UNIVERSITY INSTITUTE OF INFORMATION TECHNOLOGY, PMAS ARID AGRICULTURE UNIVERSITY, MURREE ROAD, RAWALPINDI

Windows Application development In Microsoft Visual Studio 2010

Easy to follow Steps to become a Professional in C#

By Saif Ur Rehman

11/14/2016

Windows Application development In Microsoft Visual Studio 2010

By: Saif ur Rehman

Hands out are prepared by

Saif Ur Rehman

Assistant Professor

University Institute of Information

Technology

PMAS Arid Agriculture University

Murree Road, Rawalpindi

More Hands out on ASP.Net with C#, PHP with MySQL, Java, Programming Fundamentals and much more can be found at

www.aspuiit.blogspot.com

Suggestions are welcome at

Saif at uaar dot edu dot pk

Contents in Brief

About the Tutorial

This tutorial is designed specifically to teach you to think like a programmer and the C# language is just a tool that can be replaced by any other modern programming languages, such as Java.

I hope as time passes this tutorial will become mature and we will try to remove all the mistakes you conveyed to us

- **Saif ur Rehman**

Dedication

To my great mentor, teacher Prof. Dr. Sohail Asghar

Saif ur Rehman

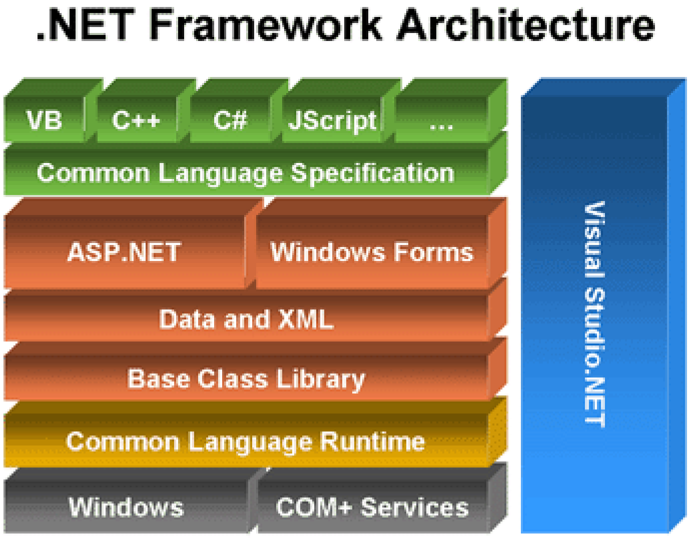
Chapter No.1

Introduction to .Net Framework

Fundamental Concepts of .Net Framework

<u>The .NET Framework</u> is a technology that supports building and running the next generation of applications and XML Web services. The .NET Framework is designed to fulfil the following objectives [1]

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

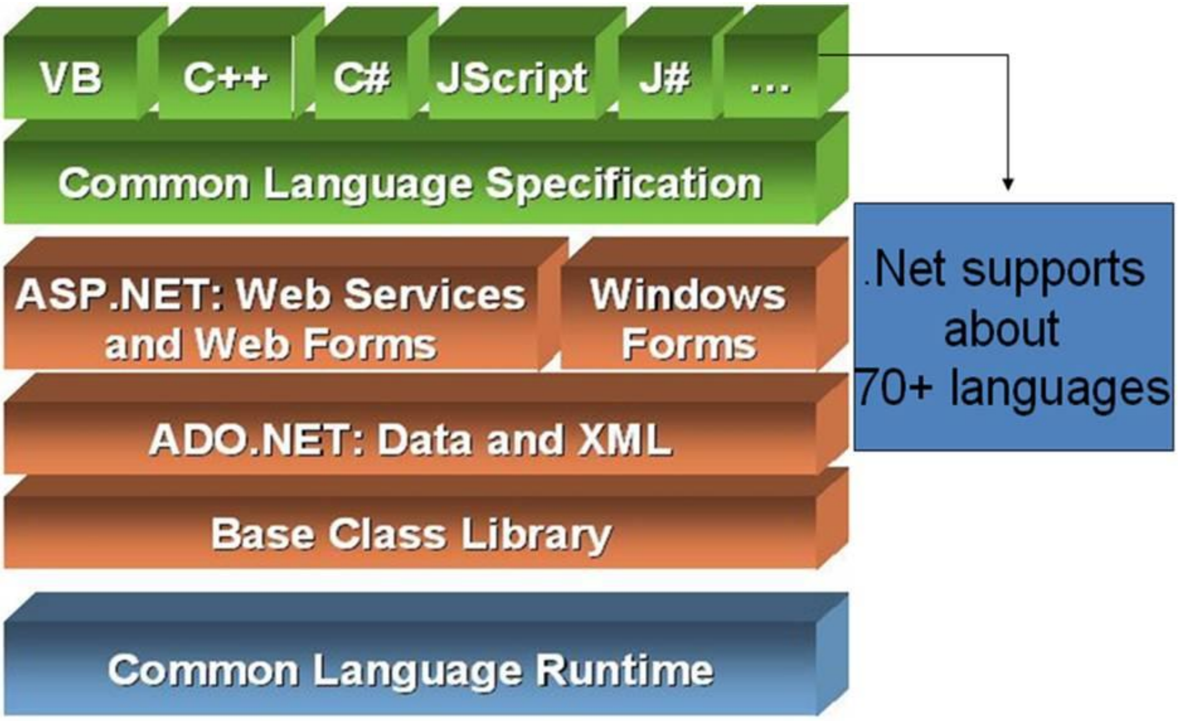


[1]

.Net Framework Components

The .NET Framework consists of the common language runtime and the .NET Framework class library.

The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that promote security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.


VB
C++
C#
JScript
J#
...
Common Language Specification
ASP.NET: Web Services and Web Forms
Windows Forms
ADO.NET: Data and XML
Base Class Library
Common Language Runtime
.Net supports about 70+ languages

[2]

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable ASP.NET applications and XML Web services, both of which are discussed later in this topic.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and isolated file storage.

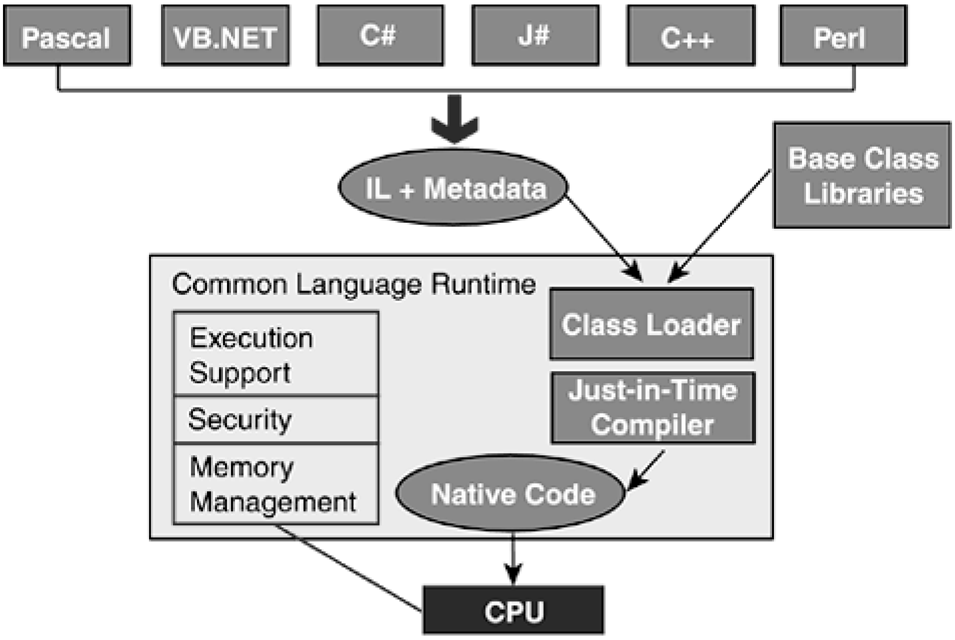
The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.

Features of the Common Language Runtime

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich.



[3]

The runtime also enforces code robustness by implementing a strict type-and-code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

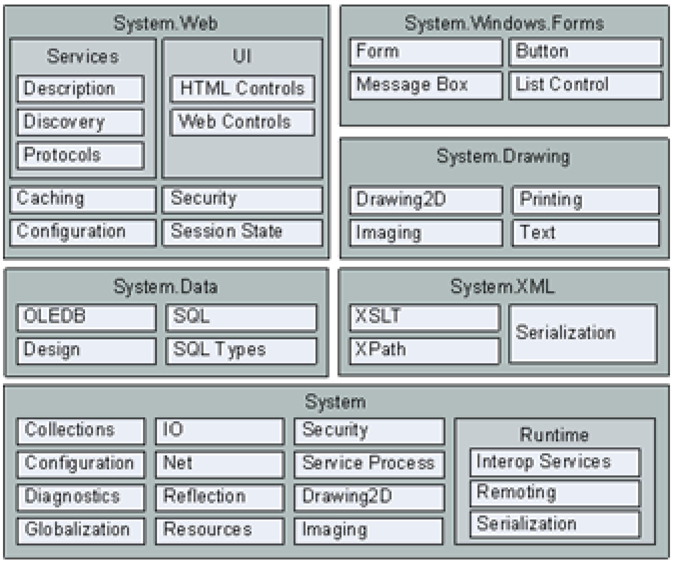
The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

.NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework

.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.



[4]

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

-
- Console applications
- Windows GUI applications (Windows Forms)
- Windows Presentation Foundation
- ASP.NET applications.
- Windows services
- Service-oriented applications using Windows Communication Foundation (WCF).
- Workflow-enabled applications using Windows Workflow Foundation (WF).
- .

For example, the Windows Forms classes are a comprehensive set of reusable types that vastly simplify Windows GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

NET Framework Versions

The following table provides a brief review of the .NET Framework versions and the associated CLR version. It also shows the Visual Studio version that provided the development environment when that version of the .NET Framework was introduced. However, with the multi-targeting feature of Visual Studio, you are not limited to only that version of the .NET Framework **[6]**

.NET Framework version	CLR version	Visual Studio version	Description
1.0	1.0	Visual Studio .NET	Contained the first version of the CLR and the first version of the base class libraries.
1.1	1.1	Visual Studio .NET 2003	Included updates to ASP.NET and ADO.NET. This version was subsequently updated twice, with Service Pack 1 (SP1) and SP2. This version also introduced side-by-side execution, which enables applications on a single computer to run against multiple versions of the CLR.
2.0	2.0	Visual Studio 2005	Introduced a new version of the CLR with additions to the base class libraries, including generics, generic collections, and significant additions to ASP.NET. This version was subsequently updated with SP1 and SP2.
3.0	2.0	Visual Studio 2005	This version is essentially .NET Framework 2.0 with the addition of Windows Presentation Foundation (WPF), Windows Communications Foundation (WCF), Windows Workflow Foundation (WF), and CardSpace. It was subsequently updated with SP1 and SP2.
3.5	2.0	Visual Studio 2008	Added new features such as AJAX-enabled Web sites and LINQ. The SP1 update added the .NET Framework Client Profile, Dynamic Data, and a small set of additional enhancements.
4	4	Visual Studio 2010	Includes a new version of the CLR, expanded base class libraries, and new features such as the Managed Extensibility Framework (MEF), dynamic language runtime (DLR), and code contracts.

Programming Languages Supported by .Net Framework [7]

Choosing a programming language depends on your language experience and the scope of the application you are building. While small applications are often created using only one language, it is not uncommon to develop large applications using multiple languages.

The .NET Platform programming languages — including Visual Basic .NET, Visual C#, Managed Extensions for C++, and many other programming languages from various vendors — use .NET Framework services and features through a common set of unified classes.

In most situations, you can effectively use all of the Microsoft programming languages. Nevertheless, each programming language has its relative strengths and you will want to understand the features unique to each language.

Visual Basic .NET is the next generation of the Visual Basic language from Microsoft. With Visual Basic you can build .NET applications, including Web services and ASP.NET Web applications, quickly and easily. Applications made with Visual Basic are built on the services of the common language runtime and take advantage of the .NET Framework.

Visual Basic provides prototypes of some common project types, including:

- Windows Application.
- Class Library.
- Windows Control Library.
- ASP.NET Web Application.
- ASP.NET Web Service.
- Web Control Library.
- Console Application.
- Windows Service.

Visual C# (pronounced C sharp) is designed to be a fast and easy way to create .NET applications, including Web services and ASP.NET Web applications. Applications written in Visual C# are built on the services of the common language runtime and take full advantage of the .NET Framework.

C# is a simple, elegant, type-safe, object-oriented language recently developed by Microsoft for building a wide range of applications. Anyone familiar with C and similar languages will find few problems in adapting to C#. C# is designed to bring rapid development to the C++ programmer without sacrificing the power and control that are a hallmark of C and C++. Because of this heritage, C# has a high degree of fidelity with C and C++, and developers familiar with these languages can quickly become productive in C#. C# provides intrinsic code trust mechanisms for a high level of security, garbage collection, and type safety. C# supports single inheritance and creates Microsoft intermediate language (MSIL) as input to native code compilers.

C# is fully integrated with the .NET Framework and the common language runtime, which together provide language interoperability, garbage collection, enhanced security, and improved versioning support. C# simplifies and modernizes some of the more complex aspects of C and C++, notably namespaces, classes, enumerations, overloading, and structured exception handling. C# also eliminates C and C++ features such as macros, multiple inheritance, and virtual base classes. For current C++ developers, C# provides a powerful, high-productivity language alternative.

Visual C# provides prototypes of some common project types, including:

- Windows Application.
- Class Library.
- Windows Control Library.
- ASP.NET Web Application.
- ASP.NET Web Service.

- Web Control Library.
- Console Application.
- Windows Service.

Visual C++ .NET is the next generation of the Visual C++ language from Microsoft. Visual C++ has always been the best language for creating high-performance applications for Microsoft Windows and the World Wide Web.

Visual C++ provides a number of important libraries to help you code applications, including Active Template Library (a set of template-based C++ classes for COM objects), ATL Server Library (a set of native C++ classes for creating Web applications, Web Services, and other server applications), and Microsoft Foundation Classes (a set of classes that support an application written for the Windows API).

Visual C++ provides prototypes of some common project types, including:

- Active Template Library (ATL) Project.
- ATL Server Project.
- ATL Server Web Service.
- Custom Wizard.
- Extended Stored Procedure DLL.
- Makefile Project.
- ASP.NET Web Service
- Class Library (.NET)
- Console Application (.NET)
- Windows Control Library (.NET)
- Windows Forms Application (.NET)
- Windows Service (.NET)
- Microsoft Foundation Classes (MFC) ActiveX Control.
- MFC Application.
- MFC DLL.

- MFC ISAPI Extension DLL.
- Win32 Project.

Managed Extensions for C++ provides a simple way to integrate existing applications into the .NET Framework. For example, maybe you need to port some unmanaged code to .NET. Because you already have previously compiled static, linked libraries, DLLs, and various unmanaged C++ classes, you can simply compile your existing Win32 code as a .NET application. Then, as time permits, you can re-engineer the components to take advantage of managed code features.

Transact-SQL is the native language for storing, modifying, and retrieving information in Microsoft SQL Server relational databases. You can also use Transact-SQL to create databases and any of the objects stored in a database, such as tables, columns, triggers, keys, indexes, views, stored procedures, and functions. Transact-SQL is fully supported in the Visual Studio editor and in the designers provided with Visual Database Tools.

Windows Script Host (WSH) is a language-independent scripting environment for 32-bit Windows platforms. With WSH, Microsoft offers VBScript, JScript, and JScript .NET scripting engines. These scripting languages can be used in the ASP pages of a Web server, in HTML pages that run in Internet Explorer, and in Windows Script Host scripting engines on Windows 98 and Windows 2000.

Microsoft Visual Basic Scripting Edition (VBScript) is a subset of Microsoft Visual Basic — it will look very familiar to you if you have ever used Visual Basic. It is not identical, however. Because VBScript is specifically designed to work in Internet Explorer (IE) browsers, it does not include features that are normally outside the scope of scripting, such as file access and printing. However, it is common to use the FileSystem Object with VBScript to manipulate files.

Microsoft JScript is designed for Web page scripting. JScript conforms to the ECMA 262 language specification. JScript is a powerful scripting language specifically targeted at the Internet. Like VBScript, JScript is an interpreted, object-based scripting language that processes source code embedded directly in HTML pages. JScript runs on both Internet Explorer and Netscape browsers.

Like VBScript, JScript talks to host applications using Windows Script Host. With Windows Script Host, browsers and other host applications do not require special integration code for each scripting component. Windows Script Host enables a host to compile scripts, obtain and call entry points, and manage the namespace available to the developer.

JScript .NET is the next generation of Microsoft's implementation of the ECMA 262 language, developed in conjunction with ECMAScript Edition 4. It is designed to run within the common language runtime to manage the execution of code and provide services that make the development process easier. With JScript .NET you get features such as cross-language integration, cross-language exception handling, enhanced security, versioning and deployment support, a simplified model for component interaction, and debugging and profiling services.

XML provides a format for describing structured data that allows for precise declarations of content and useful search results across multiple platforms. XML defines information and data according to purpose rather than presentation so that several applications can use the information and data in ways that promote diverse application reuse and extensibility. XML is an increasingly important meta-markup language that is convenient for use on the Internet.

Visual J++ : Microsoft provides the Java User Migration Path to Microsoft .NET (JUMP to .NET) as a set of technologies and services that enable programmers to preserve, enhance, and migrate Java language projects onto the Microsoft .NET Platform. With JUMP to .NET you can continue to take advantage of existing Visual J++ skills and source code while extending your application and components onto the .NET Platform. If you are familiar with the Visual J++ language, you can use it to create new .NET applications or easily migrate existing code to the new C# language using automated migration tools.

Microsoft partners with many companies to bring their languages to the .NET Platform. In addition to the languages provided by Microsoft, there are many alternative languages that target the .NET Platform, including:

- COBOL for Microsoft .NET.
- Perl for Microsoft .NET.
- Eiffel for Microsoft .NET.
- Python for Microsoft .NET.
- Pascal for Microsoft .NET.
- Mercury for Microsoft .NET.
- Mondrian for Microsoft .NET.
- Oberon for Microsoft .NET.
- Salford FTN95 (Fortran) for Microsoft .NET.
- SmallTalk for Microsoft .NET.
- Standard ML for Microsoft .NET.
- Dyalog APL for Microsoft .NET.

Hands out are prepared by

Saif Ur Rehman

Assistant Professor

University Institute of Information Technology

PMAS Arid Agriculture University

Murree Road, Rawalpindi

More Hands out on ASP.Net with C#, PHP with MySQL, Java, Programming Fundamentals and much more can be found at

www.aspuiit.blogspot.com

Suggestions are welcome at

Saif at uaar dot edu dot pk

Software's required to complete this tutorial successfully

1. MicroSoft Visual Studio 2010 or Higher Version of Visula Studio2010

2. MicroSoft SQL Server2008 or Higher Version of MicroSoft SQL Server2008

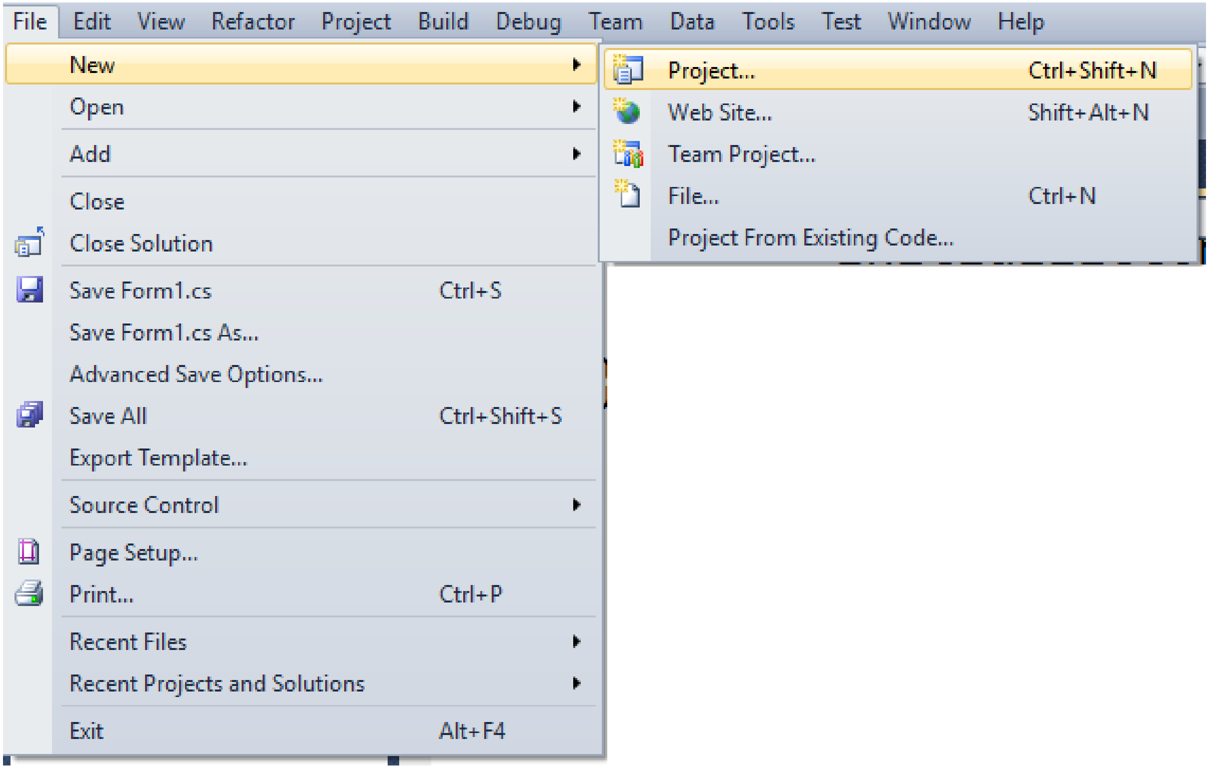
Chapter No.2

Exploring the Visual Studio 2010

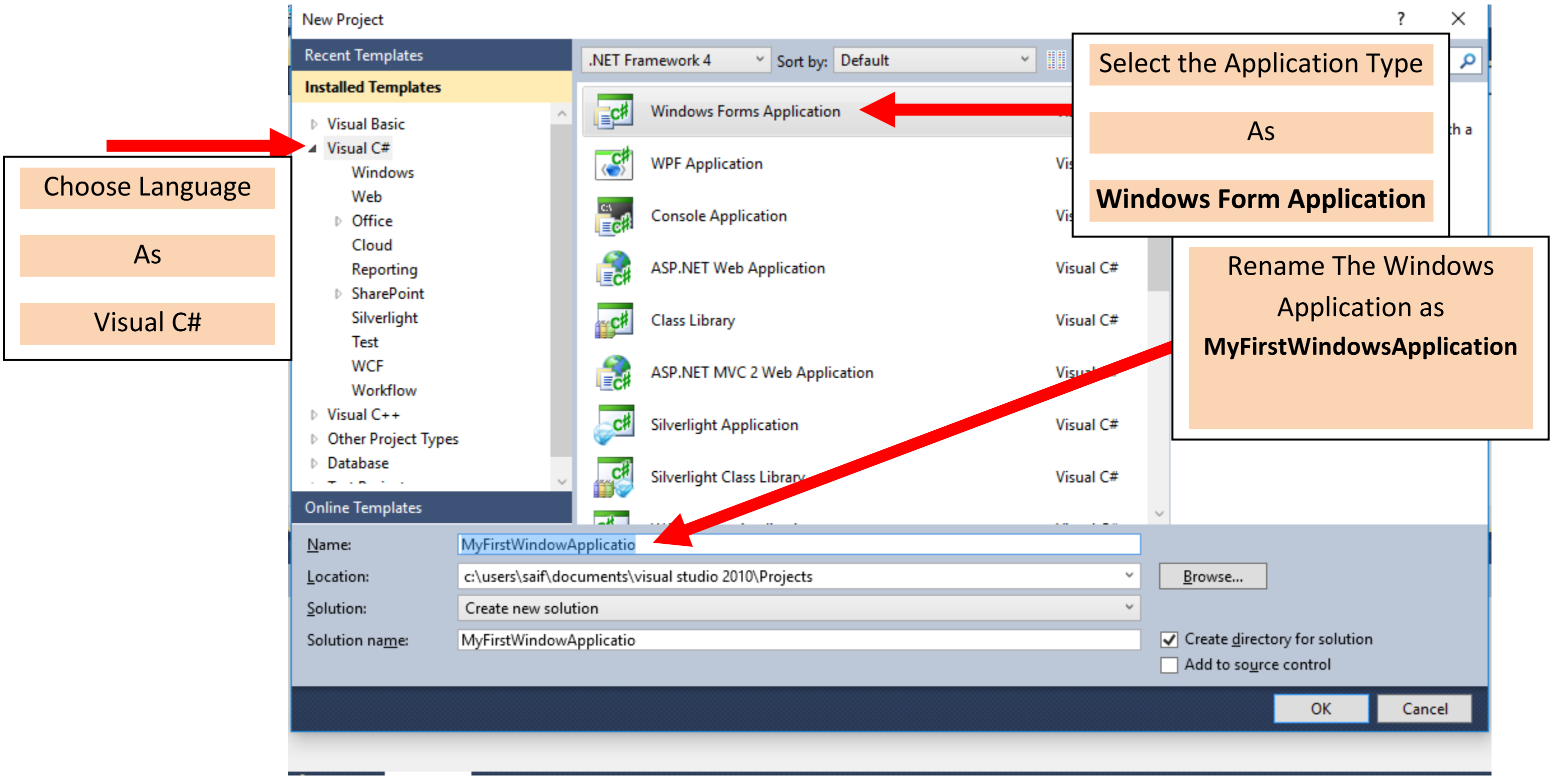
Developing our First C# Windows Form Application

