

# LAB 1

## Advanced Programming in C#

### IBB UN

#### Level 3 IT



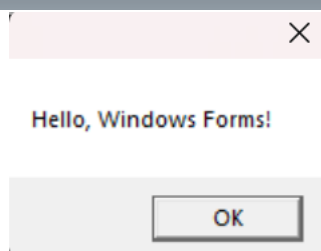
1. إنشاء تطبيق بسيط باستخدام Windows Forms.
  2. عرض نافذة باستخدام Form وMessageBox
  3. تخصيص Form وتعيين عنوان.
  4. إضافة تفاعل باستخدام Button والتعامل مع أحداثه.
- شرح تطبيقات Windows Forms من البداية

عند تطوير تطبيقات باستخدام Windows Forms، يعتبر الـ **Form** هو العنصر الأساسي. يعتبر الـ **Form** نافذة تُعرض للمستخدم، وهذه النوافذ هي وحدة واجهة المستخدم التي تستخدمها أنظمة **Windows** منذ البداية. في تطبيقات **Windows Forms**، يوجد عادة **Form** رئيسي، وهو الذي يدير نوافذ أخرى قد تظهر أثناء حياة التطبيق.

#### التطبيق الأول: رسالة بسيطة

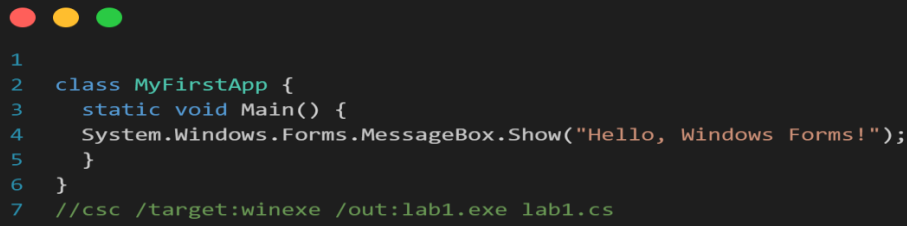
يتمثل أول تطبيق بسيط في عرض رسالة باستخدام **MessageBox**:

```
1 class MyFirstApp {
2     static void Main() {
3         System.Windows.Forms.MessageBox.Show("Hello, Windows Forms!");
4     }
5 }
6 //csc /target:winexe /out:lab1.exe lab1.cs
```



#### • فتح نافذة باستخدام Form

في تطبيقات **Windows Forms**، الـ **Form** هو وحدة أساسية في بناء واجهة المستخدم، وعند إنشاء تطبيق باستخدام **Form**، يمكنك فعل ذلك كالتالي:



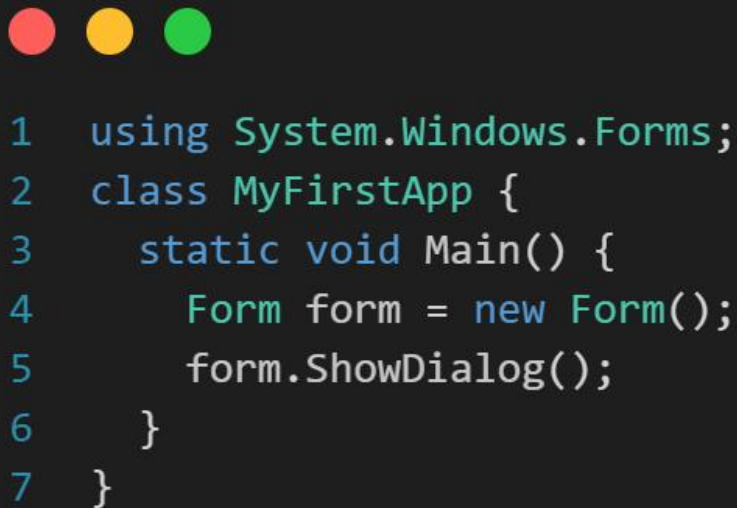
```

1
2 class MyFirstApp {
3     static void Main() {
4         System.Windows.Forms.MessageBox.Show("Hello, Windows Forms!");
5     }
6 }
7 //csc /target:winexe /out:lab1.exe lab1.cs

```

- **Show** تعرض النافذة بشكل غير دائم، مما يعني أن التحكم يعود مباشرة إلى الدالة **Main** بعد عرض الـ **Form**، مما يؤدي إلى غلق النافذة فوراً.

