

**C++ Programming in
easy steps, 4th edition
Visual Studio 2015 Update**

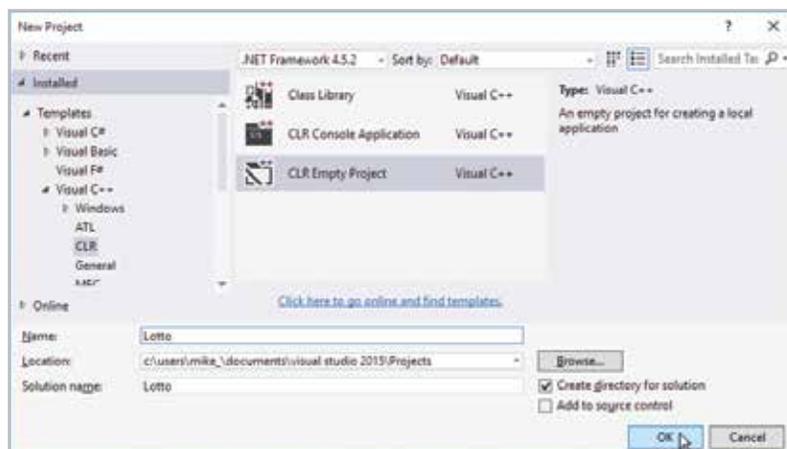


You can run the installer **vs_community.exe** at any time to add more components to your Visual Studio 2015 Community IDE.

Programming Visually with Visual Studio 2015 Community Edition

Visual Studio 2015 Community Edition does not currently include the Windows Forms Application template for Visual C++ that was present in earlier editions of Visual Studio. Instead, a Windows Forms Application can be created in Visual Studio 2015 Community edition IDE like this:

- 1 On the **File** menu, click **New > Project...** – to open the “New Project” dialog
- 2 In the left-hand pane, expand **Installed > Templates > Visual C++ > Windows**, then choose **CLR**, and in the right-hand pane select **CLR Empty Project**



- 3 Enter a project **Name**, for example “Lotto”, then click the **OK** button – to create the new project in the IDE
- 4 On the IDE’s **Project** menu, click **Add New Item** – to open the “Add New Item” dialog
- 5 In the left-hand pane, expand **Installed > Visual C++**, then choose **UI**

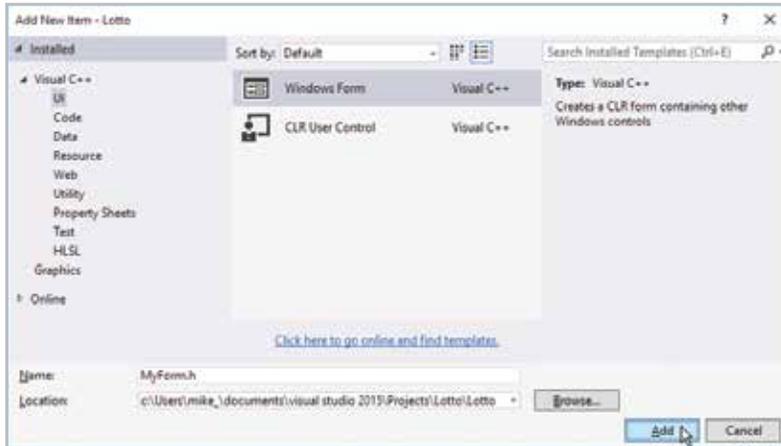


If you do not see a **CLR** item in the **New Project** dialog, use the **Click here to go online and find templates** link to install the **CLR** template.

...cont'd

6

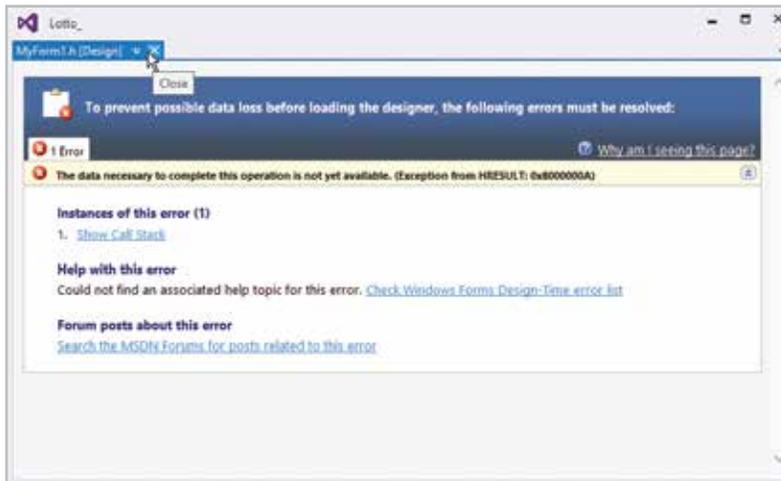
In the right-hand pane, select **Windows Form**, then click the **Add** button – to add a blank form to the project



Leave the default name of the Windows Form addition as **MyForm.h**

7

Ignore the Error message window that may now appear for **MyForm.h [Design]** – simply click the **X** button to close the Error message window



The Error window that may appear is a known issue with the first release of Visual Studio 2015 Community edition. It is due to be fixed in the subsequent update release.