Following the steps to starts a new Window Application

1. Open Visual Studio, then File ➔ New ➔ Project



2. Select Language **Visual C#, Windows Forms Application** and press the OK button as shown in next figure



Choose Language As Visual C#

Select the Application Type As **Windows Form Application**

Rename The Windows Application as **MyFirstWindowsApplication**

New Project

Installed Templates: Visual C#

.NET Framework 4; Sort by: Default

Windows Forms Application

Name: MyFirstWindowApplicatio

Location: c:\users\saif\documents\visual studio 2010\Projects

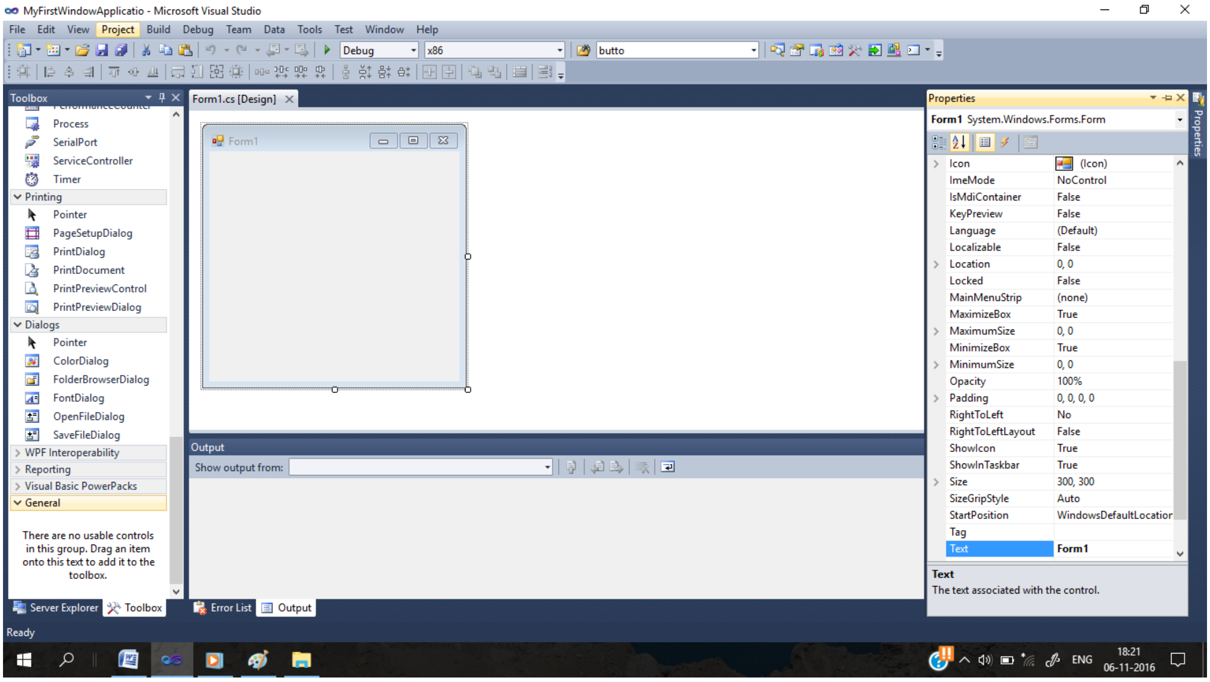
Solution: Create new solution

Solution name: MyFirstWindowApplicatio

☑ Create directory for solution

□ Add to source control

3. You will see the follow,



MyFirstWindowApplicatio - Microsoft Visual Studio

Form1.cs [Design]

Properties: Form1 System.Windows.Forms.Form

Icon	(Icon)
ImeMode	NoControl
IsMdiContainer	False
KeyPreview	False
Language	(Default)
Localizable	False
Location	0, 0
Locked	False
MainMenuStrip	(none)
MaximizeBox	True
MaximumSize	0, 0
MinimizeBox	True
MinimumSize	0, 0
Opacity	100%
Padding	0, 0, 0, 0
RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	300, 300
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	Form1

Text
The text associated with the control.

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

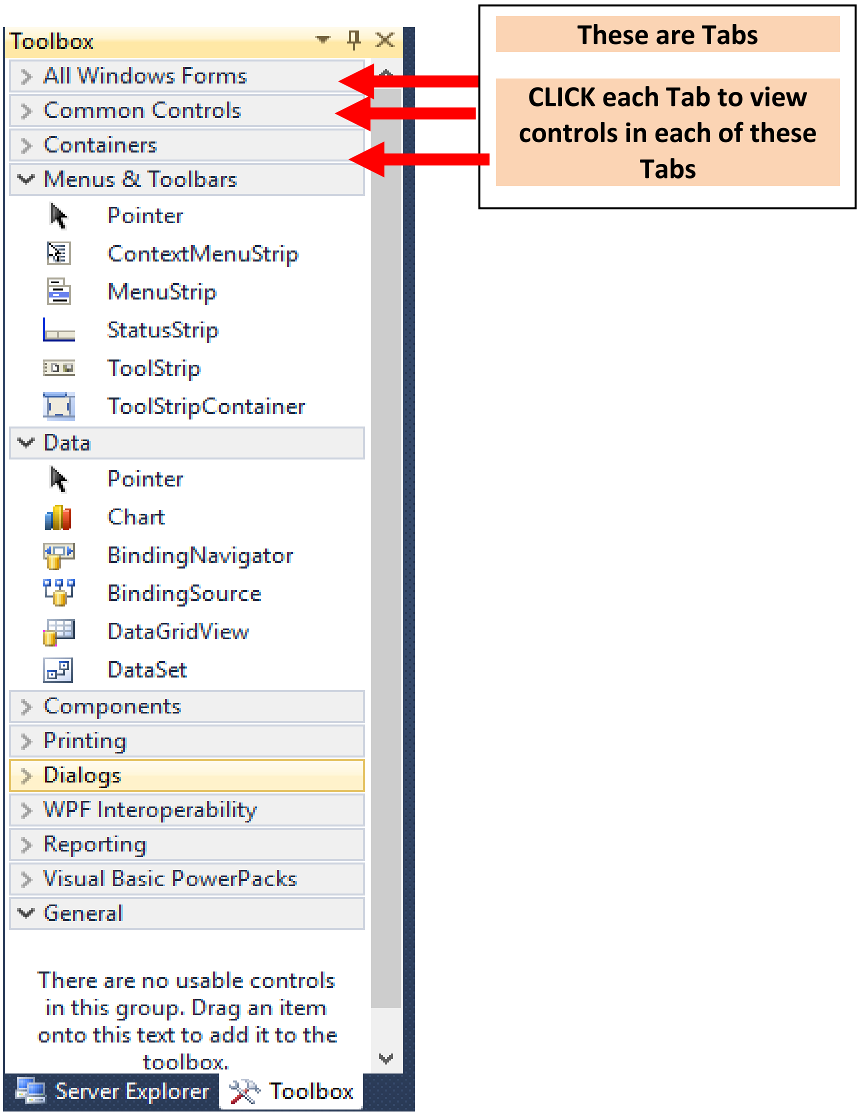
4. In Windows Applications, there are different types of Windows used to facilitate the user to accomplish their programming problems. These Windows are included

   a. ToolBox Windows

   In windows application, we use different controls such as Buttons, Text Boxes, Picture Boxes etc. These controls are listed in the ToolBox in Tabs. ToolBox window can be displayed using any one of the following method.

   1. ALT + CTRL + X
   2. View ➔ ToolBox (or View ➔ Other Windows ➔ ToolBox)

   ToolBox is shown here:



There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

These are Tabs

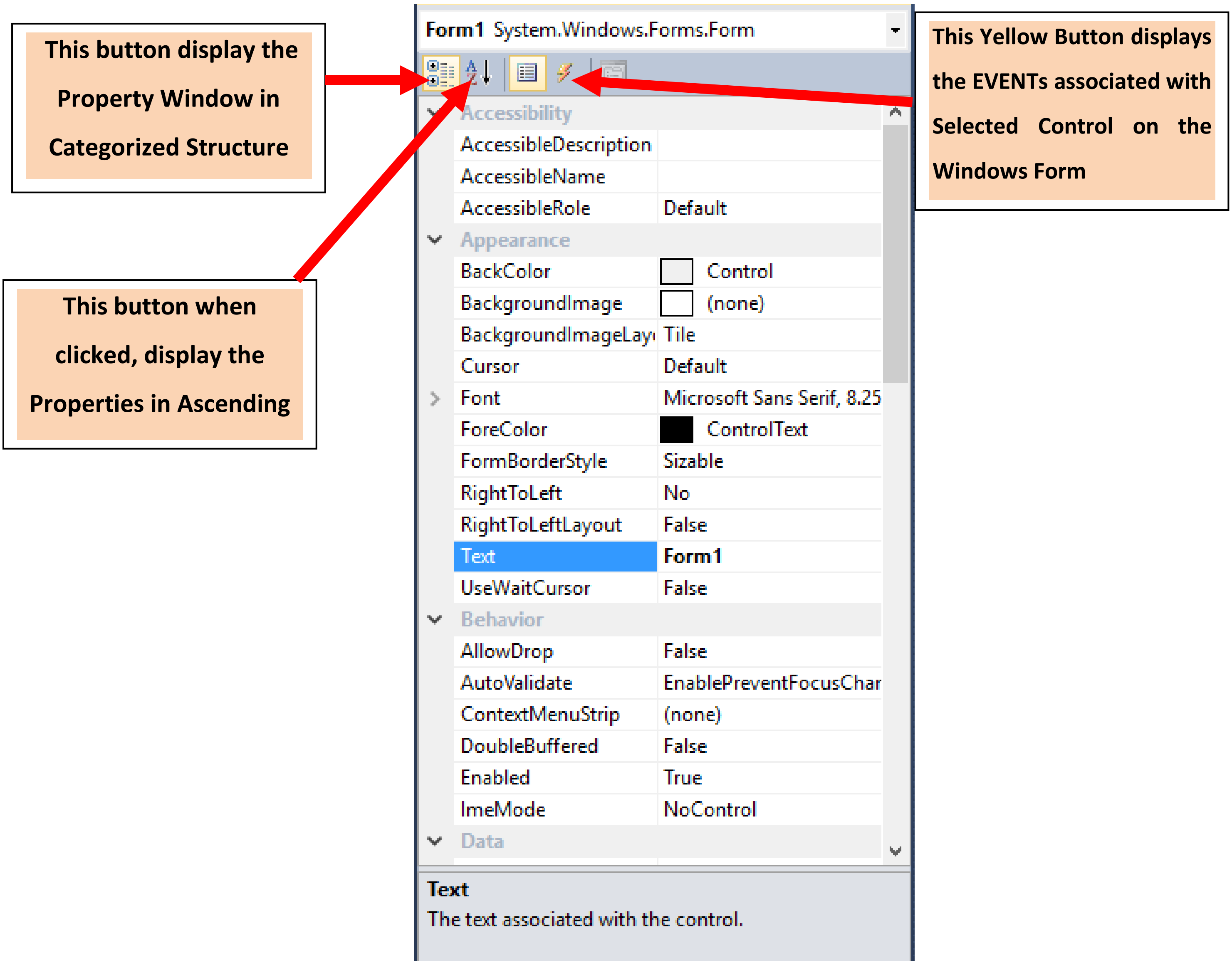
CLICK each Tab to view controls in each of these Tabs

b. Property Windows

In Windows, each control has a predefined set of properties. These properties are used to change the look and feel of the specified control. These properties are displayed in the Property window. Property window can be displayed using any of the following method

1. Right Click on the Control, Select Properties
2. View ➔ Property Window
3. Press F4 button

Property Window is shown in the next figure. In Property Window, there are different options, these are shown below



This button display the Property Window in Categorized Structure

This button when clicked, display the Properties in Ascending

This Yellow Button displays the EVENTs associated with Selected Control on the Windows Form

Form1 System.Windows.Forms.Form

Accessibility	
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	□ Control
BackgroundImage	□ (none)
BackgroundImageLay	Tile
Cursor	Default
Font	Microsoft Sans Serif, 8.25
ForeColor	■ ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	**Form1**
UseWaitCursor	False
Behavior	
AllowDrop	False
AutoValidate	EnablePreventFocusChar
ContextMenuStrip	(none)
DoubleBuffered	False
Enabled	True
ImeMode	NoControl
Data	

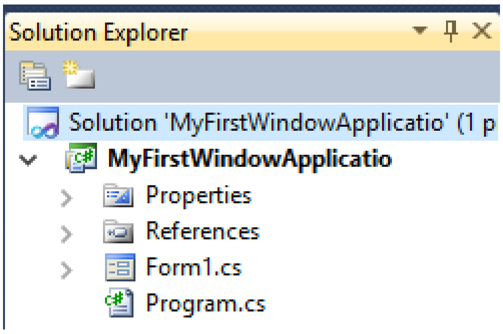
Text
The text associated with the control.

c. **Solution Explorer**

Solution Explorer Window is used to display all the files of the Project including Windows Forms, MDI Parent Forms, Images, Audios, Videos etc. Solution Explorer can be displayed using any one of the following method

1. ALT + CTRL + L
2. View ➔ Solution Explorer

Solution Explorer Window is shown below.



d. **Error List**

Error List Window is used to display all the Errors found in the code. This can be displayed using

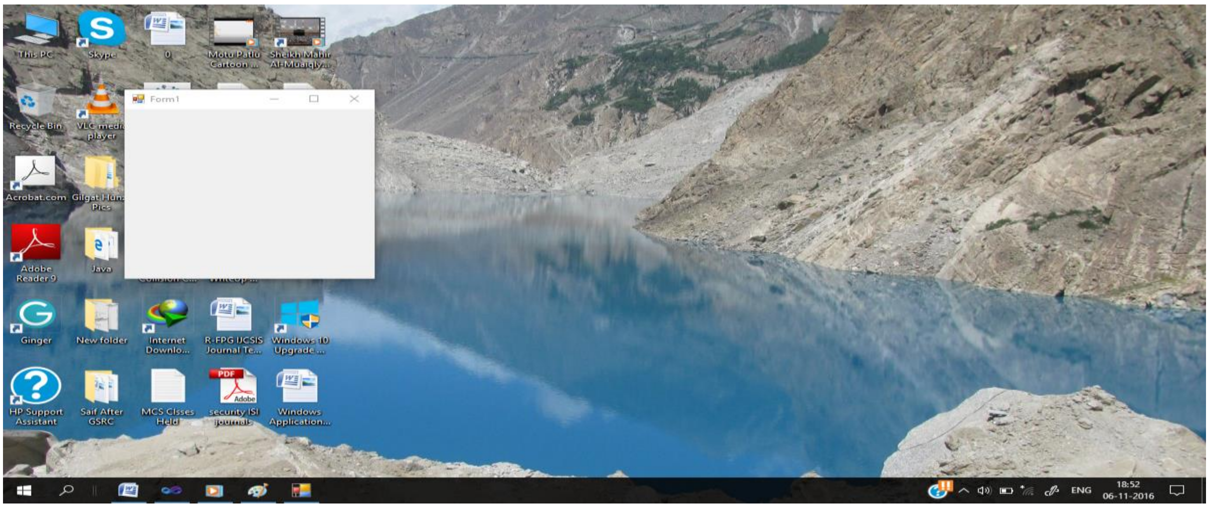
1. View ➔ Error List

e. **Server Explorer**

Server Explorer is used to create SQL Server Databases. We will be discussing Server Explorer in details in the coming days.

5. To run the Windows Form Application, use one of the following method
   1. Press the F5 button
   2. From menu, Choose **Debug ➔ Start Debug**
   3. You can run the form using the Start Debug Arrow in the Toolbar (this Arrow is shown in Green Colour in the Tool bar**)**

Here, you can run the form and see how the application runs and shows the form at run time.



Form1

Manipulating Controls Important Properties

When you start a new Windows Form Application using C# language, then it will create a Form1 for your application. You can see the **Form1.CS** in the Solution Explorer. Initially, the Form is empty. We can add as many controls on the Form as needed by our application.

Form Control has different properties. These properties are shown in the Property Window. For example,

- **Name Property**

Each control is assigned a name. To change the name of the control Name property isused. This propert is shown at the top in the property Window. Some good programming practices suggest the following prefix for different controls while chananging their names.

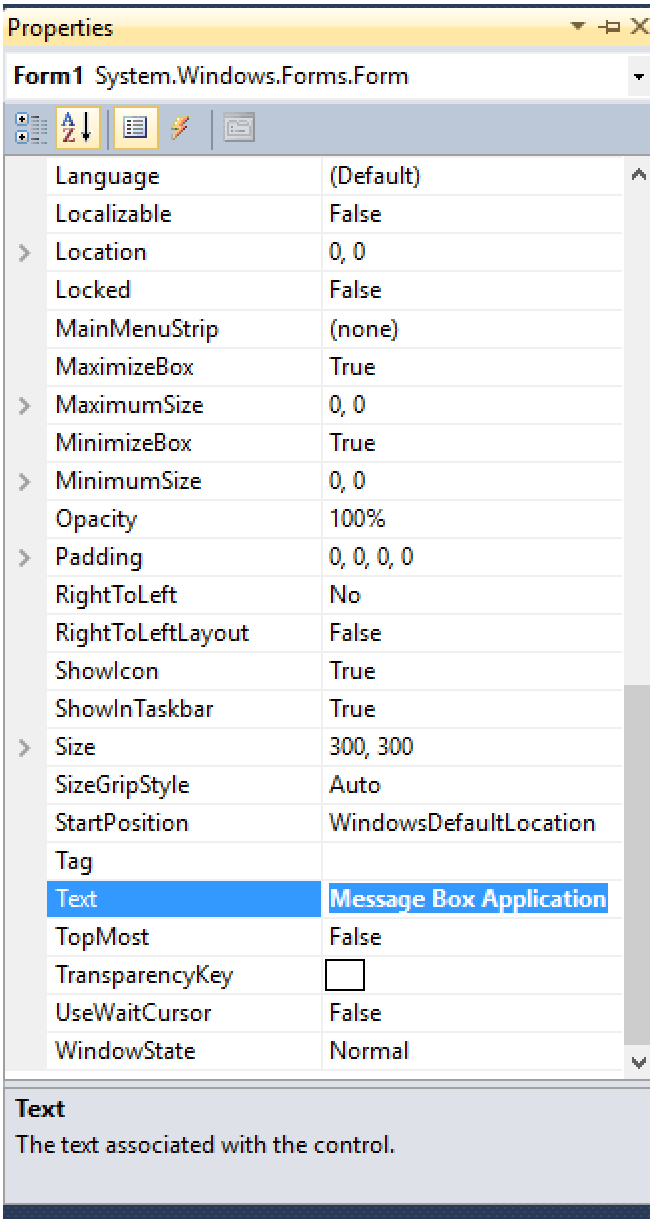
S#	Control Name	Prefix	Sample Control Name
1	Form	frm	frmEmployeeSearch
2	Button	btn	btnAddEmployee
3	TextBox	txt	txtEmployeeSalary
4	Label	lbl	lblEmployeeCNIC
5	DataGridView	gv	gvEmployeeQualificaitons
6	Radio Button	rdb	rdbMale
7	ListView	lv	lvCountry
8	DateTimePicker	dtp	dtpDateOfBirth
9	CheckBox	ckh	chkHoppyCricket
10	ComboBox	cbo	cboDepartments
11	PictureBox	pb	pbEmployeeImpage

Although, it is not necessary to follow the naming conventions mentioned in the above table but these are good pragamming practices to follow. We can also notice that the prefix of the control is written in small case and the first character of all other words in the control name must be given in the Capital letter. Some, example names for the different controls are also given in the above table.

- **Form Caption/Title**

By default, the Form1 has the Title/Caption as Form1. We can change this caption using the Text Property in the Property Window, for example change the Text Proerty to

Message Box Application



Properties

Form1 System.Windows.Forms.Form

Language	(Default)
Localizable	False
Location	0, 0
Locked	False
MainMenuStrip	(none)
MaximizeBox	True
MaximumSize	0, 0
MinimizeBox	True
MinimumSize	0, 0
Opacity	100%
Padding	0, 0, 0, 0
RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	300, 300
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	Message Box Application
TopMost	False
TransparencyKey	□
UseWaitCursor	False
WindowState	Normal

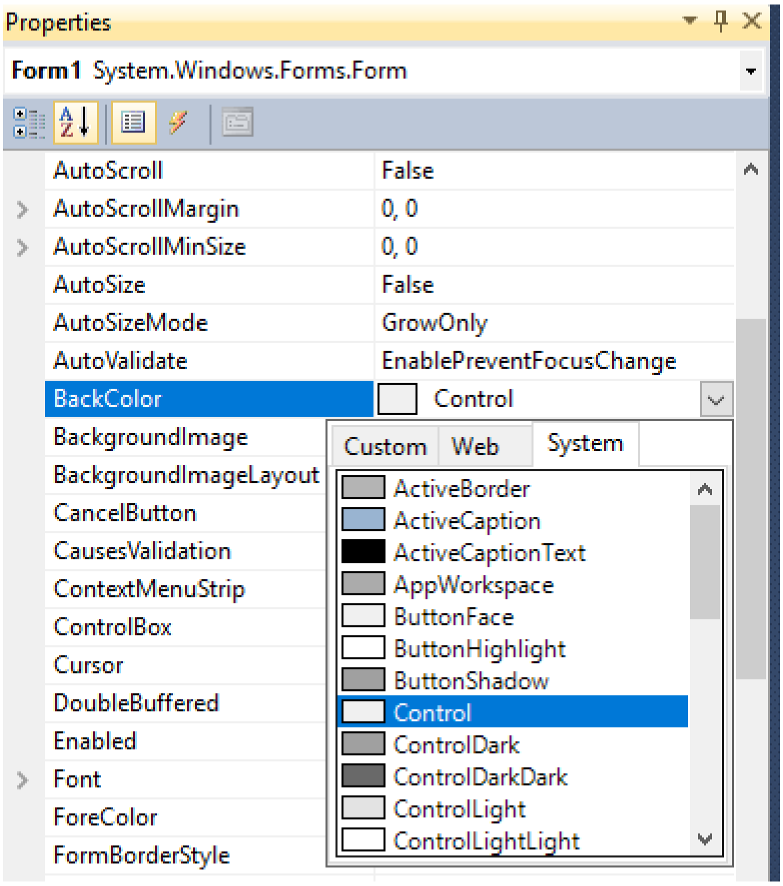
Text

The text associated with the control.

This will change the Caption of the Form1. You can run the Form1 and you can see that the Title of the Form1 is changed to “Message Box Application”.