لحل هذه المشكلة وجعل النافذة تبقى مفتوحة حتى إغلاقها، يمكننا استخدام **ShowDialog** بدلاً من **Show**:



```

1 using System.Windows.Forms;
2 class MyFirstApp {
3     static void Main() {
4         Form form = new Form();
5         form.ShowDialog();
6     }
7 }

```

عرض نافذة بشكل رئيسي:

لتعيين **Form** كـ **Form** رئيسي للتطبيق، نستخدم دالة **Application.Run** التي تستقبل الـ **Form** كوسيط:



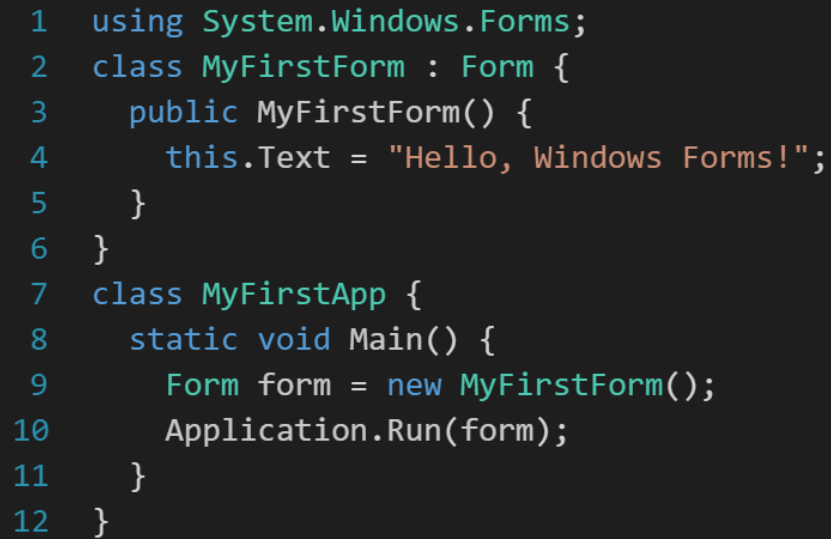
```
1 using System.Windows.Forms;
2 class MyFirstApp {
3     static void Main() {
4         Form form = new Form();
5         Application.Run(form);
6     }
7 }
```

تخصيص الـ Form:



```
1 using System.Windows.Forms;
2 class MyFirstApp {
3     static void Main() {
4         Form form = new Form();
5         form.Text = "Hello, Windows Forms!";
6         Application.Run(form);
7     }
8 }
```

## • استخدام Custom Form:



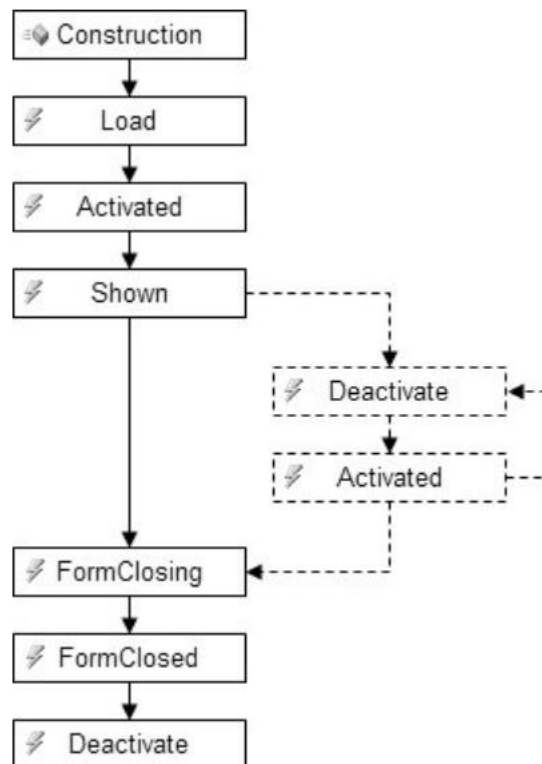
```
1 using System.Windows.Forms;
2 class MyFirstForm : Form {
3     public MyFirstForm() {
4         this.Text = "Hello, Windows Forms!";
5     }
6 }
7 class MyFirstApp {
8     static void Main() {
9         Form form = new MyFirstForm();
10        Application.Run(form);
11    }
12 }
```

إضافة تفاعل باستخدام Button:



```
1  using System;
2  using System.Windows.Forms;
3
4  class MyFirstForm : Form {
5      public MyFirstForm() {
6          this.Text = "Hello, Windows Forms!";
7          Button button = new Button();
8          button.Text = "Click Me!";
9          button.Click += new EventHandler(button_Click);
10         this.Controls.Add(button);
11     }
12
13     void button_Click(object sender, EventArgs e) {
14         MessageBox.Show("click...");
15     }
16 }
17
18 class MyFirstApp {
19     static void Main() {
20         Form form = new MyFirstForm();
21         Application.Run(form);
22     }
23 }
```