The Error window is created by a bug in the current release of the Visual Studio 2015 Community edition, at the time of writing.



If your project is named other than “Lotto” you will have to amend the **using namespace** line to substitute the name of your project for **Lotto**.

...cont'd

- 8 On the IDE’s **View** menu, click **Solution Explorer** to open the “Solution Explorer” window, then expand the project’s **Source Files** folder – to see a file **MyForm.cpp**
- 9 Double-click the **MyForm.cpp** file icon, to open that file in the “Code Editor” window, then edit the code to read exactly as below, and then click **Save All** – to save the changes

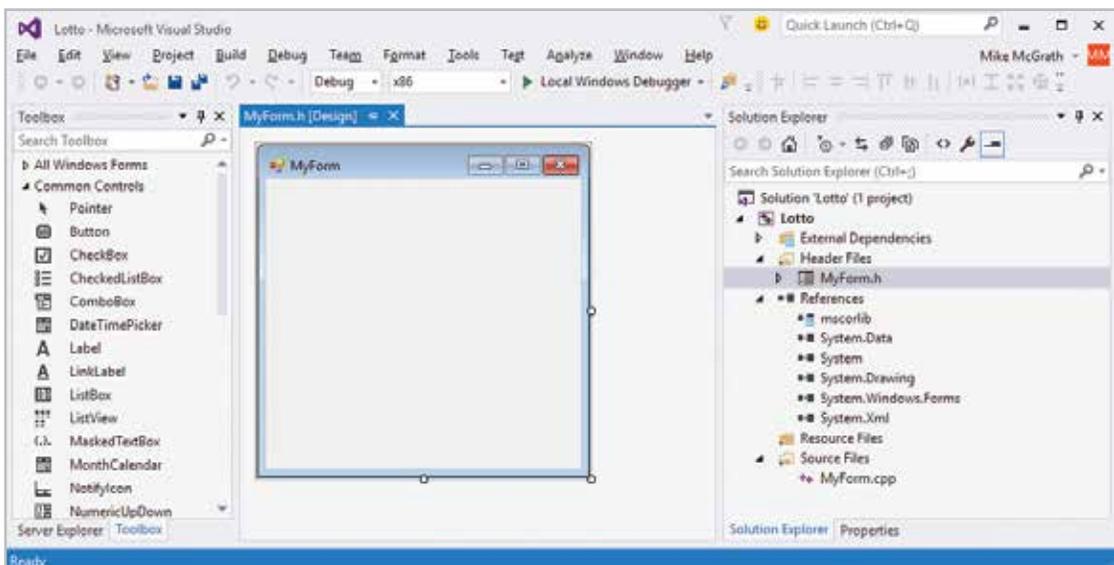
```
#include "MyForm.h"
#include <Windows.h>
```

```
using namespace Lotto ;
```

```
int __stdcall WinMain( HINSTANCE hInst, HINSTANCE hPInst,
                      LPSTR lpCmdLine, int nCmdShow )
```

```
{
    MyForm frm ;
    frm.ShowDialog() ;
    return 0 ;
}
```

- 10 Back in the “Solution Explorer” window, expand the project’s **Header Files** folder, then double-click the **MyForm.h** file icon – to see the form “Designer” window



Other changes to note

Creating a Windows Forms Application in this way with the Visual Studio 2015 Community IDE makes a number of changes to the generated code described in previous editions:

- The Windows Form added to the project has the default name “MyForm”, so the form’s header file is named **MyForm.h** – rather than **Form1.h**
- Similarly, the code in the **MyForm.h** header file describes a class named **MyForm** – rather than **Form1**
- The generated code in the **MyForm.h** header file does not have a **Form1_Load** function, (or **MyForm_Load** function) – it now has a “constructor” function where you can place calls to run functions when the form first loads. The constructor code has a “TODO” reminder comment and looks like this:

```
public:
    MyForm( void )
    {
        InitializeComponent() ;
        // TODO: Add constructor code here
    }
```

- In our “Lotto” example, the **Clear** function can be called from the constructor, rather than **Form1_Load** as before, like this:

```
public:
    MyForm( void )
    {
        InitializeComponent() ;
        // TODO: Add constructor code here
        Clear() ;
    }
```

Other steps in the “Lotto” example can be implemented as before in earlier versions of Visual Studio Express edition.