- **BackColor Property**

BackColor Property is used to change the BackColor of the Controls used in your Windows Applicatioms. For example to change the BackColor of the Form Control, Select the Form Control and Press F4 to display the Property Windows, then search for the BackColor Property. You can see that there are three different options available in the form of Tabs to change the Color of the Controls used in your Windows Applicaiton. This is shown in the next figure:



Properties

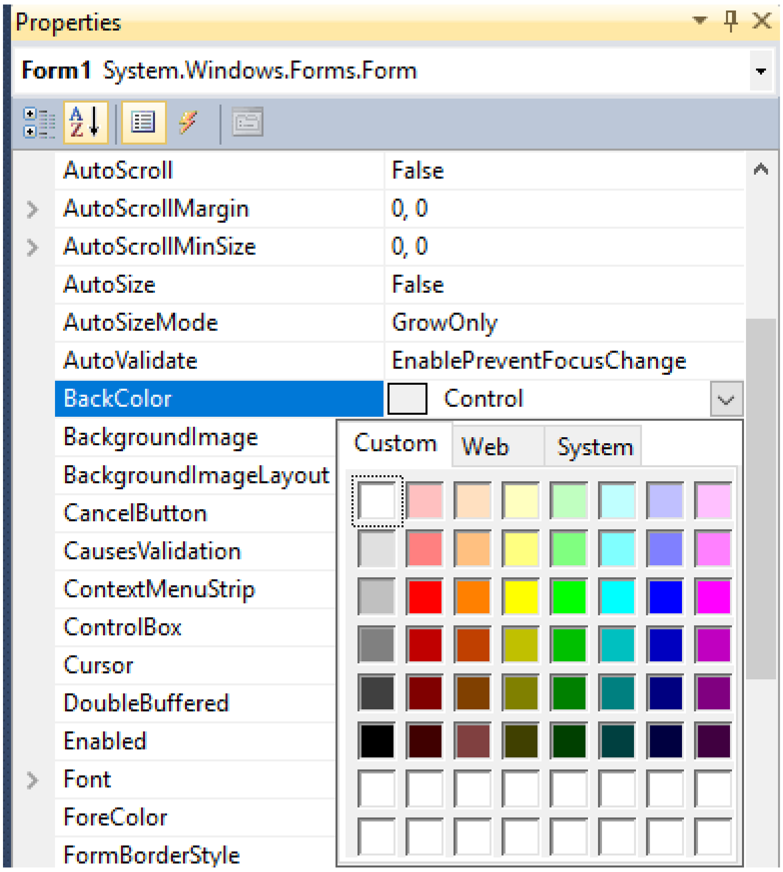
Form1 System.Windows.Forms.Form

AutoScroll	False
AutoScrollMargin	0, 0
AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
AutoValidate	EnablePreventFocusChange
BackColor	Control
BackgroundImage	
BackgroundImageLayout	
CancelButton	
CausesValidation	
ContextMenuStrip	
ControlBox	
Cursor	
DoubleBuffered	
Enabled	
Font	
ForeColor	
FormBorderStyle	

Custom Web System

ActiveBorder
ActiveCaption
ActiveCaptionText
AppWorkspace
ButtonFace
ButtonHighlight
ButtonShadow
Control
ControlDark
ControlDarkDark
ControlLight
ControlLightLight

Here,you can Select Custom Tab, Web Tab or System Tab to check the different colors available. For example, here we open the Custom Tab and choose the BackColor of the Form as shown



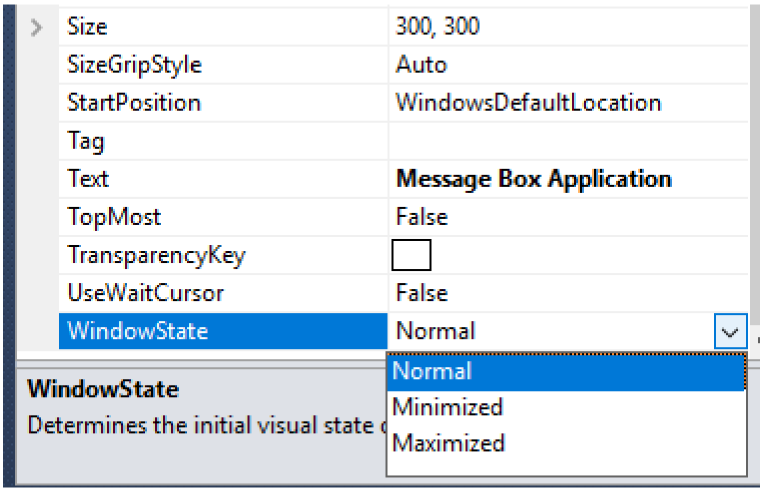
Form1 System.Windows.Forms.Form

AutoScroll	False
AutoScrollMargin	0, 0
AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
AutoValidate	EnablePreventFocusChange
BackColor	Control
BackgroundImage	
BackgroundImageLayout	
CancelButton	
CausesValidation	
ContextMenuStrip	
ControlBox	
Cursor	
DoubleBuffered	
Enabled	
Font	
ForeColor	
FormBorderStyle	

Now, you can see that the BackColor of the Form has been changed. This way, you can change the BackColor of any other Control used on the Form by modifying the BackColor Property.

- **WindowState Property**

This property is available for FORM control only. Using WindowState property, you can change the Size of the FORM at runtime. This propert is shown here



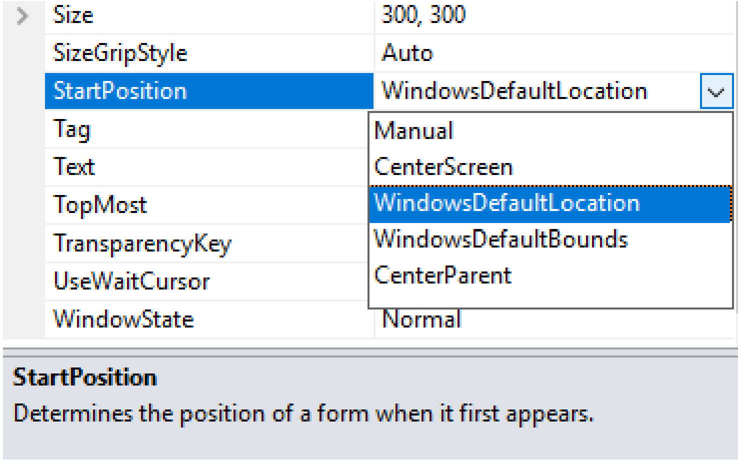
Size	300, 300
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	**Message Box Application**
TopMost	False
TransparencyKey	□
UseWaitCursor	False
WindowState	Normal

WindowState
Determines the initial visual state

Select the Maximized Value and run the form, you will see that FORM has covered the full Screen.

- **StartPosition Property**

This property is also available for FORM control only. This propert has different options which helps users to control the Position of the FORM when displayed during application running. Differen options of StartPostion Property are shown here



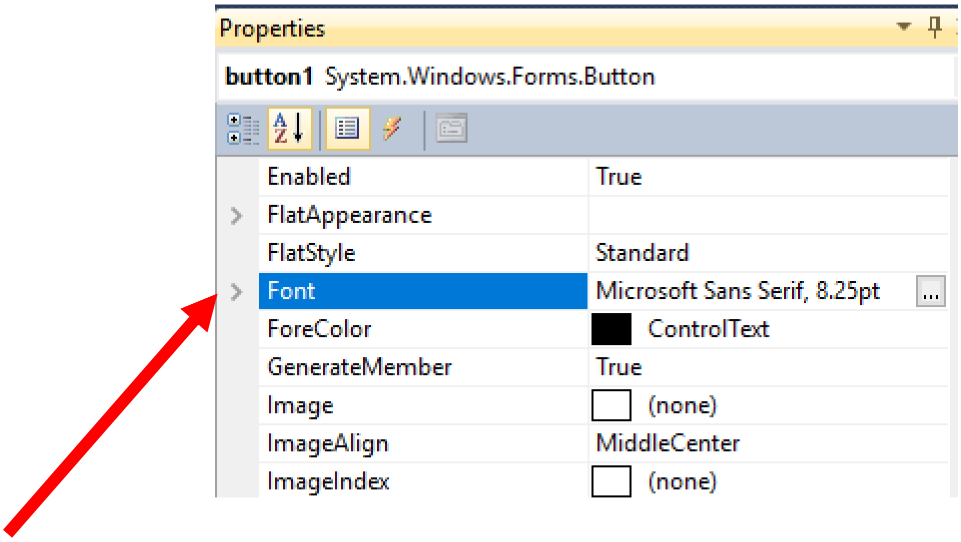
Size	300, 300
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	
TopMost	
TransparencyKey	
UseWaitCursor	
WindowState	Normal

StartPosition
Determines the position of a form when it first appears.

By Selecting the option CentreScreen value for the StartPostion property, the FORM will be displayed in the centre of your computer screen.

- **Font Property Collection**

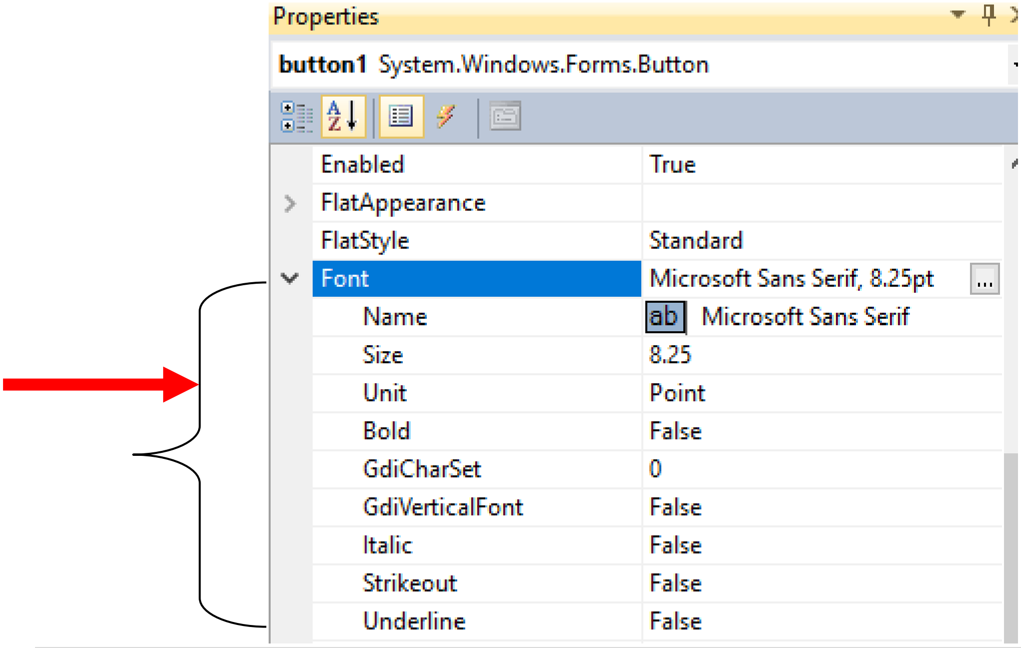
Font propert collection has different properties to modify the font of the text used with different controls, for example to bold the Text on the button Font property collection is used. Similarly, you can change the font family, font size and other font related properties in the Font property collection. Ths Font property collection is shown here (First Drag and Drop a button Control on the FORM from ToolBox, open the property Window for the Button control and look for the Font property)



button1 System.Windows.Forms.Button

Enabled	True
FlatAppearance	
FlatStyle	Standard
Font	Microsoft Sans Serif, 8.25pt
ForeColor	■ ControlText
GenerateMember	True
Image	□ (none)
ImageAlign	MiddleCenter
ImageIndex	□ (none)

As indicated by the arrown, Click on the small Arrow in property Window as shown above, you an see that different properties under the Font Collection is displayed



button1 System.Windows.Forms.Button

Enabled	True
FlatAppearance	
FlatStyle	Standard
Font	Microsoft Sans Serif, 8.25pt
Name	ab Microsoft Sans Serif
Size	8.25
Unit	Point
Bold	False
GdiCharSet	0
GdiVerticalFont	False
Italic	False
Strikeout	False
Underline	False

When you click on Font, these different sub properties of Font are displayed including

Name

Size

Bold

etc

For example, to Bold the Text of Button and Increase the size of text to 14Pts

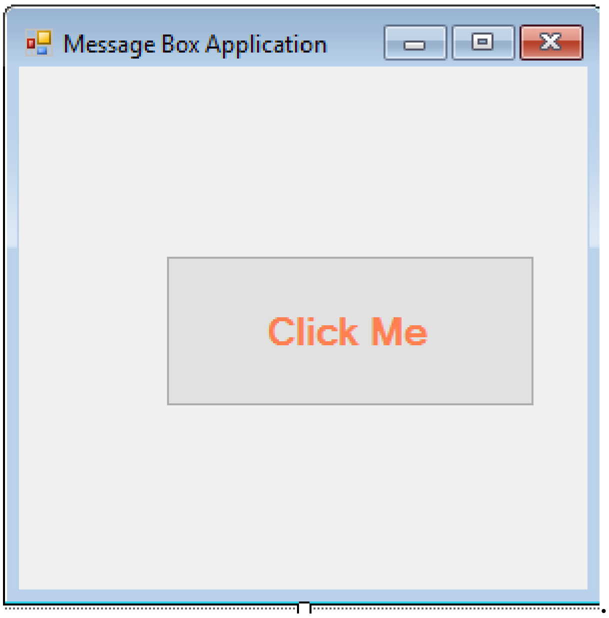
Bold : True

Size : 14Pts

You will see that the Font Property of the button control are changed.

- **ForeColor Property**

This property is used to change the Text Color. For example, to change the text color of the button control, goto ToolBox and look for the ForeColor Property and set the ForeColor value as you like the Color. Run the form , you will see the output as shown



More properties will be discussed as we explore the more controls in the next lectures

Using Text boxes, Buttons, Labels and MessageBox

TextBox is most commonly used control in the windows and web applications. Normally, TextBox is used as an input control. The input entered in TextBox is in the String data type.

Following syntax is used to retrieve the text entered in the TextBox

```
string stdName = textBox1.Text;
```

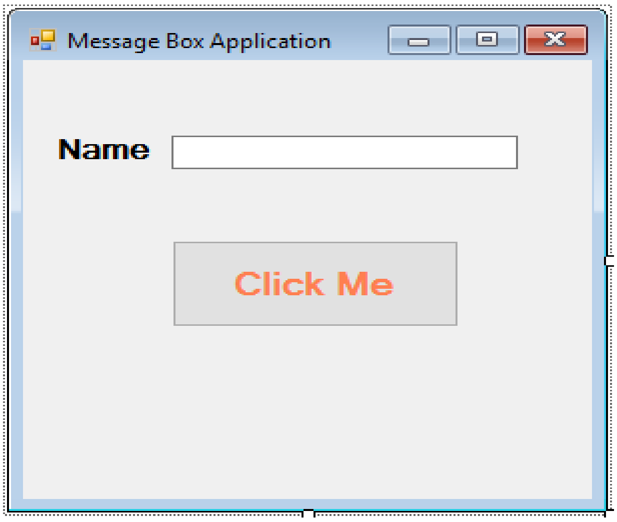
Button control is used to activate the Click event. Like TextBox, Button control is also commonly used control.

Label Control is used for labelling the TextBoxes. Text property is used to change the label for the Label control

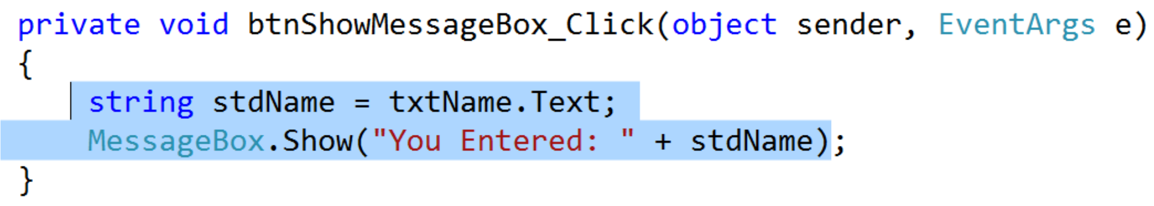
A **MessageBox** is used to display the messages to the users. A MessageBox is displayed using the following syntax

```
MessageBox.Show("Wel Come To C#");
```

For example, here is a simple Form with a TextBox, a Label and a Button. When we press the Button then a Message Box will be open showing the text entered in the TextBox. Designed from is



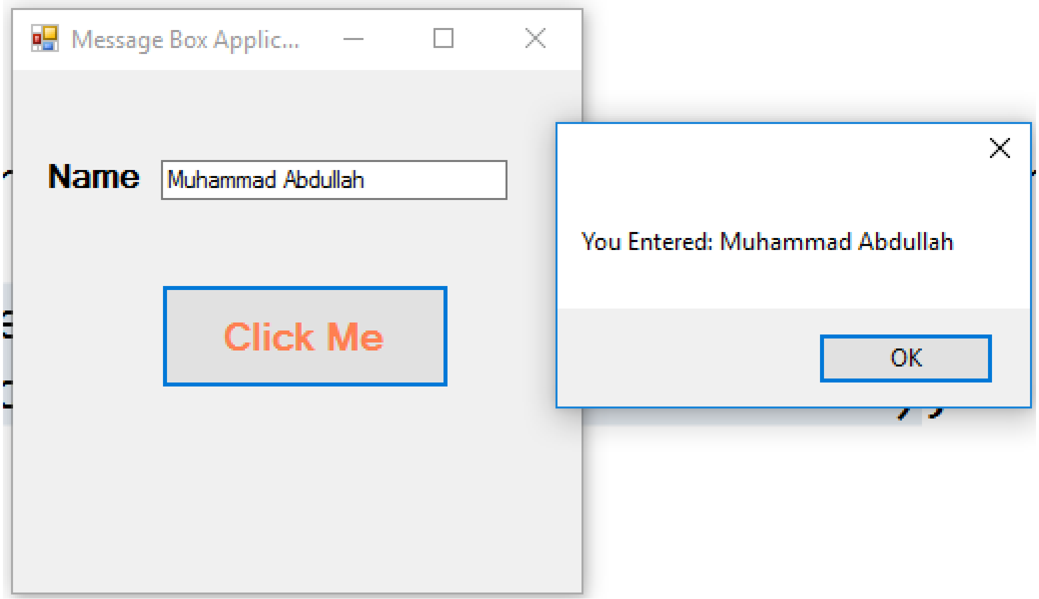
Double click on the button to open the code window for th Click Event of the button. An event is the action performed by the user. For example Click is an event which is fired when user clicks on the button at runtime



```
private void btnShowMessageBox_Click(object sender, EventArgs e)
{
    string stdName = txtName.Text;
    MessageBox.Show("You Entered: " + stdName);
}
```

.

Here, first we get the text entered in the string variable stdName and then in the MessageBox, we concatenate the text by using “+” operator. Run the application and see the result.



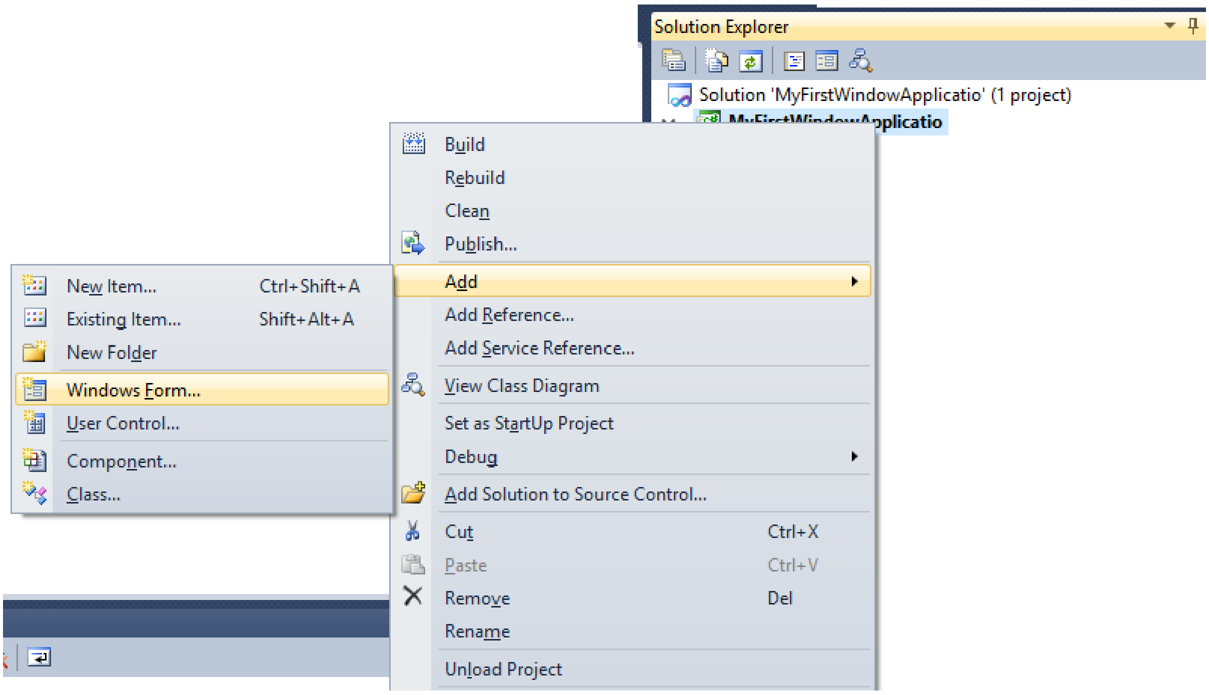
Name: Muhammad Abdullah

You Entered: Muhammad Abdullah

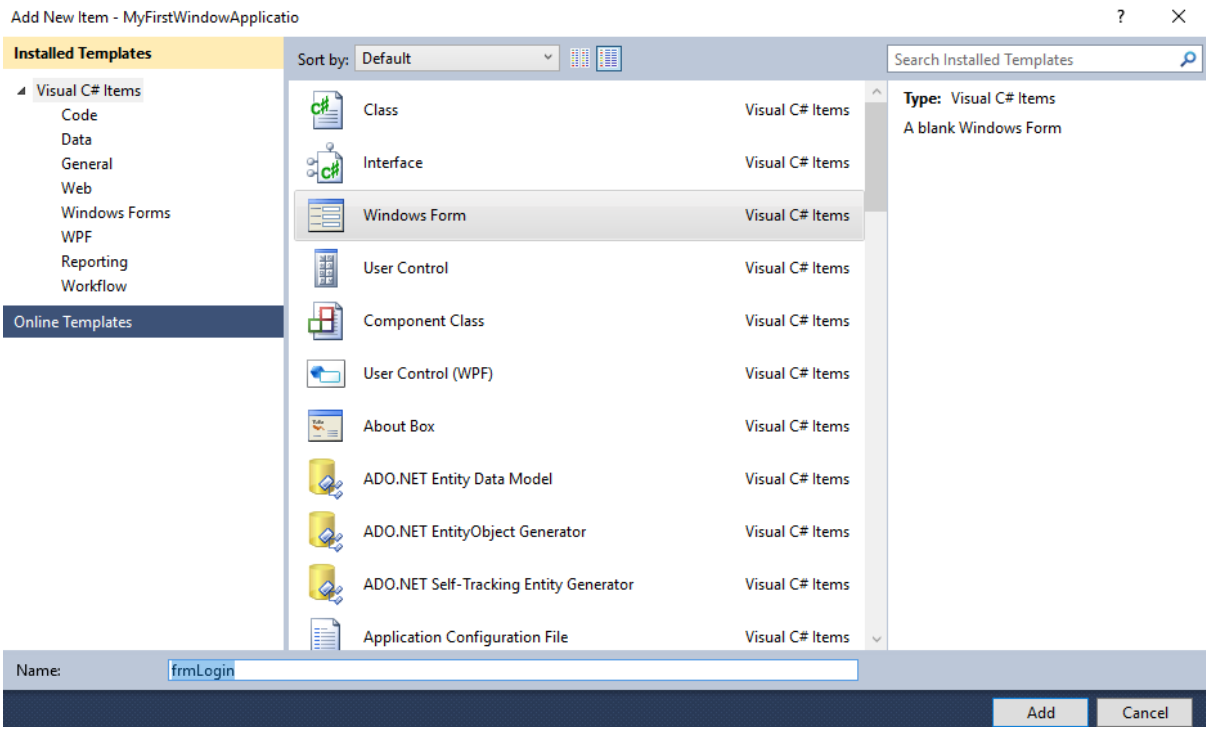
Congratulations! You have successfully implemented the Buttons, TextBox, Label and MessageBox controls.

