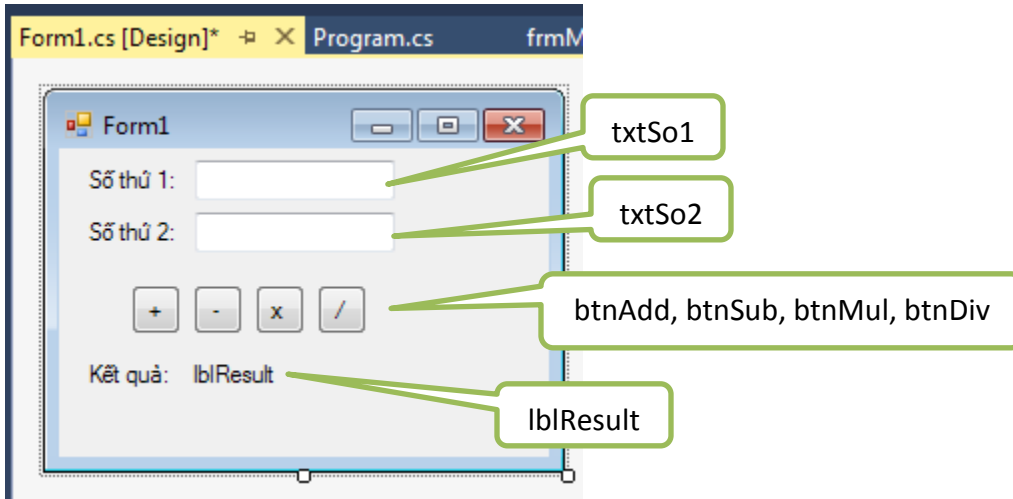
✓ Bước 1: Tạo project dạng Windows Forms Application



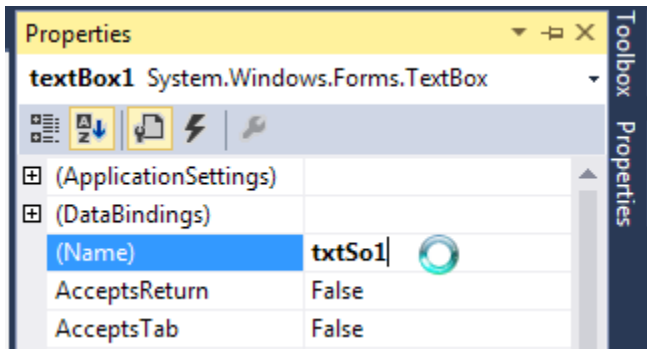
✓ Bước 2: Mở Toolbox để thêm các controls cần thiết lên giao diện form



- ✓ Bước 3: tạo giao diện như sau



- ✓ Bước 4: đặt tên các control cần sử dụng truy suất giá trị (thuộc tính "(name)" trong properties của design).



- ✓ Bước 5: code trong class Form1

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        //gán hàm xử lý sự kiện
        btnAdd.Click += new EventHandler(btn_Click);
        btnSub.Click += new EventHandler(btn_Click);
        btnMul.Click += new EventHandler(btn_Click);
        btnDiv.Click += new EventHandler(btn_Click);
    }
}
```

```

void btn_Click(object sender, EventArgs e)
{
    //lấy các giá trị nhập vào cho số 1 và số 2
    float so1, so2;
    //lưu ý khi chuyển chuỗi sang số thì nên sử dụng phương thức TryParse
    if(!float.TryParse(txtSo1.Text, out so1))
    {
        so1 = 0;
    }
    if (!float.TryParse(txtSo2.Text, out so2))
    {
        so2 = 0;
    }
    switch((sender as Button).Text)
    {
        case "+":
            lblResult.Text = (so1 + so2).ToString();
            break;
        case "-":
            lblResult.Text = (so1 - so2).ToString();
            break;
        case "x":
            lblResult.Text = (so1 * so2).ToString();
            break;
        case "/":
            if(so2 == 0)
            {
                so2 = 1;
            }
            lblResult.Text = (so1 / so2).ToString();
            break;
    }
}
}