- فيما يلي شرح لفترة حياة النموذج (Form Lifetime) في تطبيقات Windows Forms:



### Form Lifetime Event Sequence

#### 1. فتح النموذج (Form Opening)

يبدأ النموذج في الظهور عند استدعاء المُنشئ الخاص به (constructor) ، حيث يتم تنفيذ `InitializeComponent` الذي يقوم بإنشاء جميع عناصر التحكم (controls) المضافة مسبقًا عبر المصمم.



```
1 public Form1() {  
2     InitializeComponent();  
3 }
```

يتم استدعاء الحدث **Load** بعد استدعاء **Show** أو **ShowDialog** لعرض النموذج، وهو الوقت المناسب لإجراء أي تهيئة نهائية أو لاستعادة الإعدادات السابقة:



```
1 void Form1_Load(object sender, EventArgs e) {  
2     MessageBox.Show("Loading Form1!");  
3 }
```

## 2. تنشيط النموذج (Form Activation)

- عند ظهور النموذج، يصبح النموذج النشط ويستقبل المدخلات من لوحة المفاتيح. يتم إطلاق حدث **Activated** عند تنشيط النموذج:



```
1 void Form1_Activated(object sender, EventArgs e)
2 {
3     MessageBox.Show("Form1 activated!");
4 }
```

- بعد ذلك، يتم إطلاق حدث Shown للإشارة إلى أن النموذج قد تم عرضه بالكامل:



```
1 void Form1_Shown(object sender, EventArgs e)
2 {
3     MessageBox.Show("Form1 shown!");
4 }
```

### 3. إلغاء تنشيط النموذج (Form Deactivation)

- عندما ينتقل المستخدم بعيداً عن التطبيق، مثلما يحدث عند الضغط على **Alt+Tab**، يتم إلغاء تنشيط النموذج ويتم إطلاق الحدث **Deactivate**. يمكنك استخدام هذا الحدث لإيقاف الأنشطة التي تحتاج إلى التفاعل مع المستخدم:



```
1 void Form1_Deactivate(object sender, EventArgs e)
2 {
3     MessageBox.Show("Form1 Deactivate!");
4 }
```



- عند العودة إلى التطبيق، يتم إطلاق الحدث **Activated** مرة أخرى لاستئناف الأنشطة المتوقفة:



```
1 void Form1_Activated(object sender, EventArgs e)
2 {
3     MessageBox.Show("Form1 activated!");
4 }
```

#### 4. إغلاق النموذج (Form Closing)

- عند إغلاق النموذج، سواء من خلال القائمة أو زر الإغلاق (X)، يتم إطلاق حدث **FormClosing**. يمكنك إلغاء إغلاق النموذج باستخدام **FormClosingEventArgs.Cancel**:



```
1 void Form1_FormClosing(object sender, FormClosingEventArgs e)
2 {
3     DialogResult result = MessageBox.Show("Abort your Form1?", "Form1 In Progress", MessageBoxButtons.YesNo);
4     e.Cancel = (result == DialogResult.No);
5 }
```

- بعد إغلاق النموذج، يتم إطلاق الحدث **FormClosed**



```
1 void Form1_FormClosed(object sender, FormClosedEventArgs e)
2 {
3     MessageBox.Show("Your Form1 was aborted");
4     // MessageBox.Show("Your Form1 was aborted: " + e.CloseReason.ToString());
5 }
```

#### 5. السبب في إغلاق النموذج (CloseReason)

- يمكنك معرفة سبب إغلاق النموذج من خلال خاصية **CloseReason** في **FormClosedEventArgs** و **FormClosingEventArgs**



```
1 void Form1_FormClosed(object sender, FormClosedEventArgs e)
2 {
3     MessageBox.Show("Your Form1 was aborted");
4     // MessageBox.Show("Your Form1 was aborted: " + e.CloseReason.ToString());
5 }
```

## 6. إلغاء التنشيط قبل الإغلاق (Deactivation before Closing)

- بعد إطلاق **FormClosed**، يتم إطلاق حدث **Deactivated** قبل أن يختفي النموذج تمامًا من الواجهة.



```
1 private void Form1_Paint(object sender, PaintEventArgs e)
2 {
3     MessageBox.Show("Form1_Paint");
4 }
```

الواجب:

- 1- تطبيق المحاضرة كامل بملف واحد مع اخذ صورة لاسم مستخدم الجهاز.
- 2- الاطلاع على خصائص الفورم ومحاولة تطبيقها.