A Sample Login Page

Add a new Window Form by Right Click on the Project in the Solution Explorer Window and rename it as **frmLogin**



Renaming the Windows Form



Sort by: Default

Class	Visual C# Items
Interface	Visual C# Items
Windows Form	Visual C# Items
User Control	Visual C# Items
Component Class	Visual C# Items
User Control (WPF)	Visual C# Items
About Box	Visual C# Items
ADO.NET Entity Data Model	Visual C# Items
ADO.NET EntityObject Generator	Visual C# Items
ADO.NET Self-Tracking Entity Generator	Visual C# Items
Application Configuration File	Visual C# Items

Type: Visual C# Items

A blank Windows Form

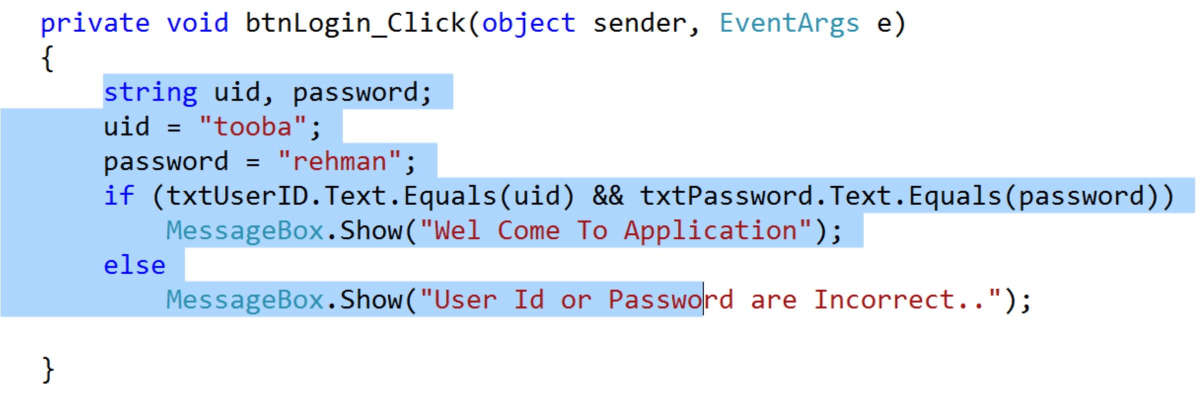
Name: frmLogin

Now, design the form as shown below

For the TextBox used for Password set the **PasswordChar** property to *****

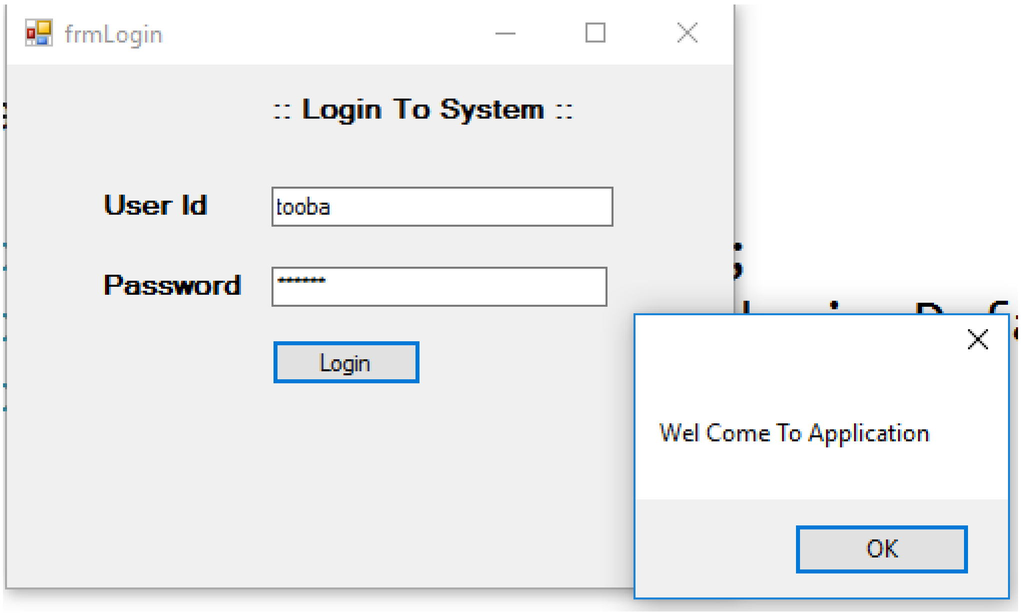
PasswordChar property is used to define th character which will be used for the password character**,**

Double click on the button to define the Click Event as shown. Type the following code



```
private void btnLogin_Click(object sender, EventArgs e)
{
    string uid, password;
    uid = "tooba";
    password = "rehman";
    if (txtUserID.Text.Equals(uid) && txtPassword.Text.Equals(password))
        MessageBox.Show("Wel Come To Application");
    else
        MessageBox.Show("User Id or Password are Incorrect..");

}
```

Here, we are assuming uid and password as static value, afterwards we will be doing this task with database. Run the application



frmLogin

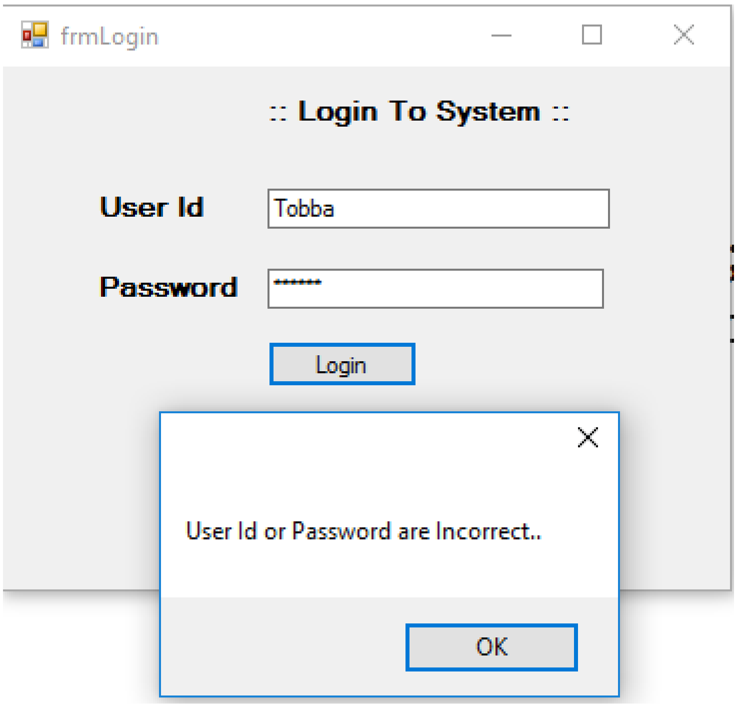
:: Login To System ::

User Id: tooba

Password: ******

Wel Come To Application

When you enter incorrect Uid or Password then following message box will appear

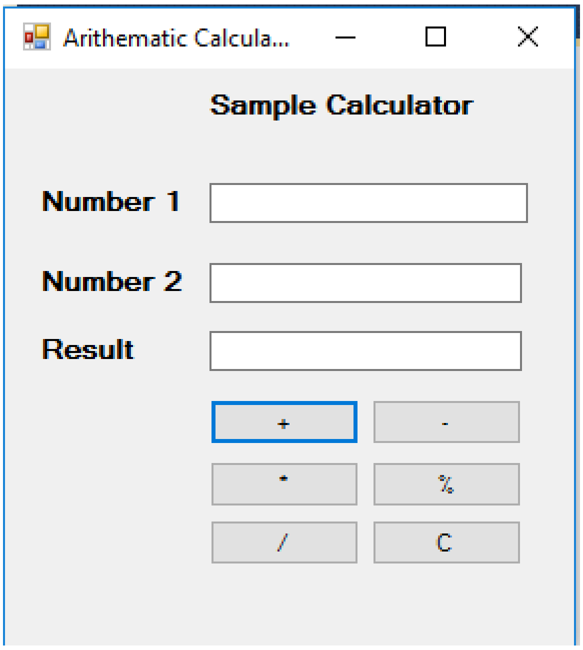


User Id or Password are Incorrect..

A Sample Calculator for Arithmetic Operations

To design a sample Calculator, follow the following steps

1. **Start a** new Windows Form Application and Rename the Application as **SampleCalculator**
2. Design the Form as shown below

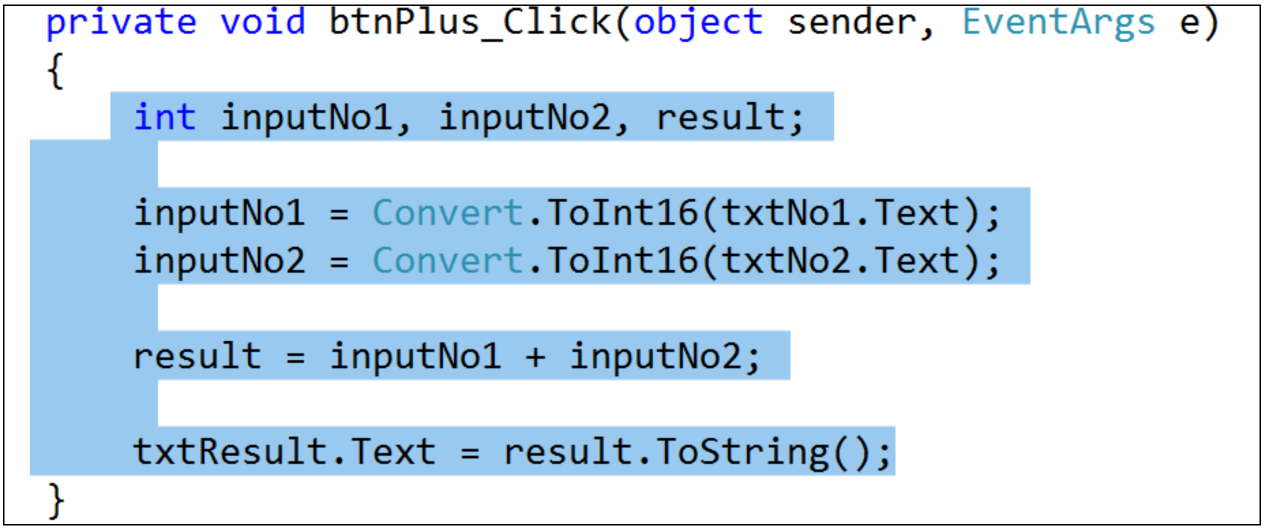


3. Rename the different controls used on the form as follow

S#	Control	Rename as
1	Label1	lblTitle
2	Label2	lblNo1
3	Label3	lblNo2
4	Label4	lblResult
5	TextBox1	txtNo1
6	TextBox2	txtNo2
7	TextBox3	txtResult
8	button1	btnPlus
9	Button2	btnMinus
10	Button3	btnMultiplication

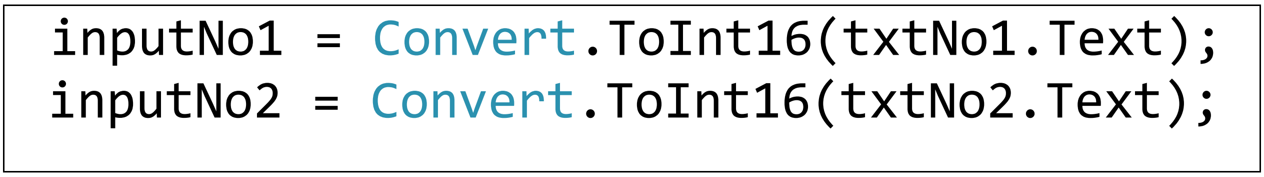
11	Button4	btnRemainder
12	Button5	btnDivision
13	Button6	btnClear

4. Now, double Click on Plus (+) button to create the Click event of btnPlus button and type the following code



```
private void btnPlus_Click(object sender, EventArgs e)
{
    int inputNo1, inputNo2, result;

    inputNo1 = Convert.ToInt16(txtNo1.Text);
    inputNo2 = Convert.ToInt16(txtNo2.Text);

    result = inputNo1 + inputNo2;

    txtResult.Text = result.ToString();
}
```

See the statement,

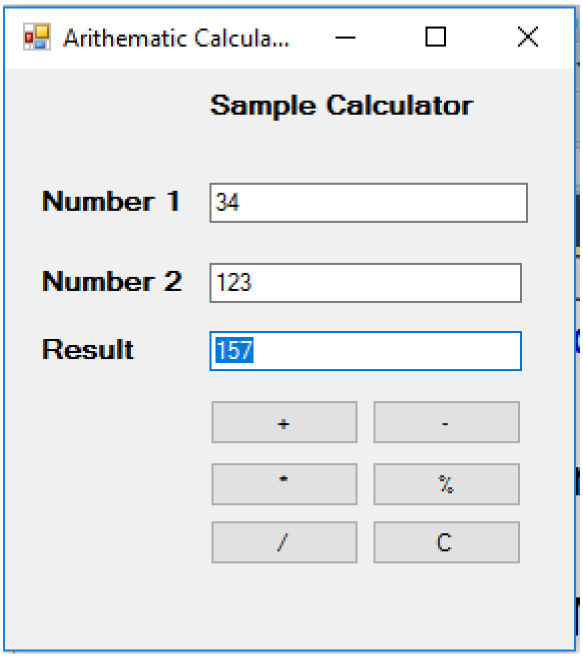


```
inputNo1 = Convert.ToInt16(txtNo1.Text);
inputNo2 = Convert.ToInt16(txtNo2.Text);
```

Here, as we have seen in previous lecture that input entered in the TextBox is in String data type, so therefore we have to convert the TextBox value to int data type before performing any arithematic operations on the TextBox inputs.

Convert.ToInt16(StringValue) is used to convert the String Value to 16-bits signed inter data type values.

Now, run the form and see the output



Sample Calculator

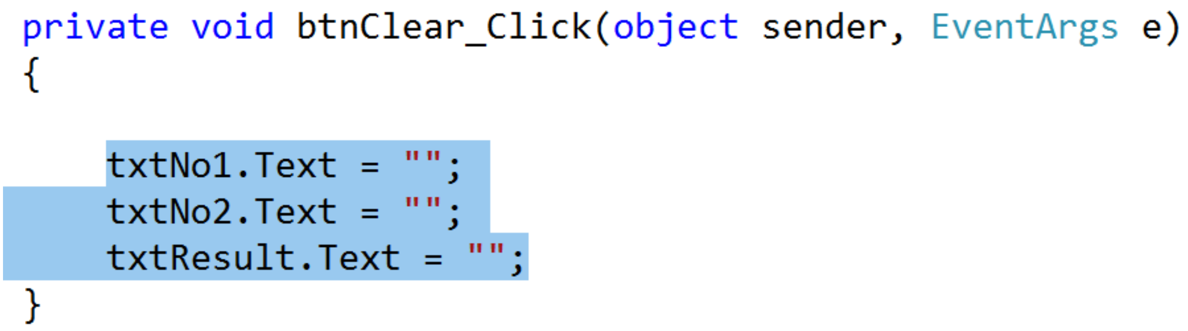
Number 1: 34

Number 2: 123

Result: 157

Finally, you can write the code against Click event of each button. The code will be similar to one given in the Click event of the Plus button, just operator will be replaced with either -, *, /, % .

The coding for C (clear button) is shown here

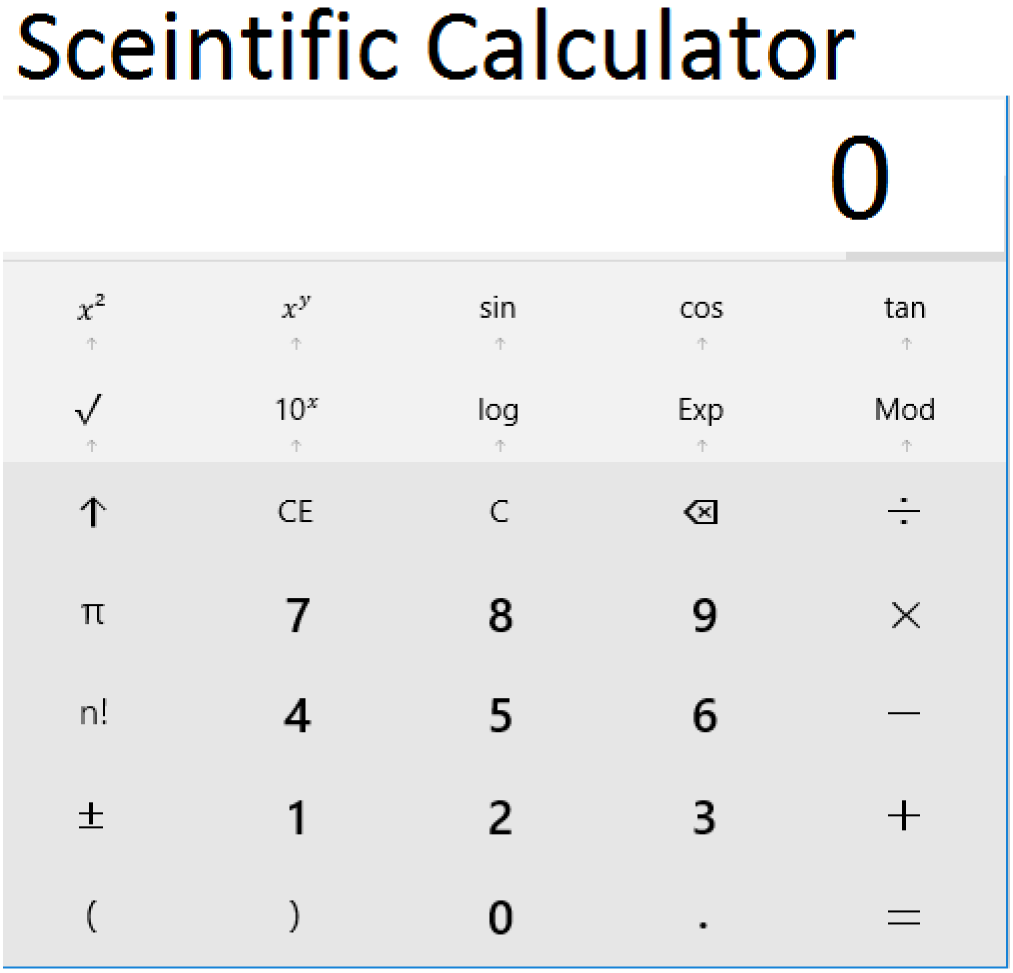


```
private void btnClear_Click(object sender, EventArgs e)
{

        txtNo1.Text = "";
        txtNo2.Text = "";
        txtResult.Text = "";
}
```

Congratulations! You have successfully learnt how to design a Sample Arithmetic Calculator

Assignment:

Design a Scientific Calculator as shown below



Then, write the code against each button

Hands out are prepared by

Saif Ur Rehman

Assistant Professor

University Institute of Information Technology

PMAS Arid Agriculture University

Murree Road, Rawalpindi

More Hands out on ASP.Net with C#, PHP with MySQL, Java, Programming Fundamentals and much more can be found at www.aspuiit.blogspot.com

Suggestions are welcome at

Saif at uaar dot edu dot pk

Chapter No.3

Designing Database in MS SQL Server 2008

Designing Database in MS SQL Server 2008

Before we start working to design database in SQL Server, we briefly overview the SQL Server.

Microsoft SQL Server

Microsoft SQL Server is a Relational Database Management System (RDBMS) designed to run on platforms ranging from laptops to large multiprocessor servers. SQL Server is commonly used as the backend system for websites and corporate CRMs and can support thousands of concurrent users.

SQL Server comes with a number of tools to help you with your database administration and programming tasks.

SQL Server is much more robust and scalable than a desktop database management system such as Microsoft Access. Anyone who has ever tried using Access as a backend to a website will probably be familiar with the errors that were generated when too many users tried to access the database!

Although SQL Server can also be run as a desktop database system, it is most commonly used as a server database system.

Server based database systems

Server based database systems are designed to run on a central server, so that multiple users can access the same data simultaneously. The users normally access the database through an application.

For example, a website could store all its content in a database. Whenever a visitor views an article, they are retrieving data from the database. As you know, websites aren't normally limited to just one user. So, at any given moment, a website could be serving up hundreds, or even thousands of articles to its website visitors. At the same time, other users could be updating their personal profile in the members' area, or subscribing to a newsletter, or anything else that website users do.

Generally, it's the application that provides the functionality to these visitors. It is the database that stores the data and makes it available. Having said that, SQL Server does include some useful features that can assist the application in providing its functionality.

SQL Server 2008 (and Later) Editions

SQL Server 2008 comes in many different editions. The edition you choose will depend on your requirements. If you are looking for a free database management system, you will need to choose one of the Express editions or the Compact edition. You could also try the Evaluation edition, which allows you to trial SQL Server 2008 for 180 days.

The Editions

Here are the different editions available for SQL Server 2008.

- **Enterprise Edition**

Data management and business intelligence platform providing enterprise class scalability, high availability, and security for running business-critical applications

- **Standard Edition**

Data management and business intelligence platform providing ease of use and manageability for running departmental applications

- **Workgroup Edition**

Data management and reporting platform providing secure, remote synchronization, and management capabilities for running branch applications

- **Developer Edition**

May be installed and used by one user to design, develop, test, and demonstrate your programs on as many systems as needed

- **Web Edition**

A low-TCO, scalable, and manageable database option for web hosters and end customers looking to deploy publicly facing web applications and services

- **Express Edition**

A free edition of SQL Server ideal for learning and building desktop and small server applications and for redistribution by ISVs

- **Compact Edition**

A free, SQL Server embedded database ideal for building stand-alone and occasionally connected applications for mobile devices, desktops, and web clients

- **Evaluation Edition**

This edition may be installed for demonstration and evaluation purposes until an expiration period of 180 days.

SQL Server Management Studio (SSMS)

SQL Server Management Studio (SSMS) is the main administration console for SQL Server. SSMS enables you to create database objects (such as databases, tables, views etc), view the data within your database; you can configure user accounts, transfer data between databases, and more.

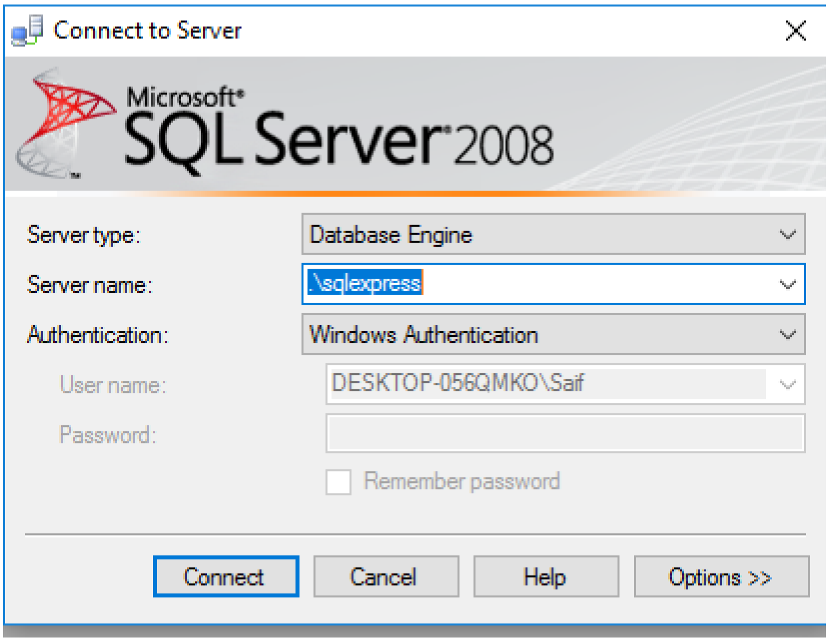
You can use SQL Server Management Studio to create as many databases as you like. You can also connect to as many databases on as many servers as you like.

Most of the tasks performed with SQL Server Management Studio are initiated either from the top menu, or by right-clicking on an icon/object.

The left pane contains the Object Explorer. The Object Explorer provides navigation to databases, server objects (such as triggers), log files, and more.

Follow these steps to design a sample database in MS SQL Server2008 Management Studio

a. Open MicroSoft SQL Server2008 Management Studio


Connect to Server

Server type: Database Engine

Server name: .\sqlexpress

Authentication: Windows Authentication

User name: DESKTOP-056QMKO\Saif

Password:

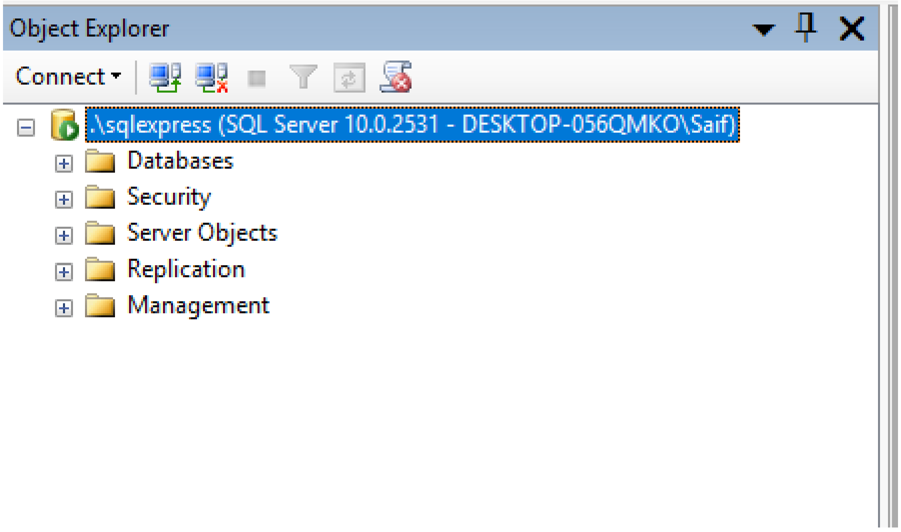
□ Remember password

b. Enter the Server Name and Press the **Connect** button. For this demo, Server Name is

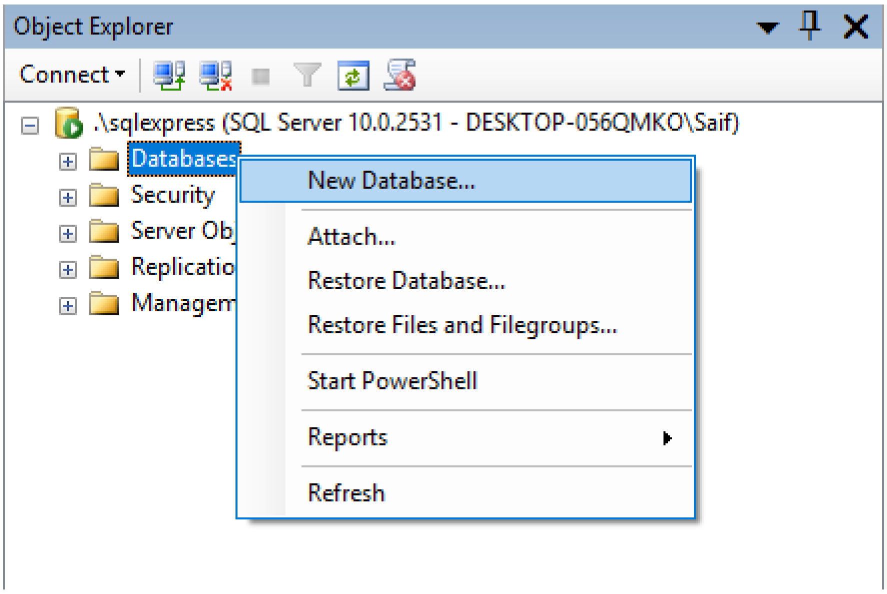
.\SQLEXPRESS

Here, **.** represents the localhost (you can replace . with your computer name). SQLEXPRESS is an instance of the SQL Server2008 installed on the System. There can be many instance of SQL Server on your PC SQLEXPRESS is the default instance name.