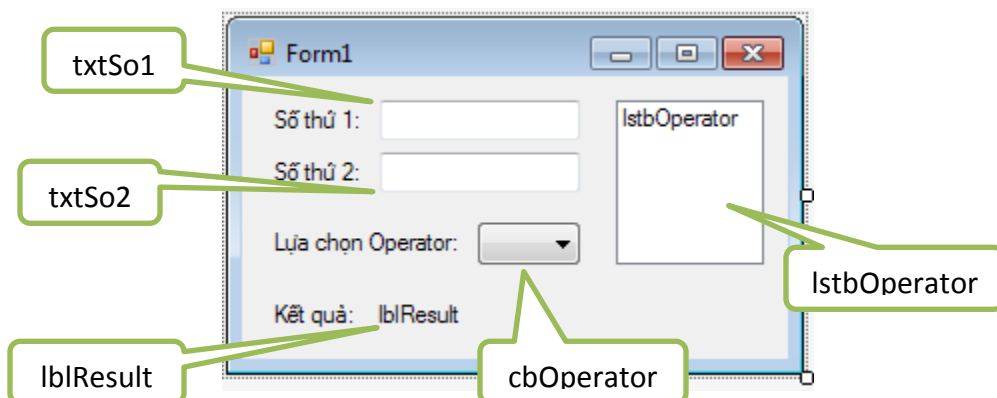
```

✓ Bước 6: chạy chương trình.

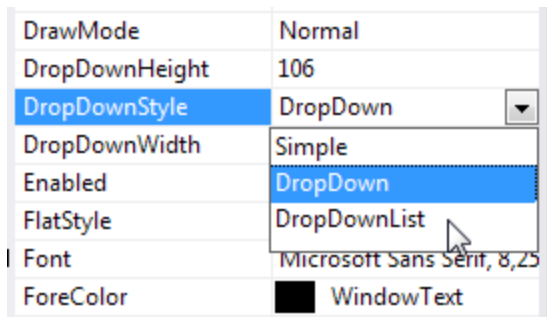
Sử dụng các controls thể hiện danh sách (ComboBox và ListBox)

✓ Bước 1: Tạo project loại Windows Forms Application

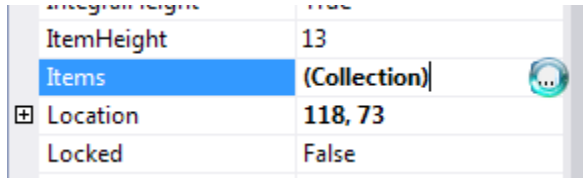
✓ Bước 2: Tạo giao diện như sau



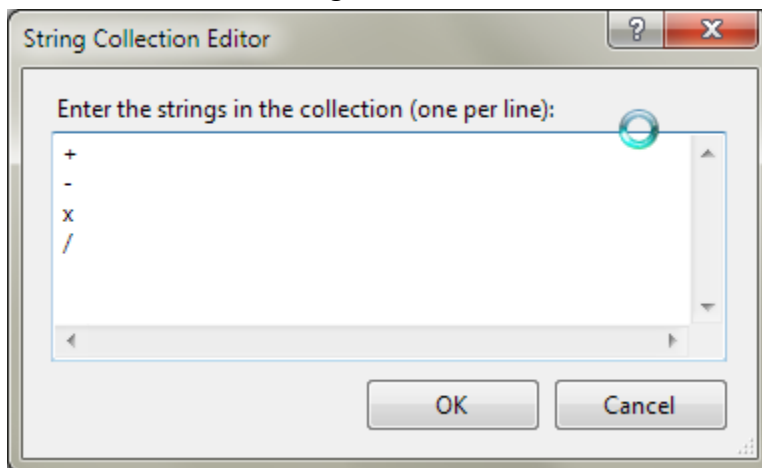
✓ Bước 3: Mở bảng Properties của ComboBox để set các giá trị thuộc tính như sau



Chọn kiểu DropDownList



Chọn Items để điền thông tin như sau



- ✓ Bước 4: Set Items của ListBox tương tự.
- ✓ Bước 5: Code class Form1 như sau

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

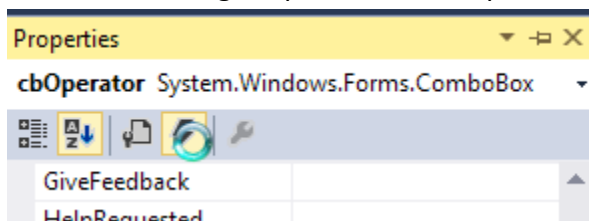
    void selectedIndexChanged(object sender, EventArgs e)
    {
        float so1, so2;
        if (!float.TryParse(txtSo1.Text, out so1))
        {
            so1 = 0;
        }
        if (!float.TryParse(txtSo2.Text, out so2))
        {
            so2 = 0;
        }
    }
}
```

```

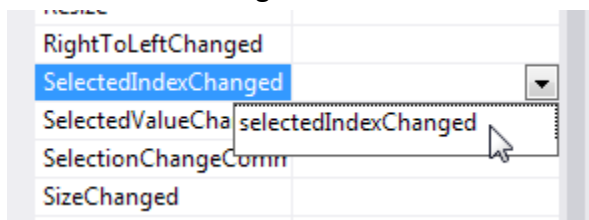
//lưu ý thuộc tính Text là mặc định từ class Control do đó dù là ComboBox
//hay ListBox đều có thể truy suất thuộc tính Text mà không cần quan tâm
//thật sự đang là đối tượng nào bằng cách đưa về kiểu Control (xem lại
//kiến thức hướng đối tượng phần kế thừa)
switch ((sender as Control).Text)
{
    case "+":
        lblResult.Text = (so1 + so2).ToString();
        break;
    case "-":
        lblResult.Text = (so1 - so2).ToString();
        break;
    case "x":
        lblResult.Text = (so1 * so2).ToString();
        break;
    case "/":
        if (so2 == 0)
        {
            so2 = 1;
        }
        lblResult.Text = (so1 / so2).ToString();
        break;
}
}
}

```

- ✓ Bước 5: Mở bảng Properties của cbOperator, lựa chọn tab events



Lựa chọn phương thức selectedIndexChanged (đã code) cho event
SelectedIndexChanged



- ✓ Bước 6: Chạy chương trình.

Bài tập

Xây dựng chương trình với giao diện sau

Form1

Số dòng: 4 Số cột: 7

Tạo Form

Khi nhấn “Tạo Form” thì hiển thị form 2

Form2

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27

Khi nhấn bất cứ button nào thì hiển thị MessageBox thông tin vị trí của button

2, 5

OK