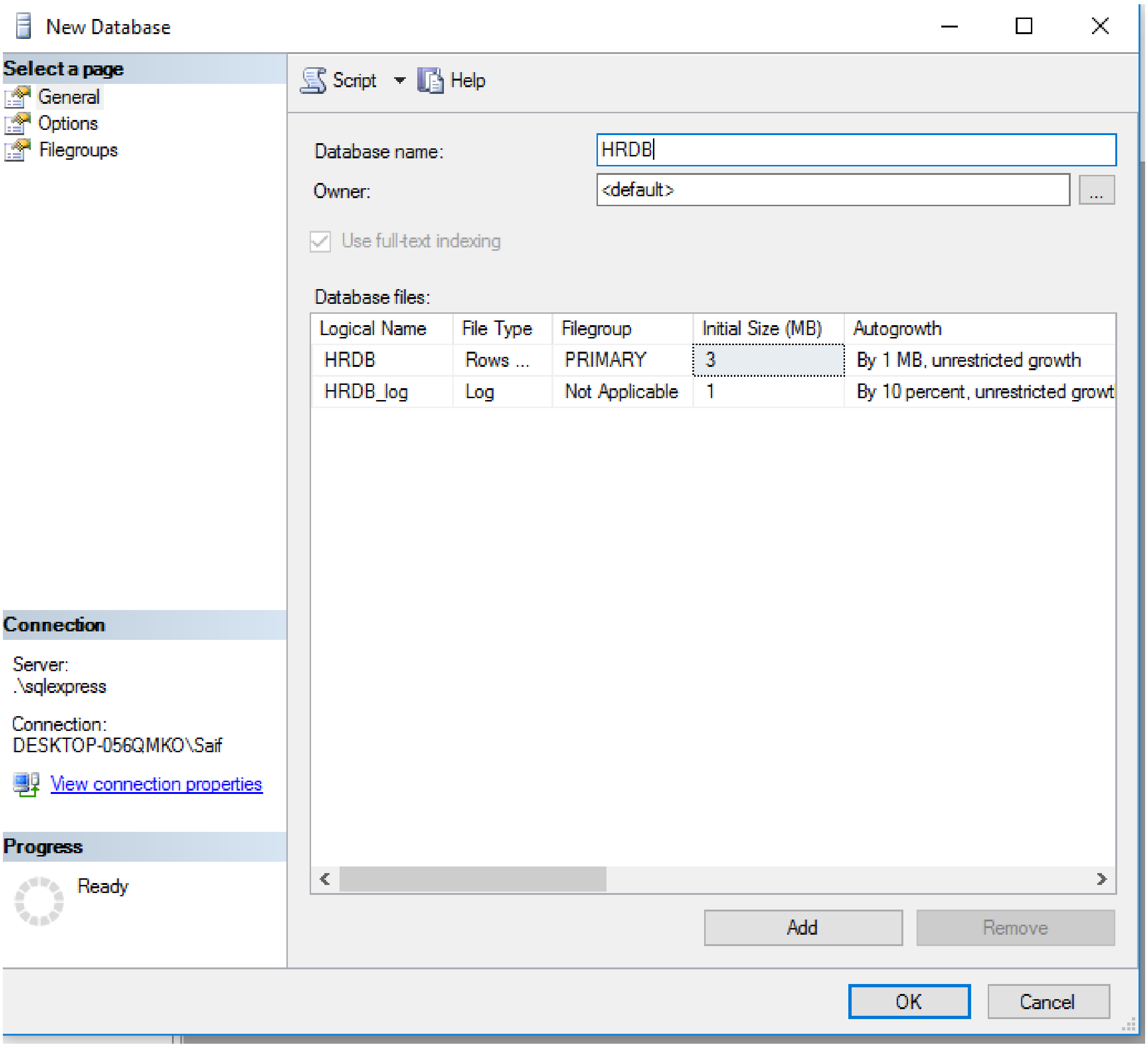
c. After, you press the **Connect** button, you will see the screen as shown



d. Write Click on Database Node and Select New Database Option as shown



e. Enter the database Name as HRDB and press the **OK** button,



Database name: HRDB

Owner: <default>

☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth
HRDB	Rows ...	PRIMARY	3	By 1 MB, unrestricted growth
HRDB_log	Log	Not Applicable	1	By 10 percent, unrestricted growt

Connection

Server:
.\sqlexpress

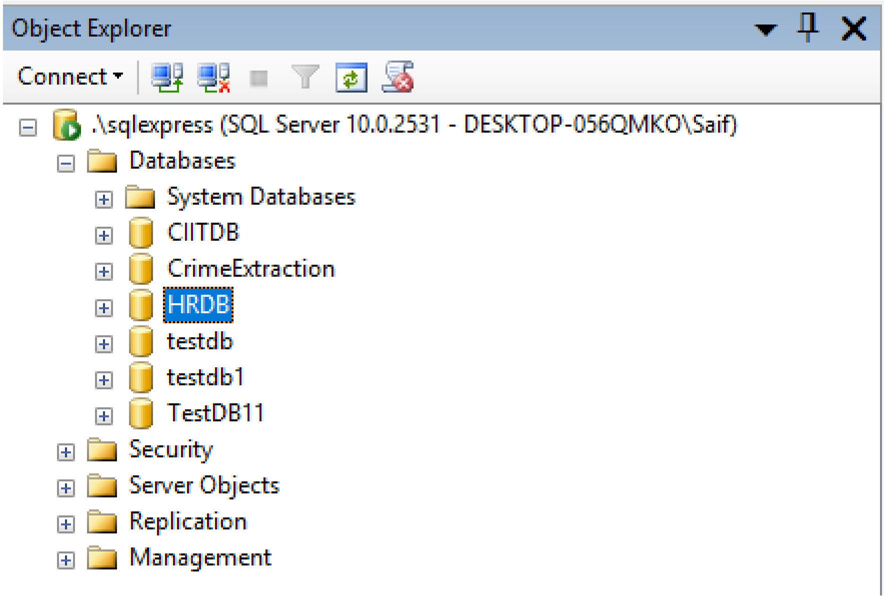
Connection:
DESKTOP-056QMKO\Saif

View connection properties

Progress

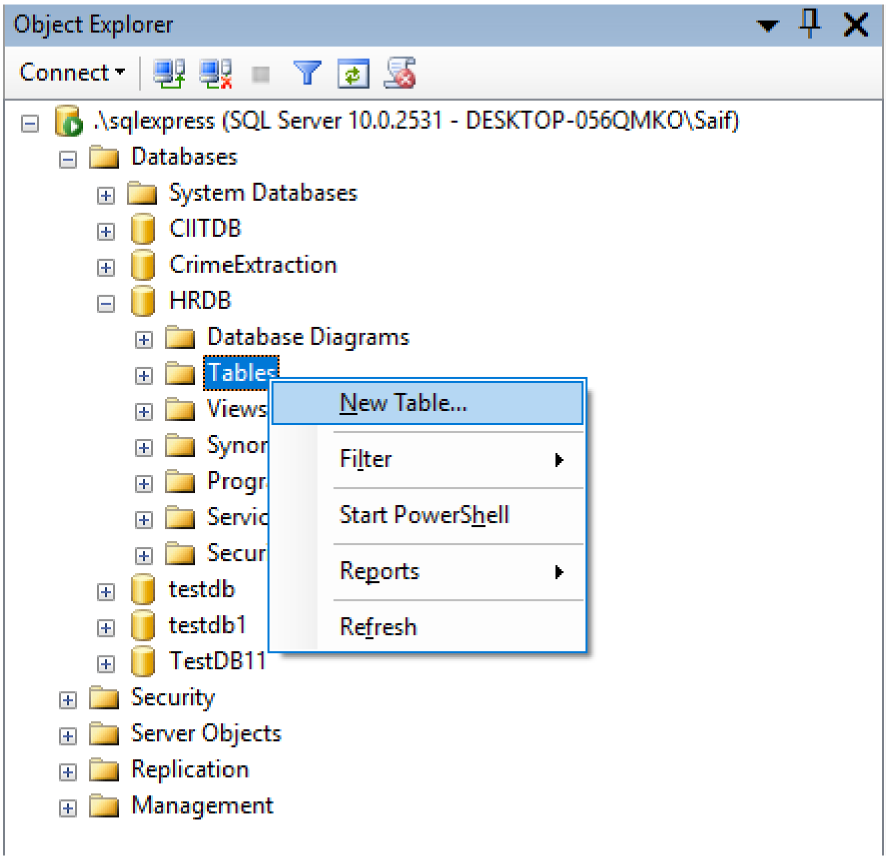
Ready

f. This way your Database with name HRDB will be created as shown

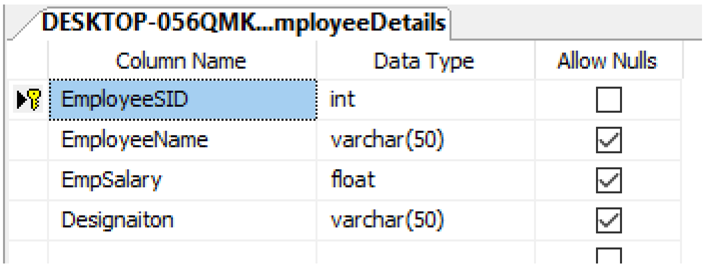


- .\sqlexpress (SQL Server 10.0.2531 - DESKTOP-056QMKO\Saif)
  - Databases
    - System Databases
    - CIITDB
    - CrimeExtraction
    - HRDB
    - testdb
    - testdb1
    - TestDB11
  - Security
  - Server Objects
  - Replication
  - Management

g. Press the + button beside the **HRDB** database, then right click on the Tables Node and choose the New Table Option as shown

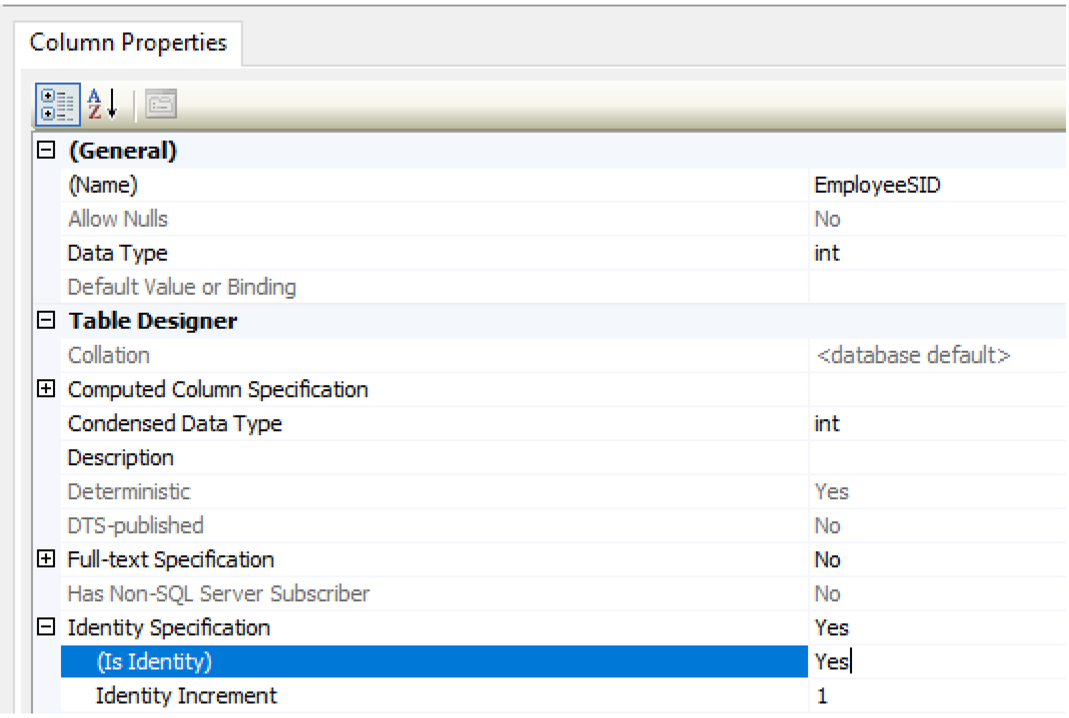


h. Design the Table as shown


DESKTOP-056QMK...mployeeDetails

Column Name	Data Type	Allow Nulls
EmployeeSID	int	□
EmployeeName	varchar(50)	☑
EmpSalary	float	☑
Designaiton	varchar(50)	☑
		□

Set the EmployeeSID as Primary Key and set EmployeeSID as Auto Increment. Primary key value can be auto incremented by setting the (IsIdentity property to YES). This is shown as follow

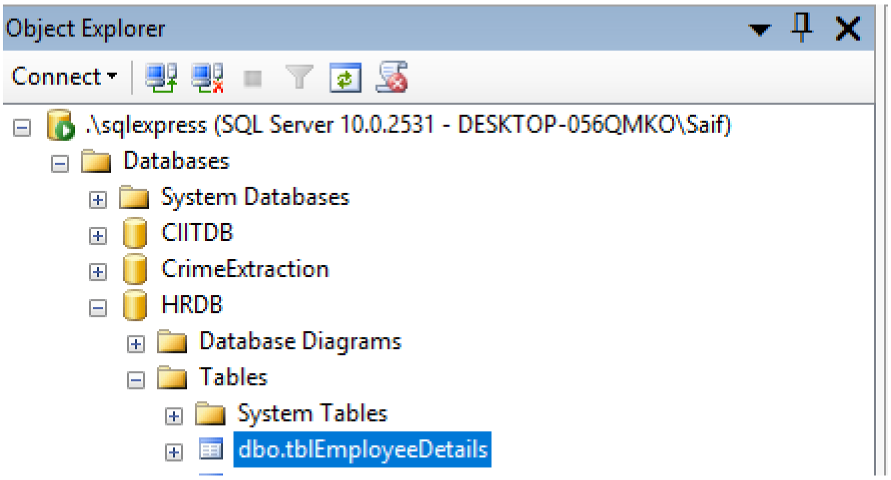


Column Properties

(General)	
(Name)	EmployeeSID
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1

IsIDentity will be set to YES only the data type of Primary Key attribute is int.

i. Save the table and name it as **tblEmployeeDetails**. Finally, you will see that the new table with the name tblEmployeeDetails is added to your database HRDB as shown

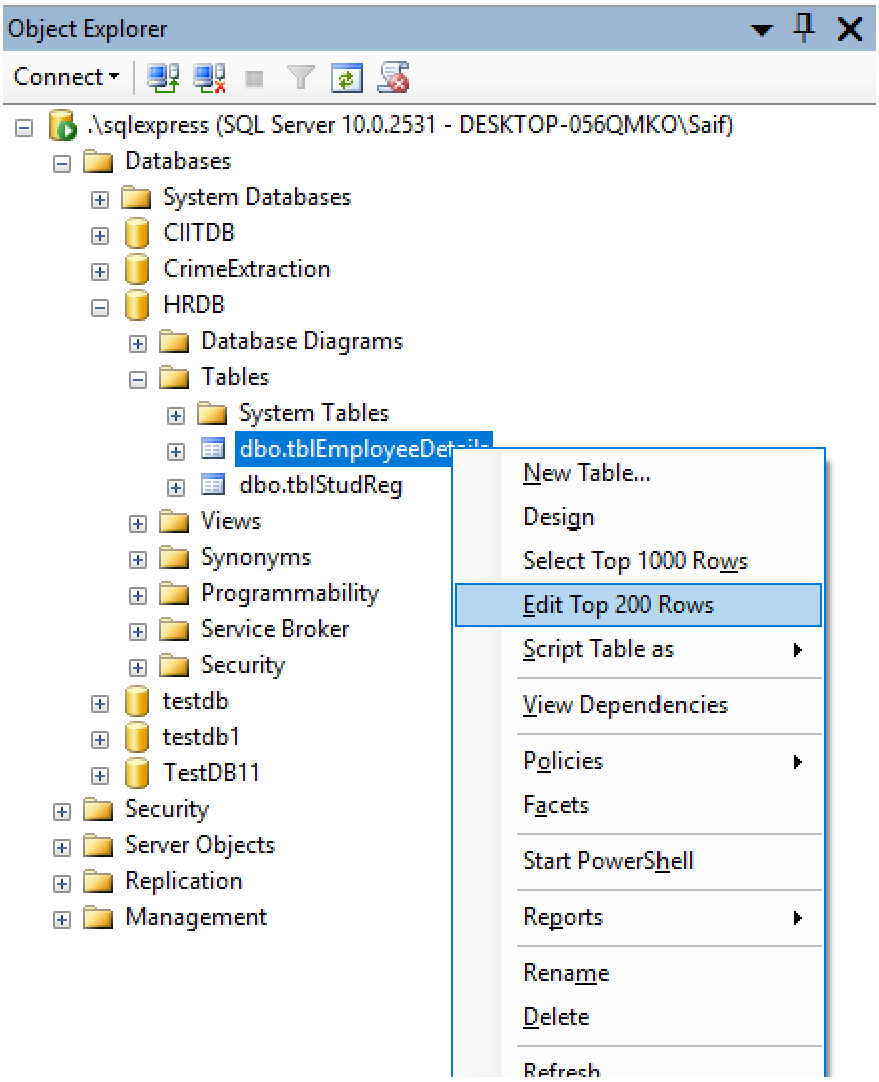


Object Explorer

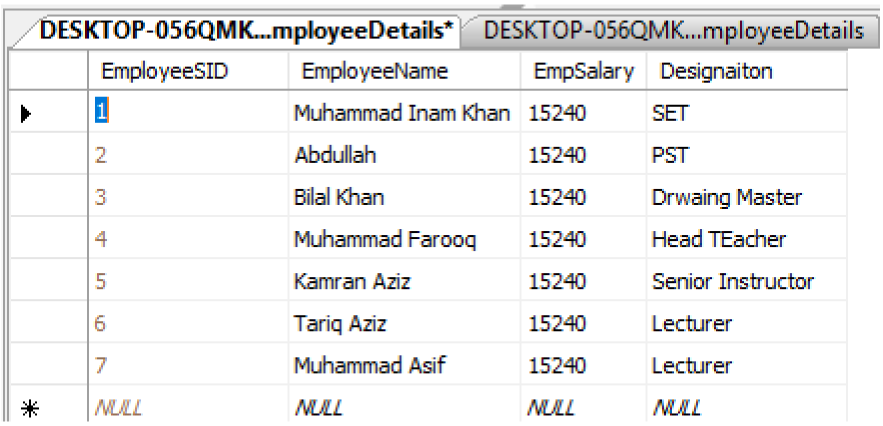
Connect

```
.\sqlexpress (SQL Server 10.0.2531 - DESKTOP-056QMKO\Saif)
  Databases
    System Databases
    CIITDB
    CrimeExtraction
    HRDB
      Database Diagrams
      Tables
        System Tables
        dbo.tblEmployeeDetails
```

j. To add some test data to table. Right Click on the table name and choose **Edit Top 200 Rows** (this option may be different for different SQL Server) as shown



k. Add some sample data as shown



DESKTOP-056QMK...mployeeDetails*
DESKTOP-056QMK...mployeeDetails

	EmployeeSID	EmployeeName	EmpSalary	Designaiton
▶	1	Muhammad Inam Khan	15240	SET
	2	Abdullah	15240	PST
	3	Bilal Khan	15240	Drwaing Master
	4	Muhammad Farooq	15240	Head TEacher
	5	Kamran Aziz	15240	Senior Instructor
	6	Tariq Aziz	15240	Lecturer
	7	Muhammad Asif	15240	Lecturer
*	NULL	NULL	NULL	NULL

That's all, here we have finally completed the Step#1 for Connecting C# to SQL Server2008 Database.

Hands out are prepared by

Saif Ur Rehman

Assistant Professor

University Institute of Information Technology

PMAS Arid Agriculture University

Murree Road, Rawalpindi

More Hands out on ASP.Net with C#, PHP with MySQL, Java, Programming Fundamentals and much more can be found at

www.aspuiit.blogspot.com

Suggestions are welcome at

Saif at uaar dot edu dot pk

Chapter No.4

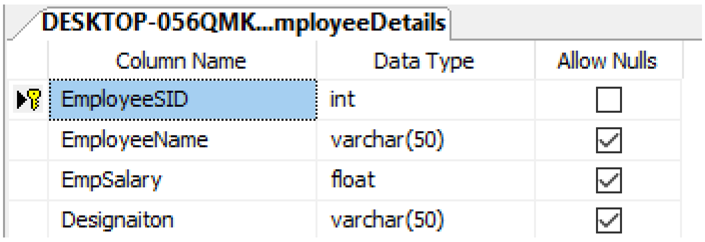
C# to MS SQL Server Database

C# to MS SQL Server Windows Form Application Development Steps

Task :

Fill a DataGridView from a MS SQL Server database Table

For this task:

Database Name:	HRDB
Table Name:	tblEmployeeDetails
Table Structure	

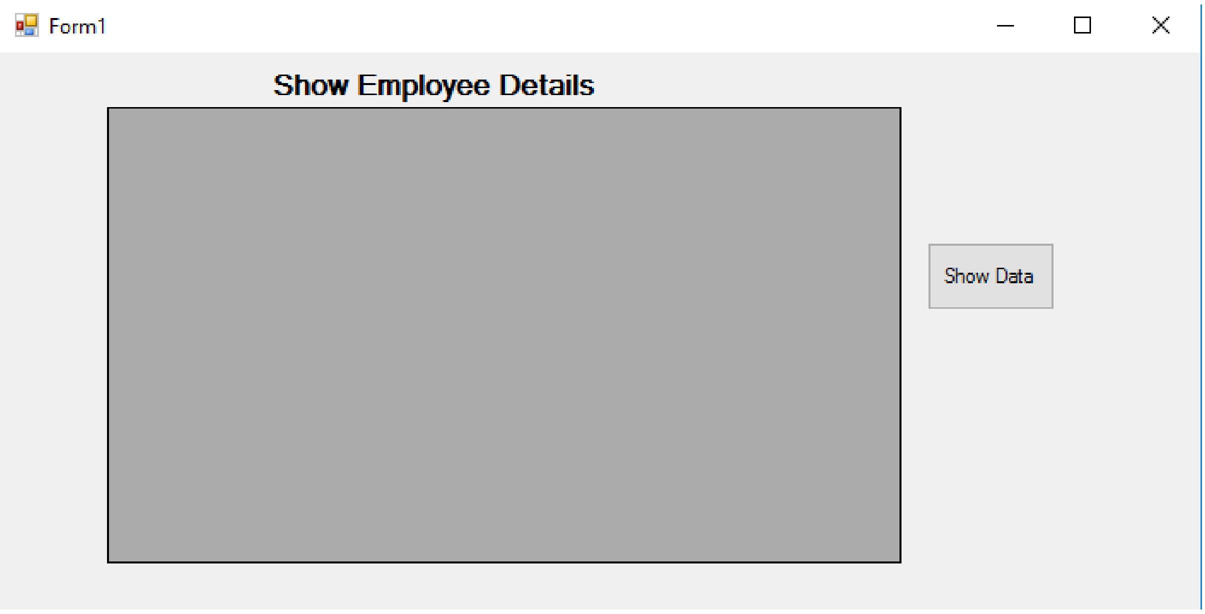
DESKTOP-056QMK...mployeeDetails

Column Name	Data Type	Allow Nulls
EmployeeSID	int	□
EmployeeName	varchar(50)	☑
EmpSalary	float	☑
Designaiton	varchar(50)	☑

1. Design a Windows Form

Start a New Windows Form Application. Rename it as **CSharpToDB**.

Design the Form as shown. Drag and Drop a DataGridView, Label and a Button control from ToolBox to Form



Next, Coding will be written behind the click event of the Button. This is described in the next step

2. Coding to Fetch Data from MicroSoft SQL Server2008 database to C# windows form

In order to display the SQL Server database data to C# dataGridView Control, following two Class Libraries must be included:

```
using System.Data;
using System.Data.SqlClient;
```

There are some classes in these two class libraries which are necessary to instantiated before C# form is connected to SQL Server database. Some details about these classes are given here

SQLConnection Class:

C# application can connect to data in a SQL Server database using the .NET Framework Data Provider for SQL Server. The first step in a C# application is to create an instance of the Server object and to establish its connection to an instance of Microsoft SQL Server.

The **SqlConnection Object** is Handling the part of physical communication between the C# application and the SQL Server Database . An instance of the **SqlConnection** class in C# is supported the Data Provider for SQL Server Database. The **SqlConnection** instance takes Connection String as argument and pass the value to the Constructor statement.

Here is a sample code for establishing C# connection to SQL Server database using SQLConnection Object

```
String conStr = @"Data Source=.\sqlexpress ;
                  Initial Catalog=HRDB;
                  Integrated Security=true;";

SqlConnection myDbCon = new SqlConnection(conStr);

myDbCon.Open();
```

When the connection is established , SQL Commands will execute with the help of the Connection Object and retrieve or manipulate the data in the database.

Where,

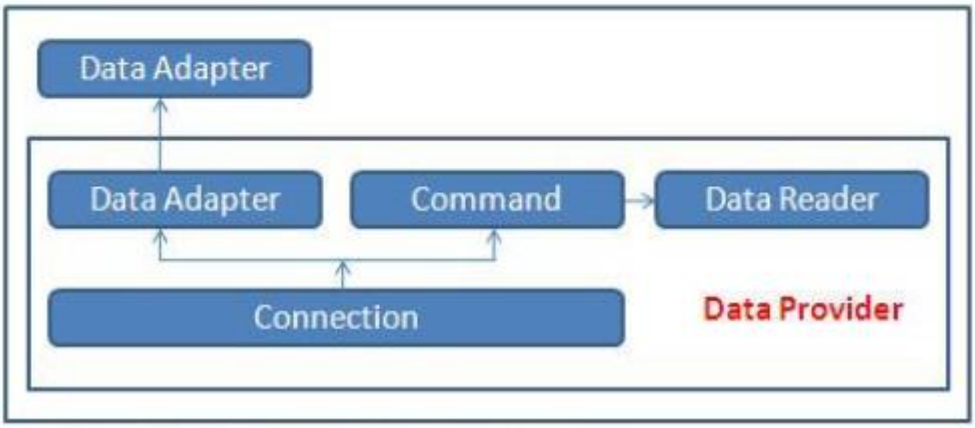
Data Source : is the name of the Machine/PC which is running the Instance of the SQL Server

Initial Catalog : is the name of database to which you want to connect.

Integrated Security : Defines the authentication for log into SQL Server

SqlDataAdapter Class:

DataAdapter is a part of the ADO.NET Data Provider. **DataAdapter** is used to retrieve data from a data source and populate tables within a **DataSet**. The **DataAdapter** also resolves changes made to the **DataSet** back to the data source. The Fill method uses the **DataReader** object implicitly to return the column names and types used to create the tables in the **DataSet**, as well as the data to populate the rows of the tables in the **DataSet**. Dataset represents a collection of data retrieved from the Data Source and saving data to the Data Source. We can use Dataset in combination with **DataAdapter** class.



C# Dataset

The ADO.NET DataSet contains DataTableCollection and their DataRelationCollection . It represents a complete set of data including the tables that contain, order, and constrain the data, as well as the relationships between the tables.

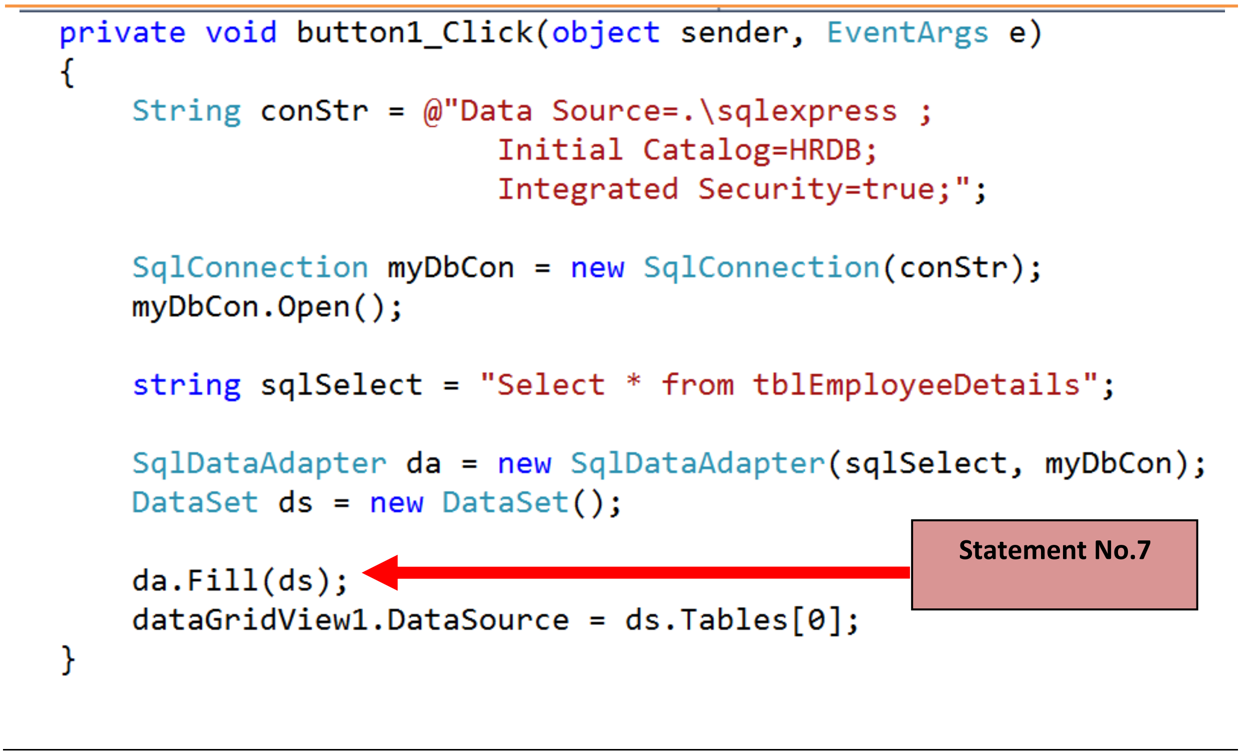
We can use Dataset in combination with DataAdapter Class . Build and fill each DataTable in a DataSet with data from a data source using a DataAdapter. The DataSet object offers a disconnected data source architecture.

The sample code for SqlDataAdapter to fill the Dataset is shown here

```
string sqlSelect = "Select * from tblEmployeeDetails";

SqlDataAdapter da = new SqlDataAdapter(sqlSelect, myDbCon);

DataSet ds = new DataSet();

da.Fill(ds);
```

Finally, the complete code in C# to fill a DataGridView from a SQL Server database

Table is given by



```
private void button1_Click(object sender, EventArgs e)
{
    String conStr = @"Data Source=.\sqlexpress ;
                        Initial Catalog=HRDB;
                        Integrated Security=true;";

    SqlConnection myDbCon = new SqlConnection(conStr);
    myDbCon.Open();

    string sqlSelect = "Select * from tblEmployeeDetails";

    SqlDataAdapter da = new SqlDataAdapter(sqlSelect, myDbCon);
    DataSet ds = new DataSet();

    da.Fill(ds);
    dataGridView1.DataSource = ds.Tables[0];
}
```

Statement No.7

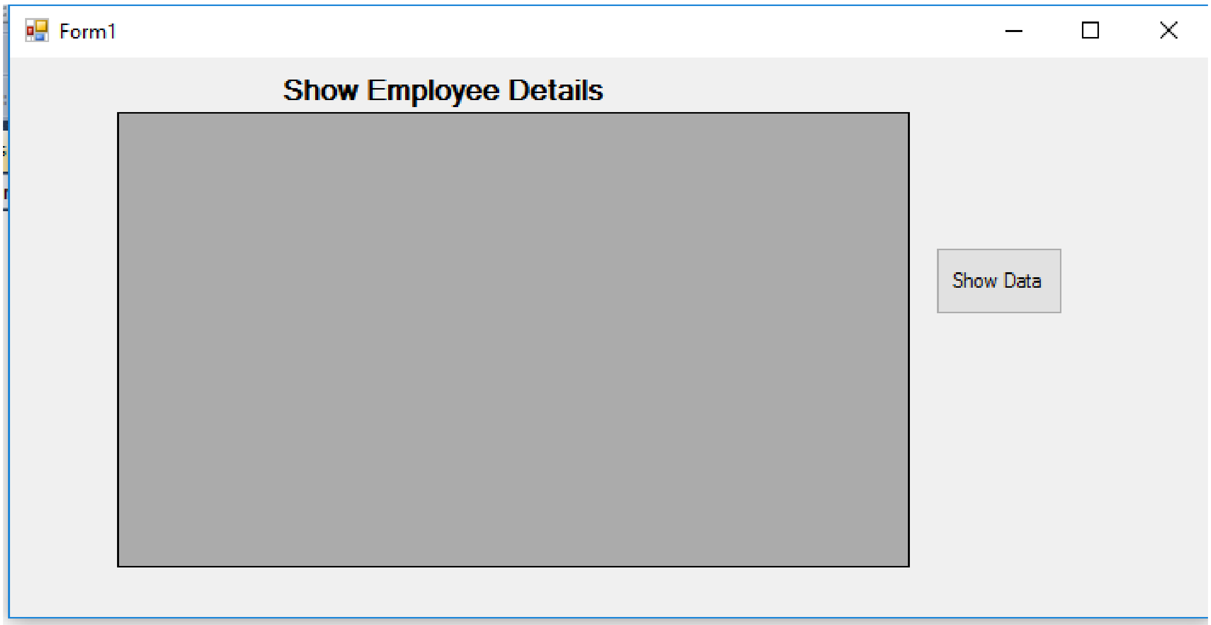
In Statement no.7, Fill method of SqlDataAdapter class is used. This method is used to fill the Dataset object with the Query against the SqlConnection specified in its constructor.

In Statement No.8, we have code

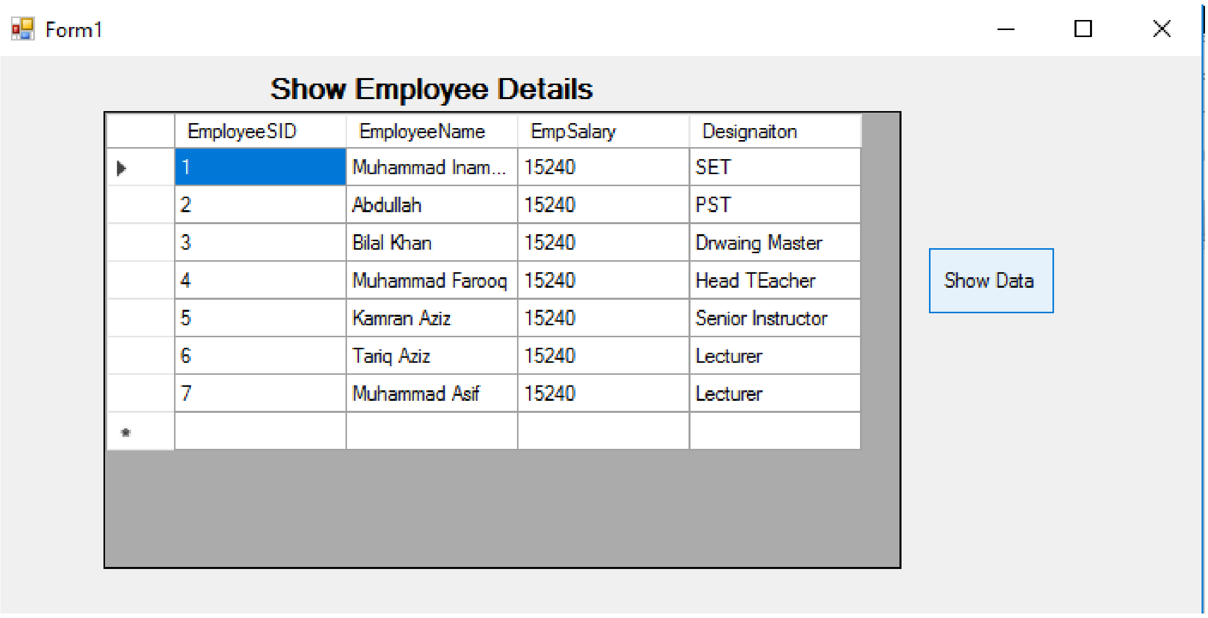
```
dataGridView1.DataSource = ds.Tables[0];
```

As the Tables in the Dataset are indexed. Indexing starts from 0. So, therefore, we have provided the index 0 of dataset Tables to dataGridView1 as its Data Source.

This code is written behind the button click event. Save the Project and run the application.



Now, press the Show Data button. You will see that the dataGridView is filled with the entire database table records as shown



	EmployeeSID	EmployeeName	EmpSalary	Designaiton
▶	1	Muhammad Inam...	15240	SET
	2	Abdullah	15240	PST
	3	Bilal Khan	15240	Drwaing Master
	4	Muhammad Farooq	15240	Head TEacher
	5	Kamran Aziz	15240	Senior Instructor
	6	Tariq Aziz	15240	Lecturer
	7	Muhammad Asif	15240	Lecturer
*				

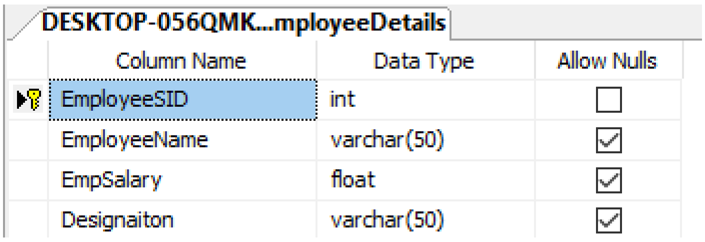
Congratulations! You have successfully connected the C# dataGridView to MS Sql Server database and executed the SELECT statement.

C# to Database

Task:

Add a New Record from C# Windows Form to MS SQL Server database Table

For this task:

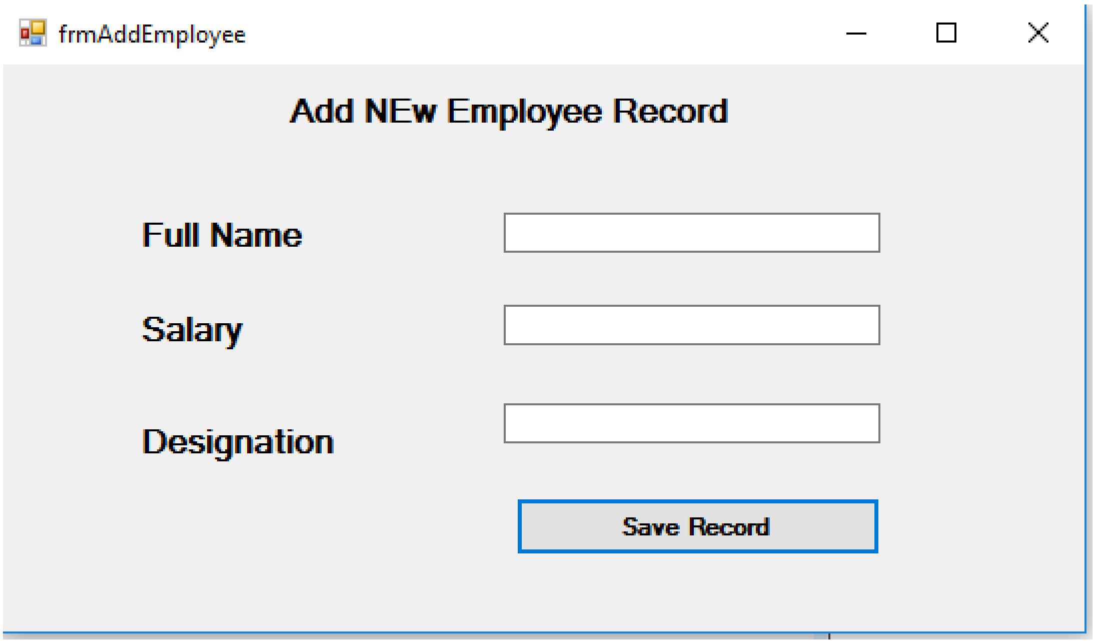
Database Name:	HRDB
Table Name:	tblEmployeeDetails
Table Structure	

DESKTOP-056QMK...mployeeDetails

Column Name	Data Type	Allow Nulls
EmployeeSID	int	□
EmployeeName	varchar(50)	☑
EmpSalary	float	☑
Designaiton	varchar(50)	☑

To add a New Record into MS SQL Server database Table, we have to follow these steps

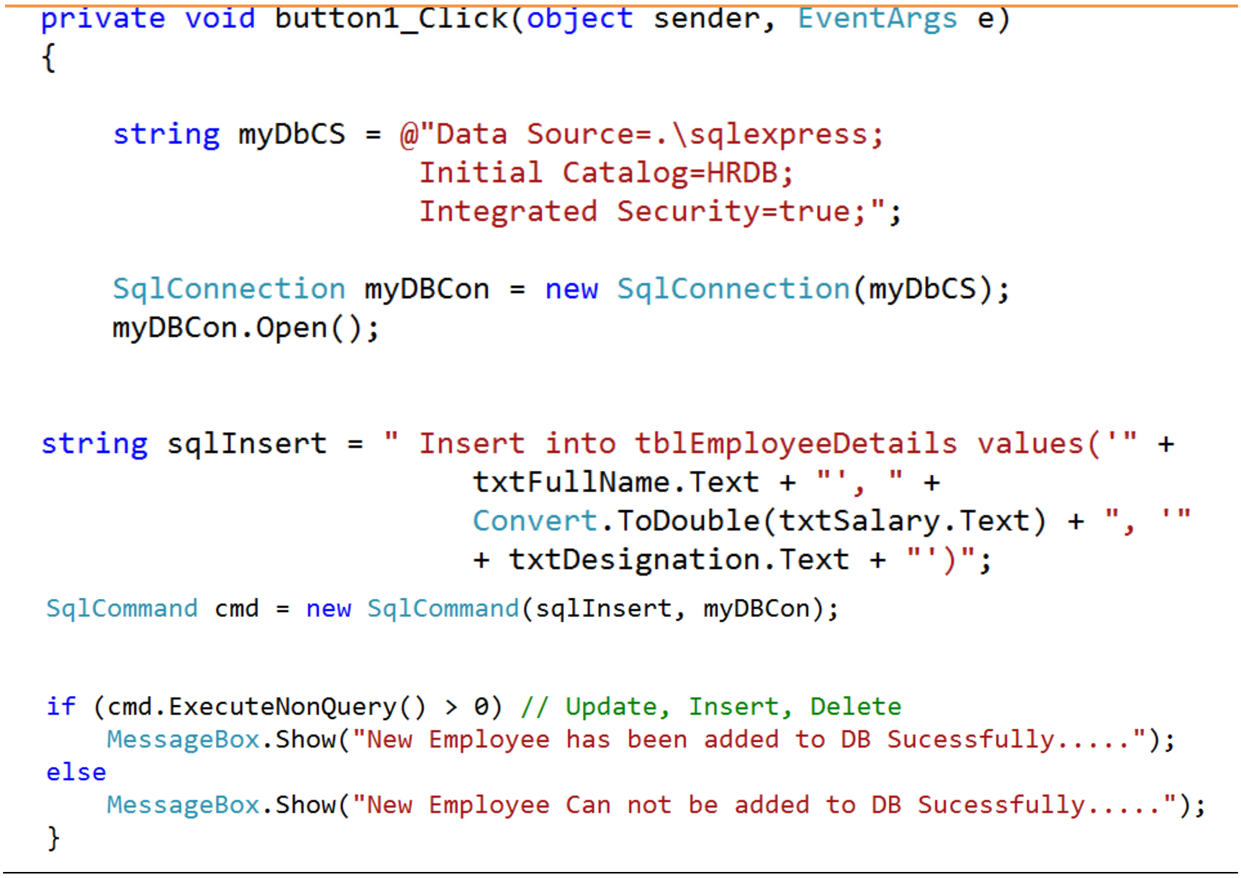
1. Design a C# Windows Form as shown here



2. Write Code to Add New Record

Double Click on the Save Record Button to create its Click Event. In the Click Event, add the following code for adding a new Record to database Table

sample code for SqlDataAdapter to fill the Dataset is shown here

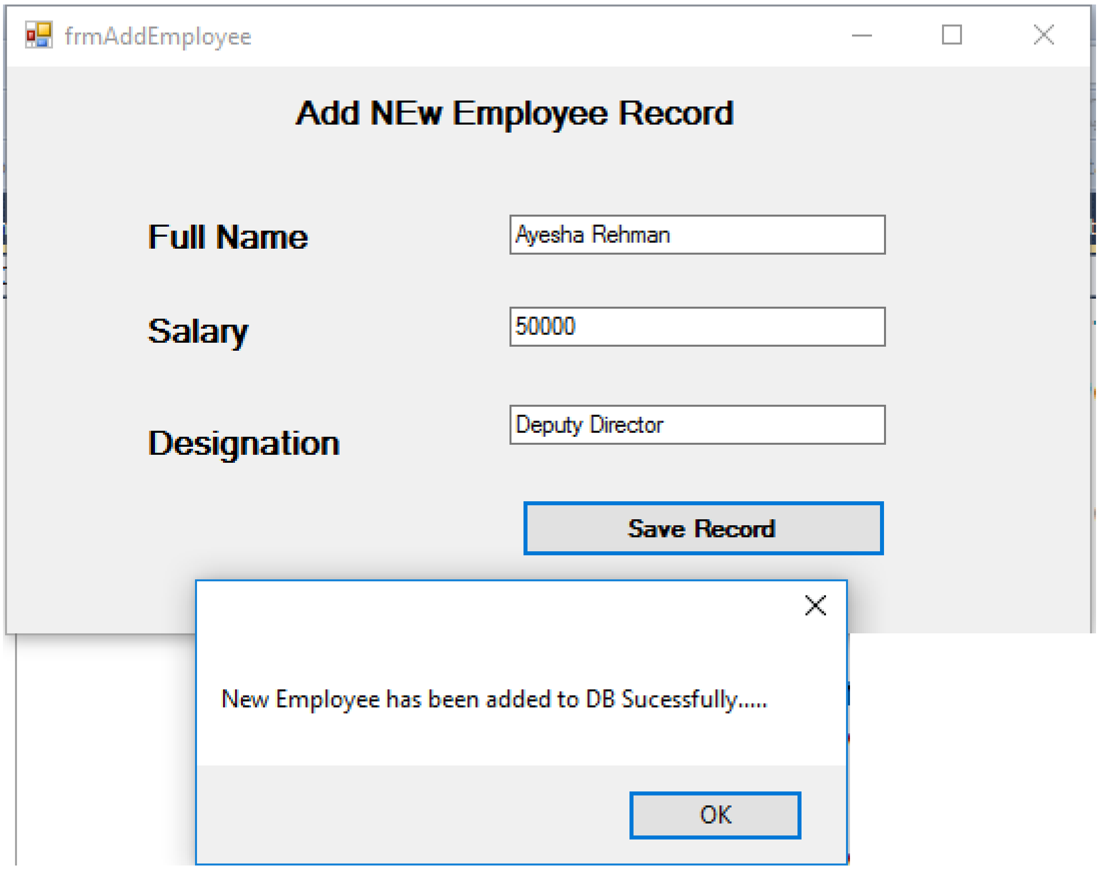


```
private void button1_Click(object sender, EventArgs e)
{

    string myDbCS = @"Data Source=.\sqlexpress;
                      Initial Catalog=HRDB;
                      Integrated Security=true;";

    SqlConnection myDBCon = new SqlConnection(myDbCS);
    myDBCon.Open();


string sqlInsert = " Insert into tblEmployeeDetails values('" +
                         txtFullName.Text + "', " +
                         Convert.ToDouble(txtSalary.Text) + ", '"
                         + txtDesignation.Text + "')";
SqlCommand cmd = new SqlCommand(sqlInsert, myDBCon);


if (cmd.ExecuteNonQuery() > 0) // Update, Insert, Delete
    MessageBox.Show("New Employee has been added to DB Sucessfully.....");
else
    MessageBox.Show("New Employee Can not be added to DB Sucessfully.....");
}
```

Here, we have used the SqlCommand object to execute the Query. The constructor of the SqlCommand takes two parameters:

1. Query to be executed
2. The database connection against the Query specified in Step# 1 (above) is executed

After specifying the two parameters in the Constructor of SqlCommand object, the **ExecuteNonQuery()** method is called. The **ExecuteNonQuery()** execute the DML queries (i.e. Insert Into, Update or Delete Statements). When this command is executed successfully then **ExecuteNonQuery()** returns a positive value representing the total number of records affected in the database with this Query. If Command is not executed successfully then it return non-Positive values. Therefore, we have used the **ExecuteNonQuery()** and checks if it returns value greater than 0 then a MessageBox is displayed showing the Message that Record has been added to DB. For example, add a new record as shown



Add NEw Employee Record

Full Name: Ayesha Rehman

Salary: 50000

Designation: Deputy Director

New Employee has been added to DB Sucessfully.....

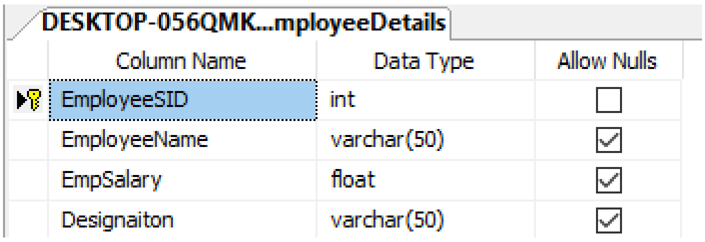
Here, a new Record has been Saved to DB successfully.

C# to Database

Task:

Delete a Record from C# Windows Form to MS SQL Server database Table

For this task:

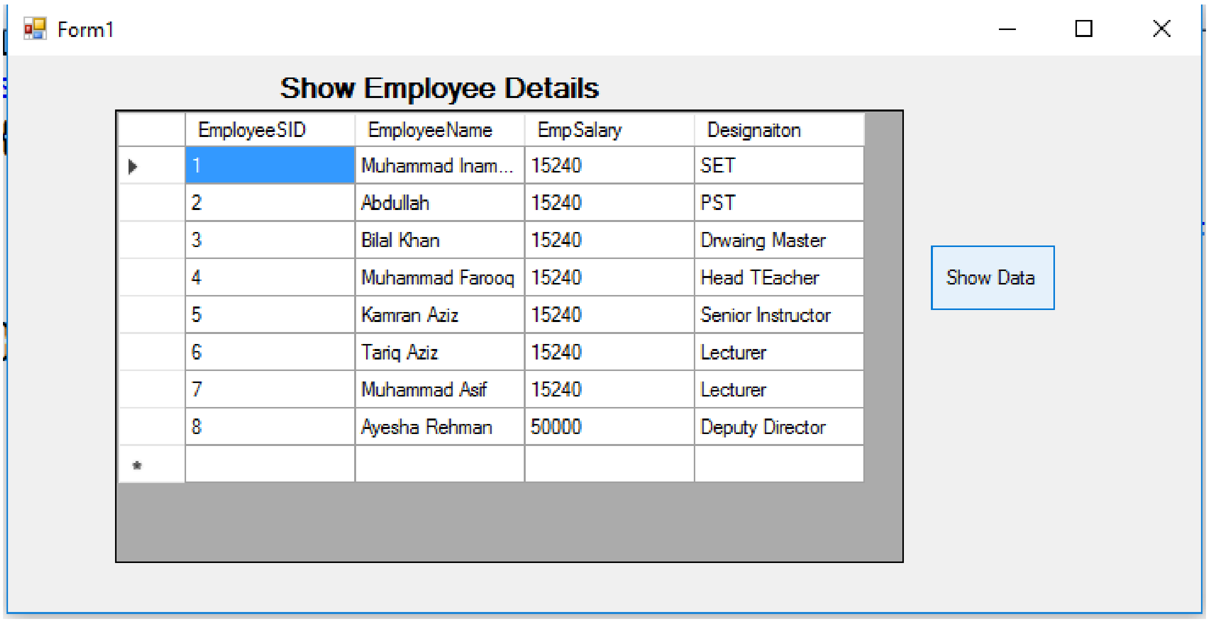
Database Name:	HRDB
Table Name:	tblEmployeeDetails
Table Structure	

DESKTOP-056QMK...mployeeDetails

Column Name	Data Type	Allow Nulls
EmployeeSID	int	□
EmployeeName	varchar(50)	☑
EmpSalary	float	☑
Designaiton	varchar(50)	☑

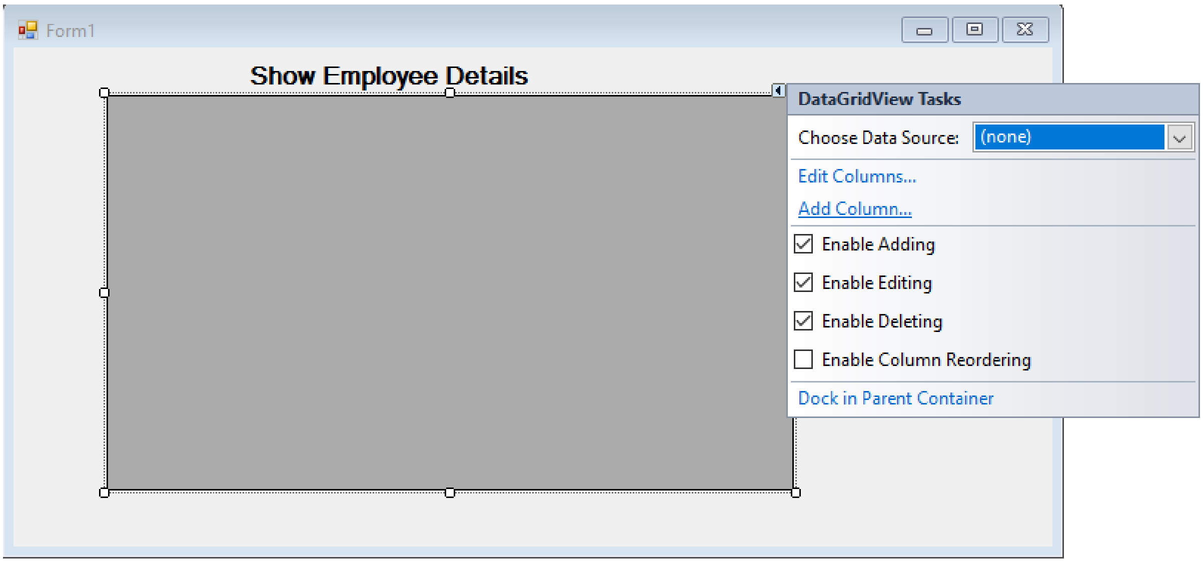
To Delete a Record From MS SQL Server database Table, we have to follow these steps

1. Design a C# Windows Form by drag and drop a datagridview on the Windows form and write the code to fill the dataGridView control. We have already done this in the previous tasks. Therefore, the designed Windows form show all the Employee Records as shown below

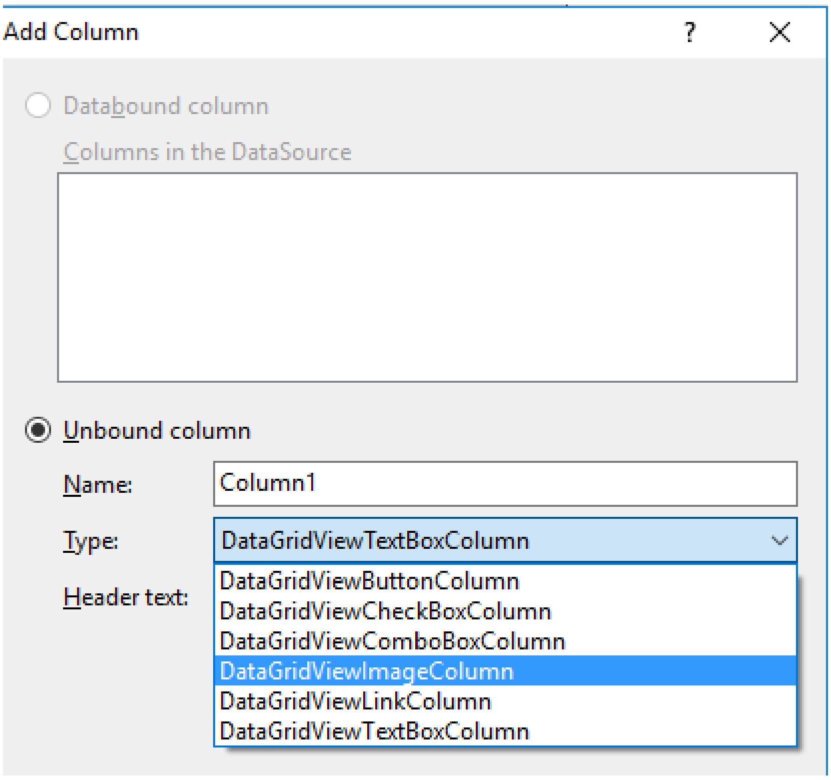


EmployeeSID	EmployeeName	EmpSalary	Designaiton
1	Muhammad Inam...	15240	SET
2	Abdullah	15240	PST
3	Bilal Khan	15240	Drwaing Master
4	Muhammad Farooq	15240	Head TEacher
5	Kamran Aziz	15240	Senior Instructor
6	Tariq Aziz	15240	Lecturer
7	Muhammad Asif	15240	Lecturer
8	Ayesha Rehman	50000	Deputy Director

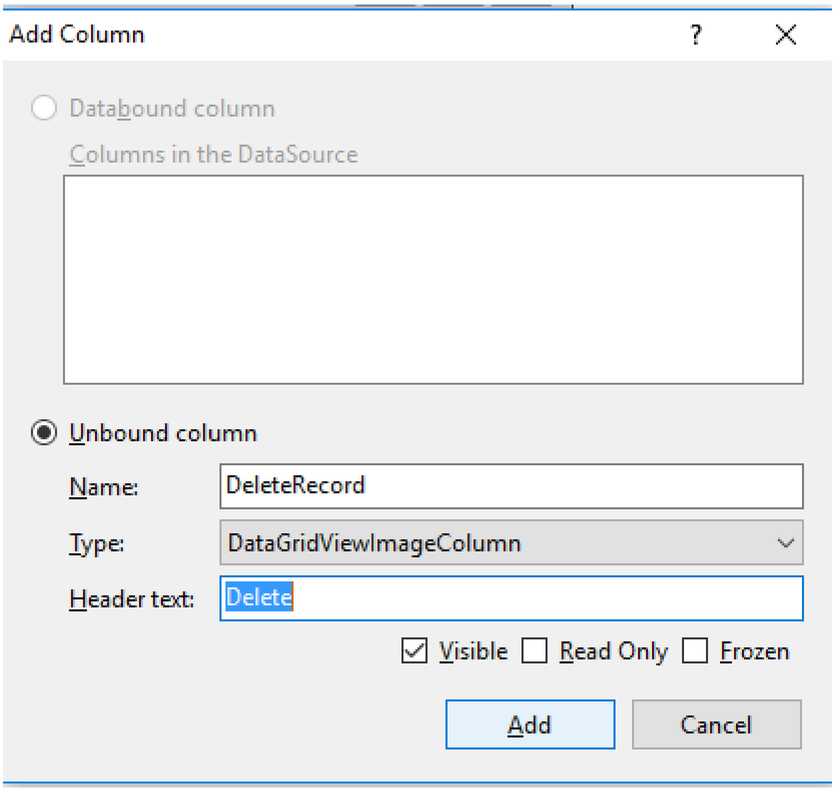
2. Now, open the Form in design View. Single Click on the dataGridView, you will see a small arrow on the Right Top Corner of the dataGridView, Click on this small arrow head as shown below



3. Choose the Add Column Option, you will get the screen as shown. Select the DataGridViewImageButton,



4. In the Add Column Window, Set the Header Text to **Delete,** and Column Name as **Delete Record** as shown



Add Column

○ Databound column

Columns in the DataSource

● Unbound column

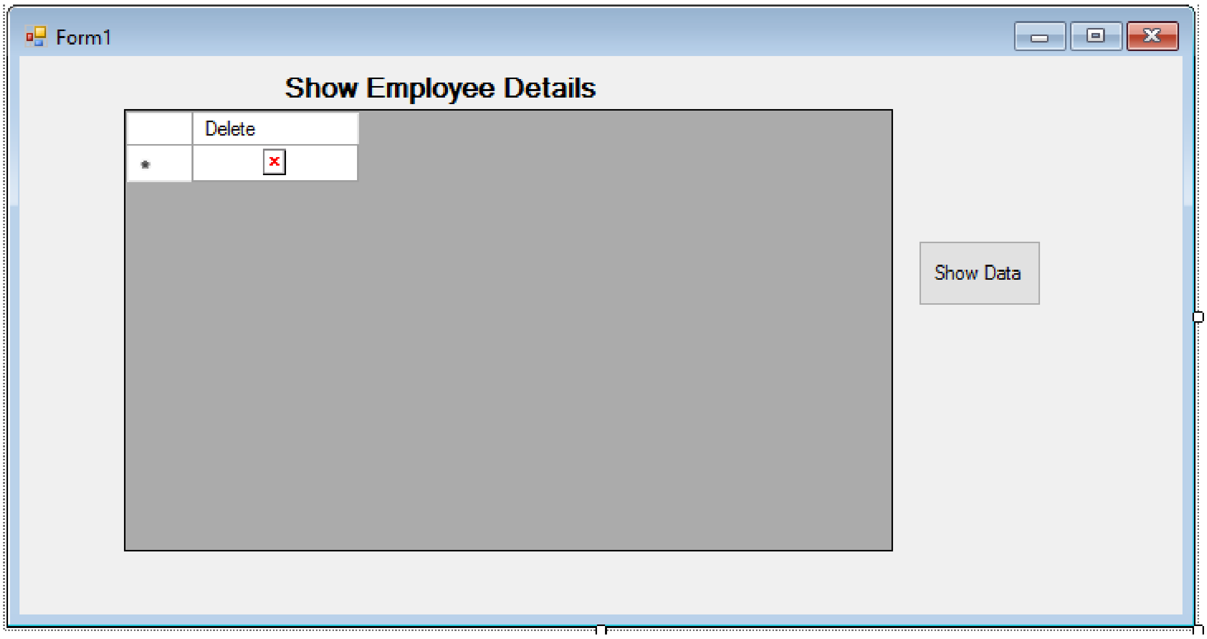
Name: DeleteRecord

Type: DataGridViewImageColumn

Header text: Delete

☑ Visible □ Read Only □ Frozen

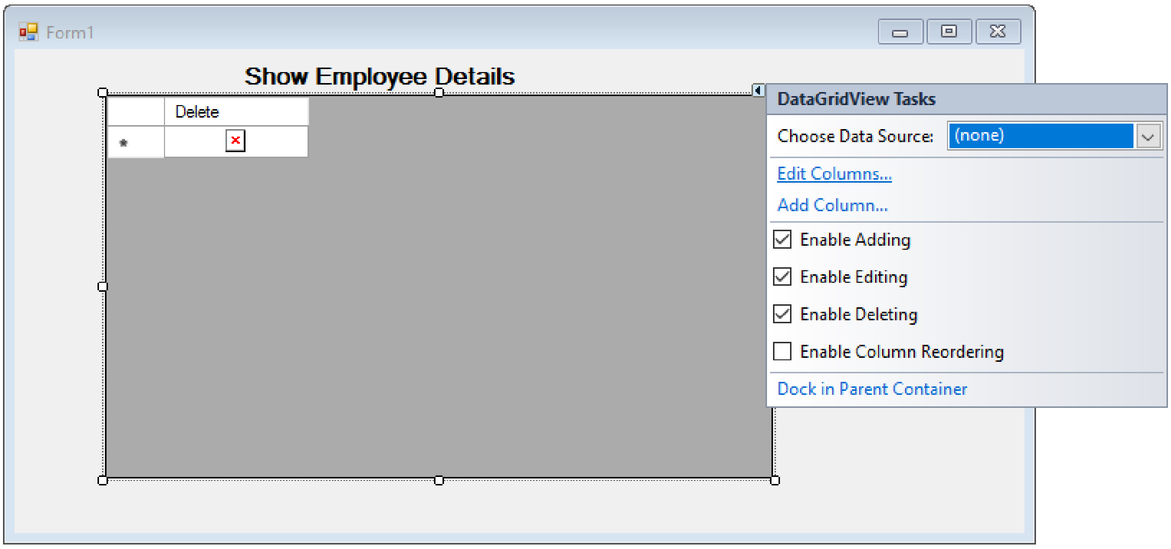
5. Click Add button you will see that a New Column is added to dataGridView as shown



Form1

Show Employee Details

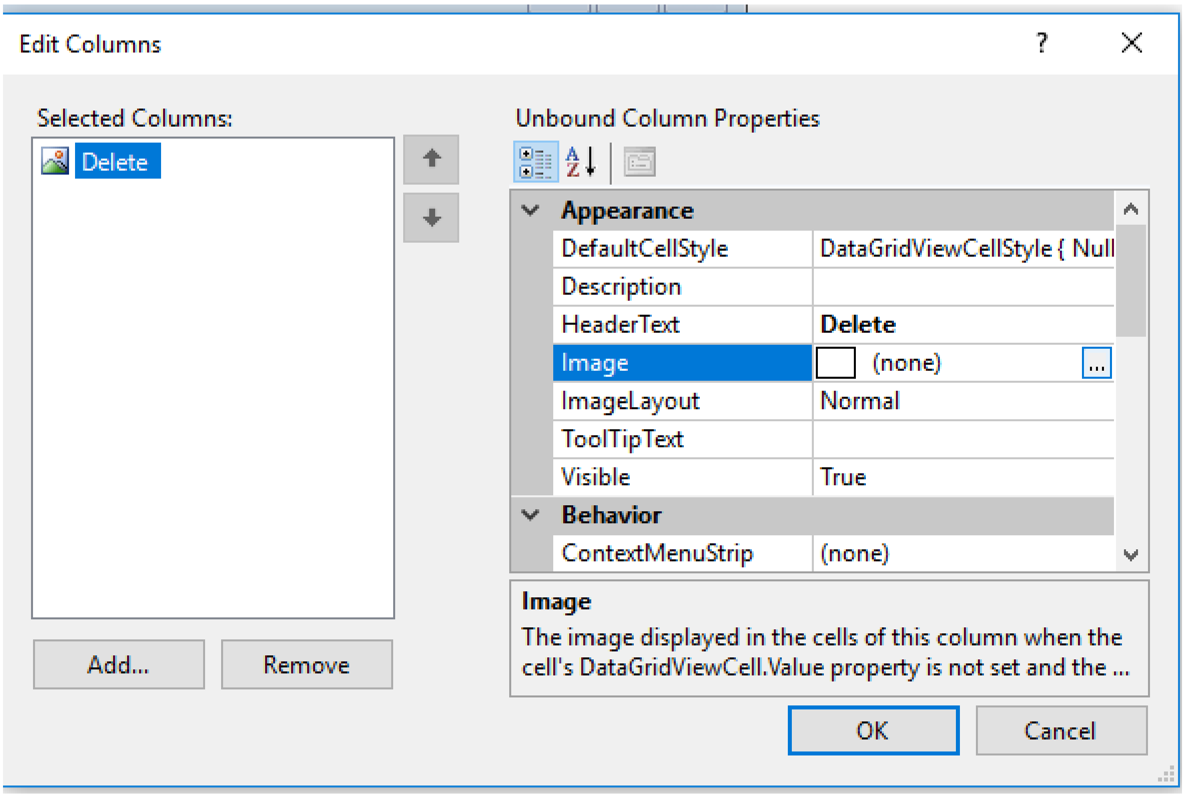
6. Choosing a Delete Image for the newly added column to dataGridView. Add a new Folder and rename it as Images to your Project. Add an image for delete record to this folder.Again Single Click on the dataGridView to bring the small arrow option. This time choose the **<u>Edit Column</u>** option.



Choose Data Source: (none)

☑ Enable Adding
☑ Enable Editing
☑ Enable Deleting
☐ Enable Column Reordering

7. You will see



Edit Columns

Selected Columns:

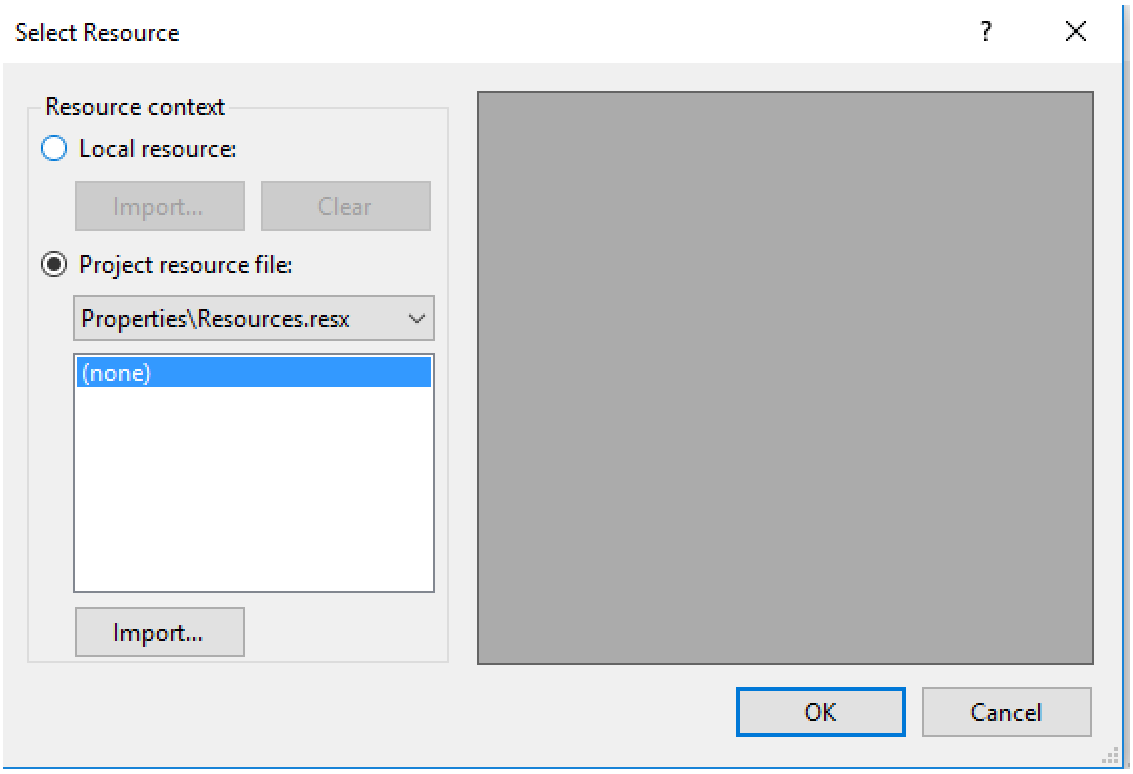
Delete

Unbound Column Properties

Appearance	
DefaultCellStyle	DataGridViewCellStyle { Null
Description	
HeaderText	**Delete**
Image	□ (none)
ImageLayout	Normal
ToolTipText	
Visible	True
Behavior	
ContextMenuStrip	(none)

Image
The image displayed in the cells of this column when the cell's DataGridViewCell.Value property is not set and the ...

8. Look for Image Property and click on the button in the Image Property to bring the window



Select Resource

Resource context

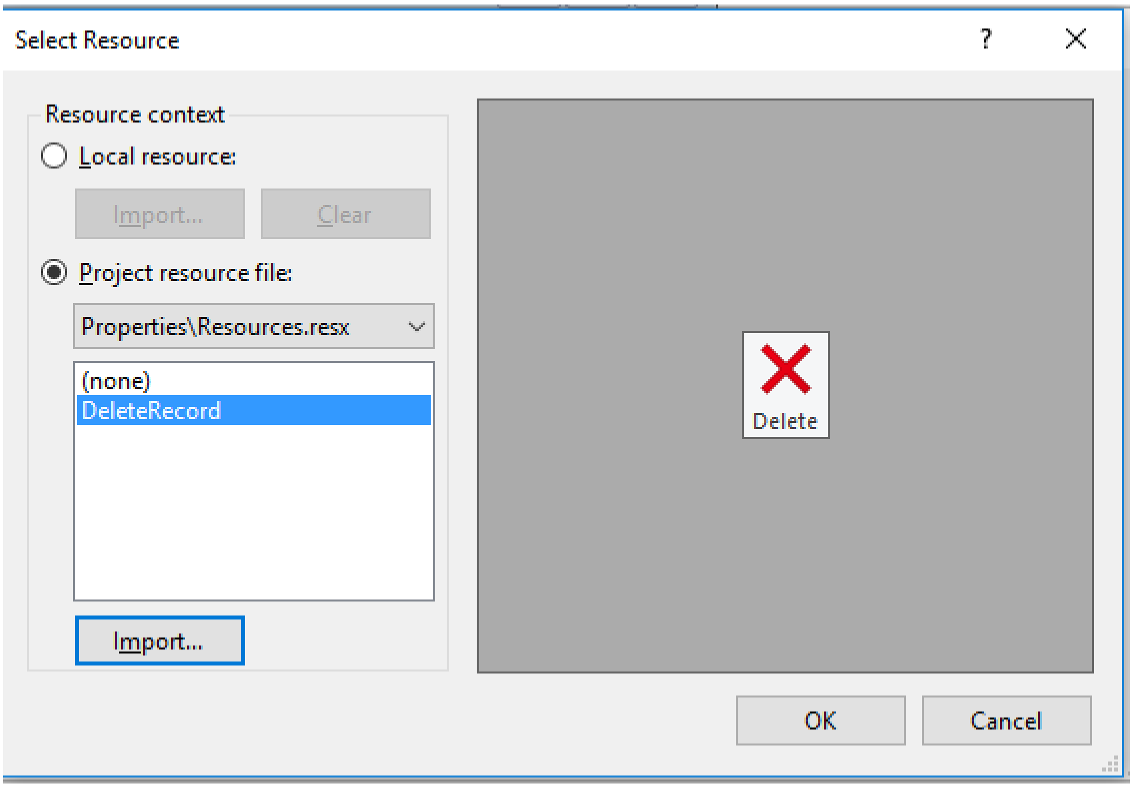
○ Local resource:

● Project resource file:

Properties\Resources.resx

(none)

9. Select the Project Resource and Press the Import button and finally browse the Delete image to set the image for the dataGridView Delete button



Select Resource

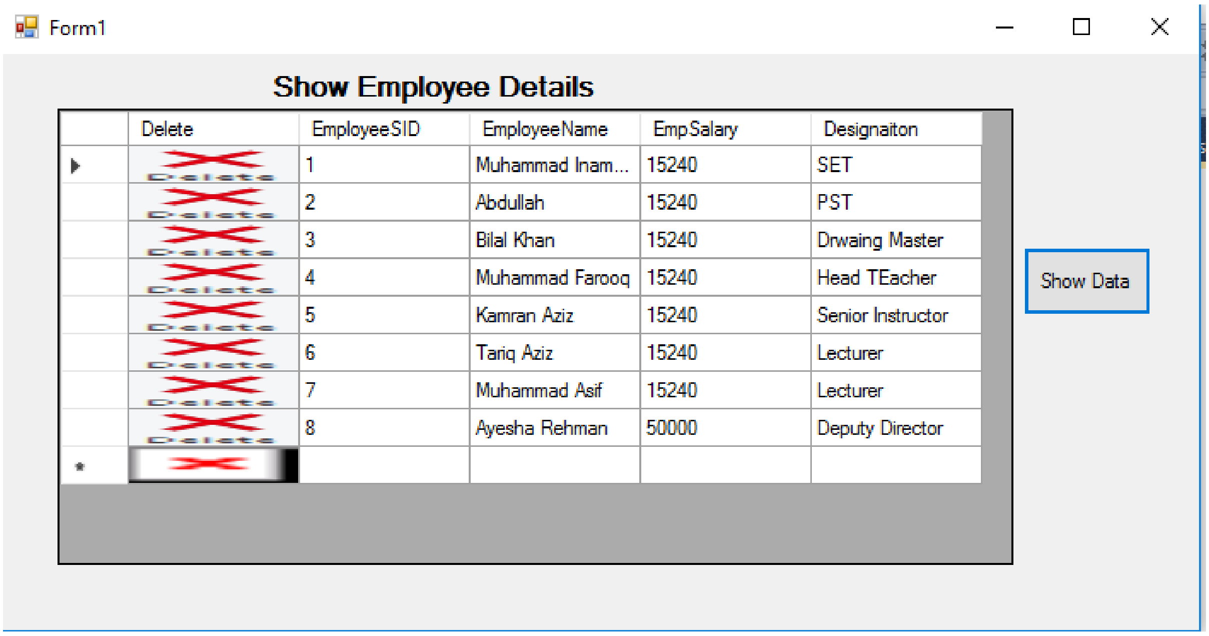
Resource context

○ Local resource:

● Project resource file: Properties\Resources.resx

(none)
DeleteRecord

10. Press the Ok button, this will modify the dataGridView as shown

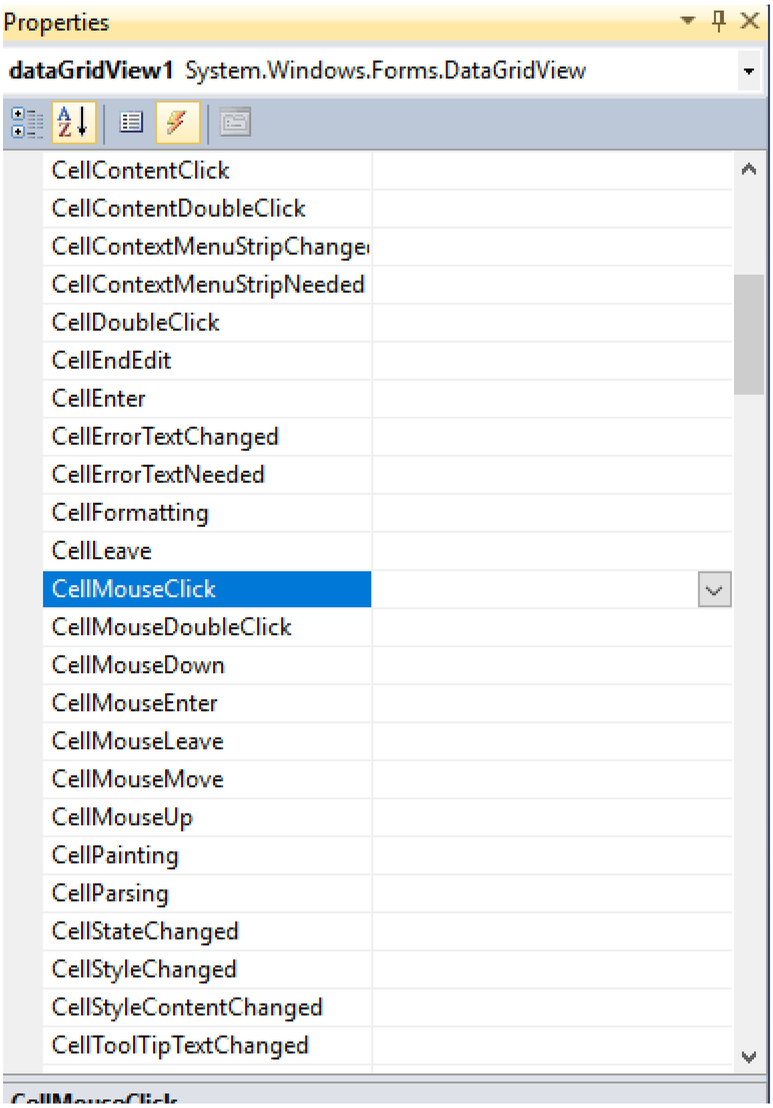


Form1

Show Employee Details

	Delete	EmployeeSID	EmployeeName	EmpSalary	Designaiton
▶	Delete	1	Muhammad Inam...	15240	SET
	Delete	2	Abdullah	15240	PST
	Delete	3	Bilal Khan	15240	Drwaing Master
	Delete	4	Muhammad Farooq	15240	Head TEacher
	Delete	5	Kamran Aziz	15240	Senior Instructor
	Delete	6	Tariq Aziz	15240	Lecturer
	Delete	7	Muhammad Asif	15240	Lecturer
	Delete	8	Ayesha Rehman	50000	Deputy Director
*					

11. Next, open the Events List for dataGridView as shown



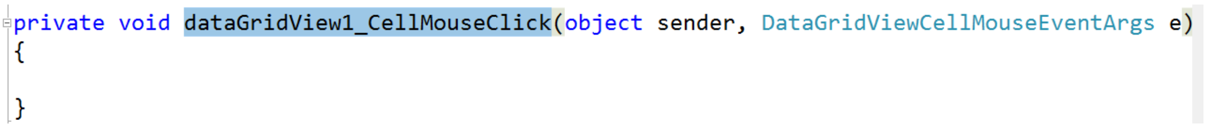
Properties

dataGridView1 System.Windows.Forms.DataGridView

CellContentClick
CellContentDoubleClick
CellContextMenuStripChange
CellContextMenuStripNeeded
CellDoubleClick
CellEndEdit
CellEnter
CellErrorTextChanged
CellErrorTextNeeded
CellFormatting
CellLeave
CellMouseClick
CellMouseDoubleClick
CellMouseDown
CellMouseEnter
CellMouseLeave
CellMouseMove
CellMouseUp
CellPainting
CellParsing
CellStateChanged
CellStyleChanged
CellStyleContentChanged
CellToolTipTextChanged

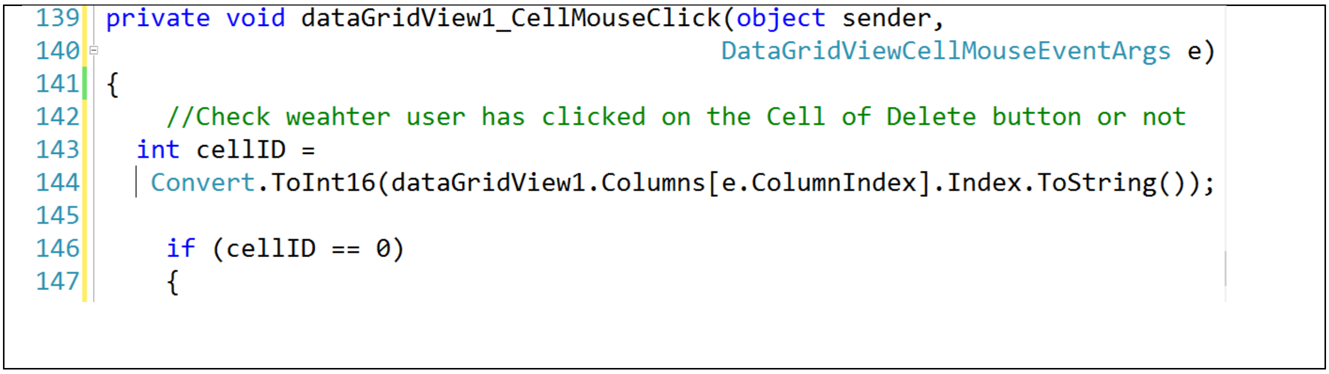
CellMouseClick

12. Double Click in CellMouseClick Event to create the Event

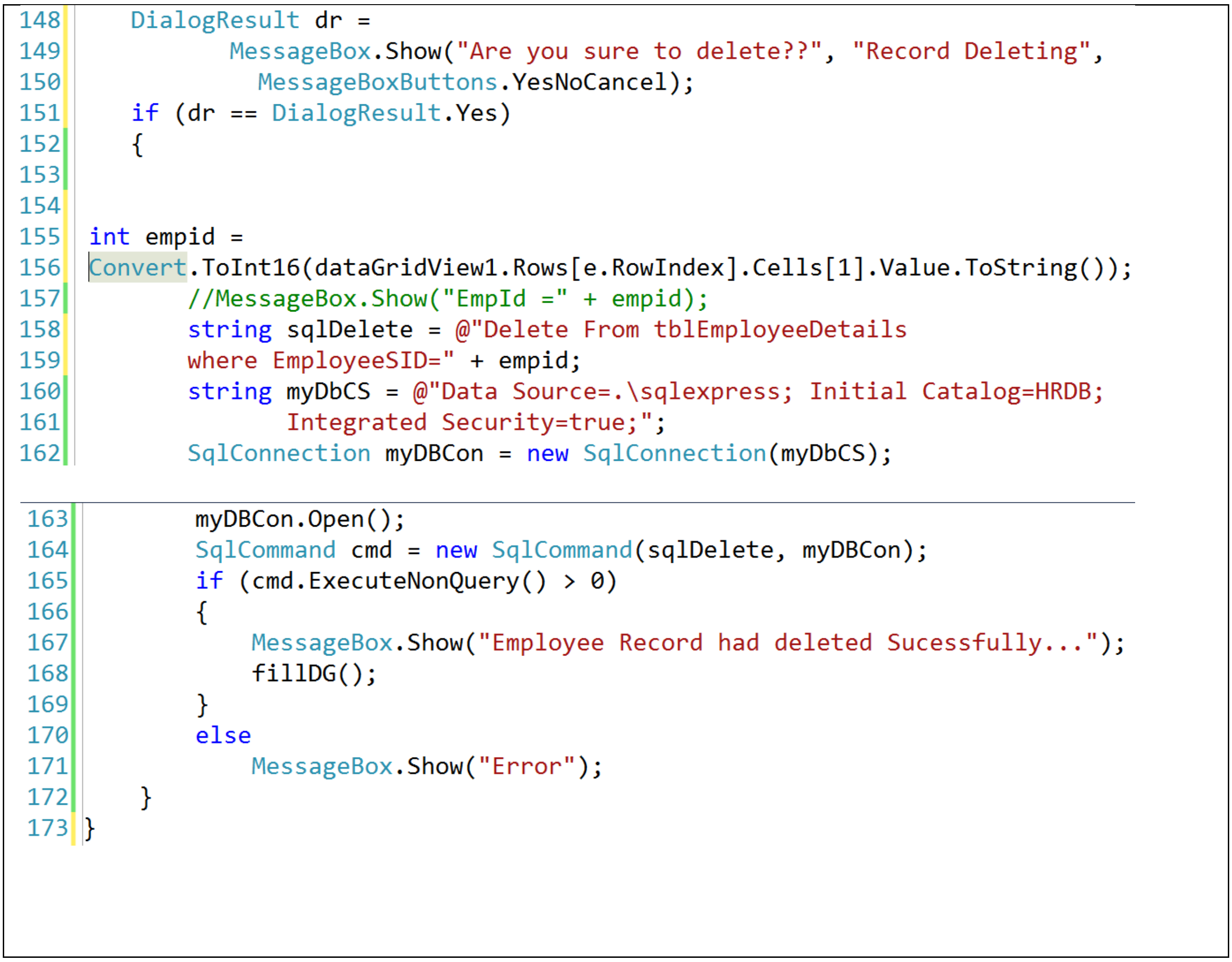


```
private void dataGridView1_CellMouseClick(object sender, DataGridViewCellMouseEventArgs e)
{

}
```

13. Type the following code in this event



```
private void dataGridView1_CellMouseClick(object sender,
                                          DataGridViewCellMouseEventArgs e)
{
    //Check weahter user has clicked on the Cell of Delete button or not
  int cellID =
  Convert.ToInt16(dataGridView1.Columns[e.ColumnIndex].Index.ToString());

    if (cellID == 0)
    {
```



```
148    DialogResult dr =
149           MessageBox.Show("Are you sure to delete??", "Record Deleting",
150             MessageBoxButtons.YesNoCancel);
151    if (dr == DialogResult.Yes)
152    {
153
154
155 int empid =
156 Convert.ToInt16(dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString());
157        //MessageBox.Show("EmpId =" + empid);
158        string sqlDelete = @"Delete From tblEmployeeDetails
159        where EmployeeSID=" + empid;
160        string myDbCS = @"Data Source=.\sqlexpress; Initial Catalog=HRDB;
161               Integrated Security=true;";
162        SqlConnection myDBCon = new SqlConnection(myDbCS);
```

```
163        myDBCon.Open();
164        SqlCommand cmd = new SqlCommand(sqlDelete, myDBCon);
165        if (cmd.ExecuteNonQuery() > 0)
166        {
167            MessageBox.Show("Employee Record had deleted Sucessfully...");
168            fillDG();
169        }
170        else
171            MessageBox.Show("Error");
172    }
173 }
```

Line number 143 and Line number 144 is shown here

```
int cellID =
Convert.ToInt16(dataGridView1.Columns[e.ColumnIndex].Index.ToString());
```

CellMouseClick() Event is called every time user clicks in any cell of the dataGridView control. In order to delete the record from dataGridView, it is necessary to identify that user has clicked in the Delete Cell, we have to check. So, here, first we will get the Cell Index clicked. This Cell Index is stored in the cellID variable. If user clicks on the Delete Cell then next line get it confirms from the user that he/she is sure to delete the record by using code given in Line number 148 to Line number 150.

```
DialogResult dr =MessageBox.Show("Are you sure to
                        delete??", "Record Deleting",
                          MessageBoxButtons.YesNoCancel);
```

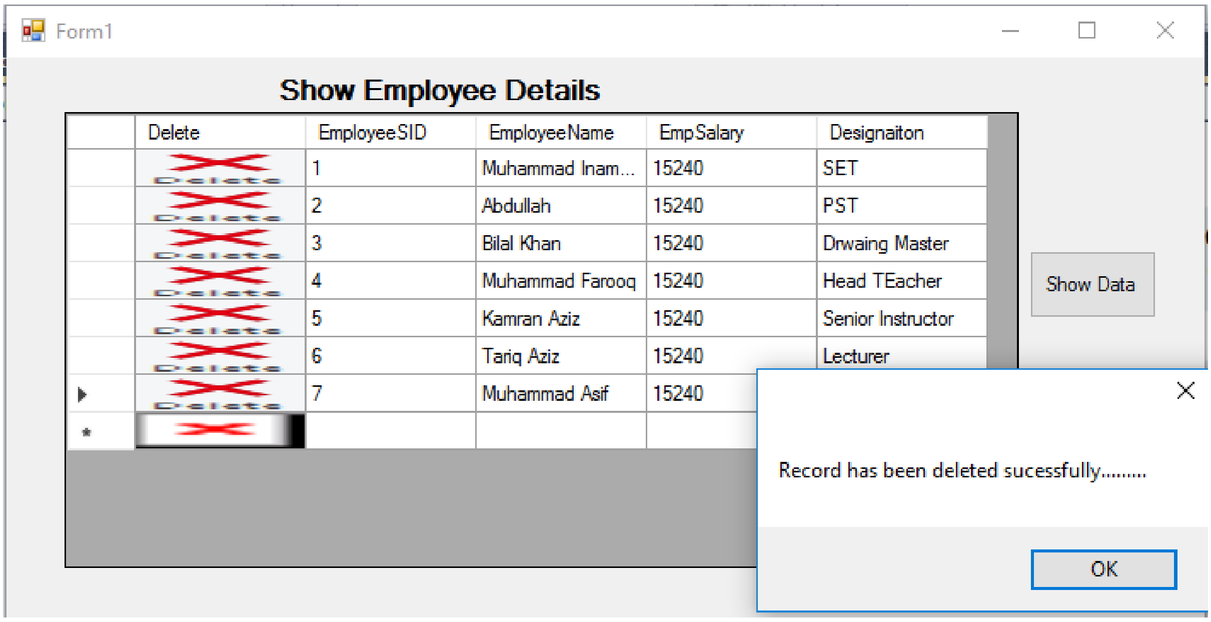
Which button on the MessageBox is clicked is saved to dr (an object od DialogResult Class). We can then check dr value with the code given in Line Number 151, as shown here

```
if (dr == DialogResult.Yes)
```

This will confirm that Yes button is clicked before the actually Delete the record from database.

The plain code is shown here

```
     private void dataGridView1_CellMouseClick(object
sender,

DataGridViewCellMouseEventArgs e)
        {
            //Check weahter user has clicked on the Cell
of Delete button or not
          int cellID =

Convert.ToInt16(dataGridView1.Columns[e.ColumnIndex].Index
.ToString());

            if (cellID == 0)
            {
                DialogResult dr =
                       MessageBox.Show("Are you sure to
delete??", "Record Deleting",
                         MessageBoxButtons.YesNoCancel);
                if (dr == DialogResult.Yes)
                {


             int empid =

Convert.ToInt16(dataGridView1.Rows[e.RowIndex].Cells[1].Va
lue.ToString());
                    //MessageBox.Show("EmpId =" + empid);
                    string sqlDelete = @"Delete From
tblEmployeeDetails
                    where EmployeeSID=" + empid;
                    string myDbCS = @"Data
Source=.\sqlexpress; Initial Catalog=HRDB;
                           Integrated Security=true;";
                    SqlConnection myDBCon = new
SqlConnection(myDbCS);
                    myDBCon.Open();
                    SqlCommand cmd = new
SqlCommand(sqlDelete, myDBCon);
                    if (cmd.ExecuteNonQuery() > 0)
                    {
                        MessageBox.Show("Employee Record
had deleted Sucessfully...");
                        fillDG();
                    }
                    else
                        MessageBox.Show("Error");
                }
            }

        }
```



Show Employee Details

	Delete	EmployeeSID	EmployeeName	EmpSalary	Designaiton
	Delete	1	Muhammad Inam...	15240	SET
	Delete	2	Abdullah	15240	PST
	Delete	3	Bilal Khan	15240	Drwaing Master
	Delete	4	Muhammad Farooq	15240	Head TEacher
	Delete	5	Kamran Aziz	15240	Senior Instructor
	Delete	6	Tariq Aziz	15240	Lecturer
▶	Delete	7	Muhammad Asif	15240	
*					

Record has been deleted sucessfully.........

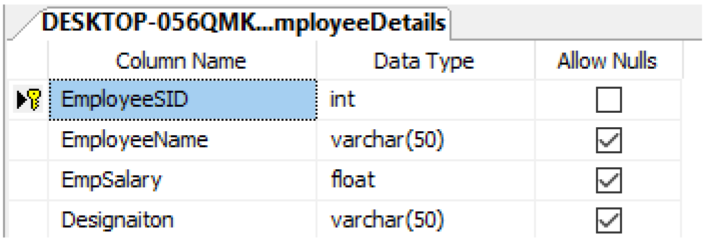
You can see that database record whose EmployeeSID is 7 and user has click on the Delete button has been deleted

C# to Database

Task:

Update a Record from C# Windows Form to MS SQL Server database Table

For this task:

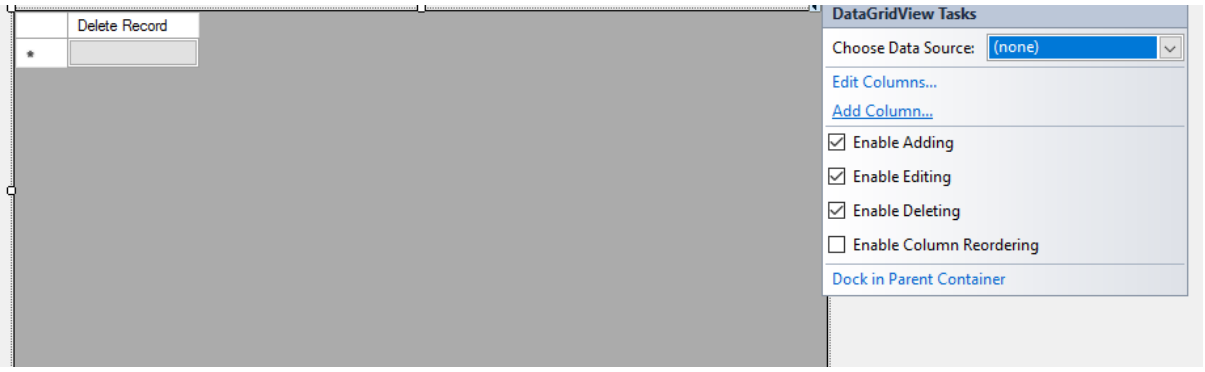
Database Name:	HRDB
Table Name:	tblEmployeeDetails
Table Structure	

DESKTOP-056QMK...mployeeDetails

Column Name	Data Type	Allow Nulls
EmployeeSID	int	□
EmployeeName	varchar(50)	☑
EmpSalary	float	☑
Designaiton	varchar(50)	☑

Repeating the Steps discussed in previous Task while we were Deleting a database record, design the Widows form as follow

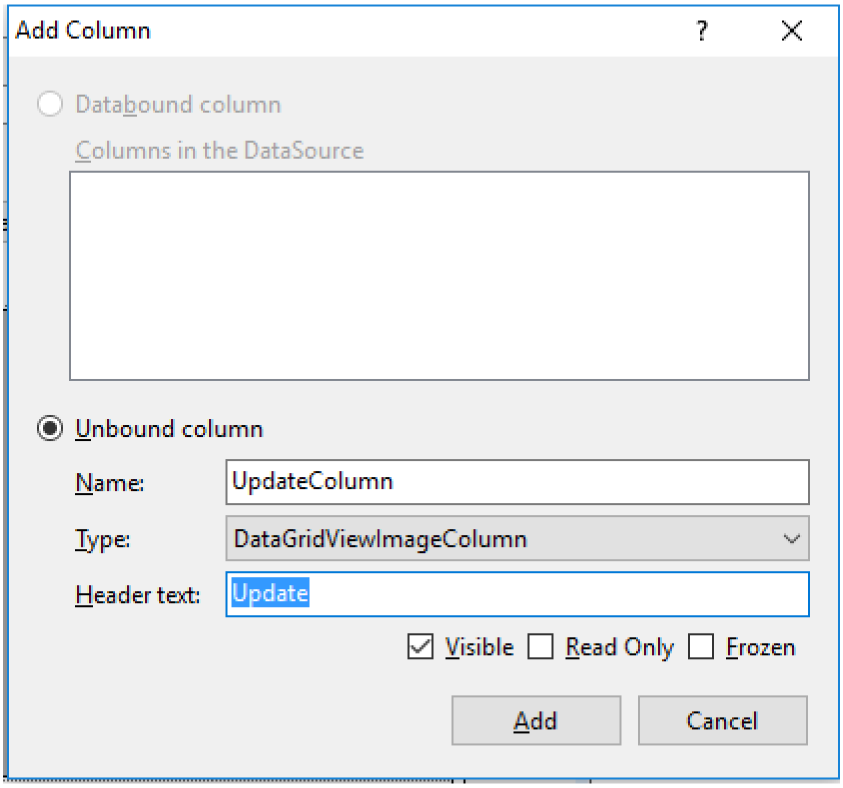
1. Right Click on the small arrow on the **dataGridView** control as shown



Choose Data Source: (none)

☑ Enable Adding
☑ Enable Editing
☑ Enable Deleting
□ Enable Column Reordering

2. Then enter the Column Name, choose the Column type and choose the column Header Text



Add Column

○ Databound column

Columns in the DataSource

● Unbound column

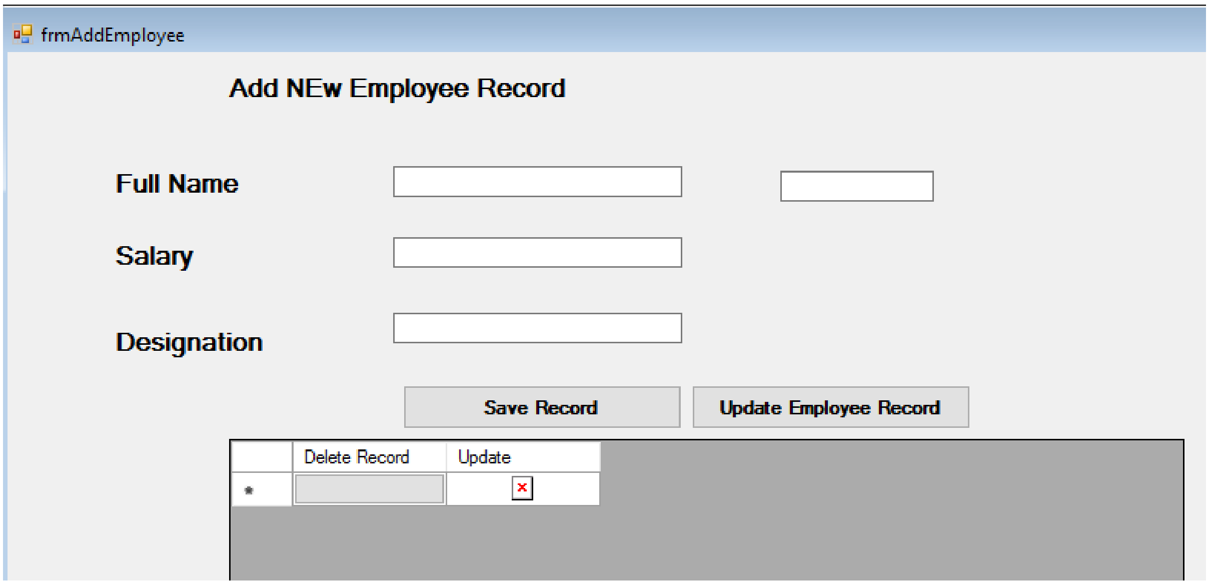
Name: UpdateColumn

Type: DataGridViewImageColumn

Header text: Update

☑ Visible □ Read Only □ Frozen

3. Press the Add Button. Again Single Click on the dataGridView to bring the small arrow option. This time choose the **Edit Column** option.
4. Add a TextBox for temporarily storing the **EmployeeSID**, as shown



frmAddEmployee

Add NEw Employee Record

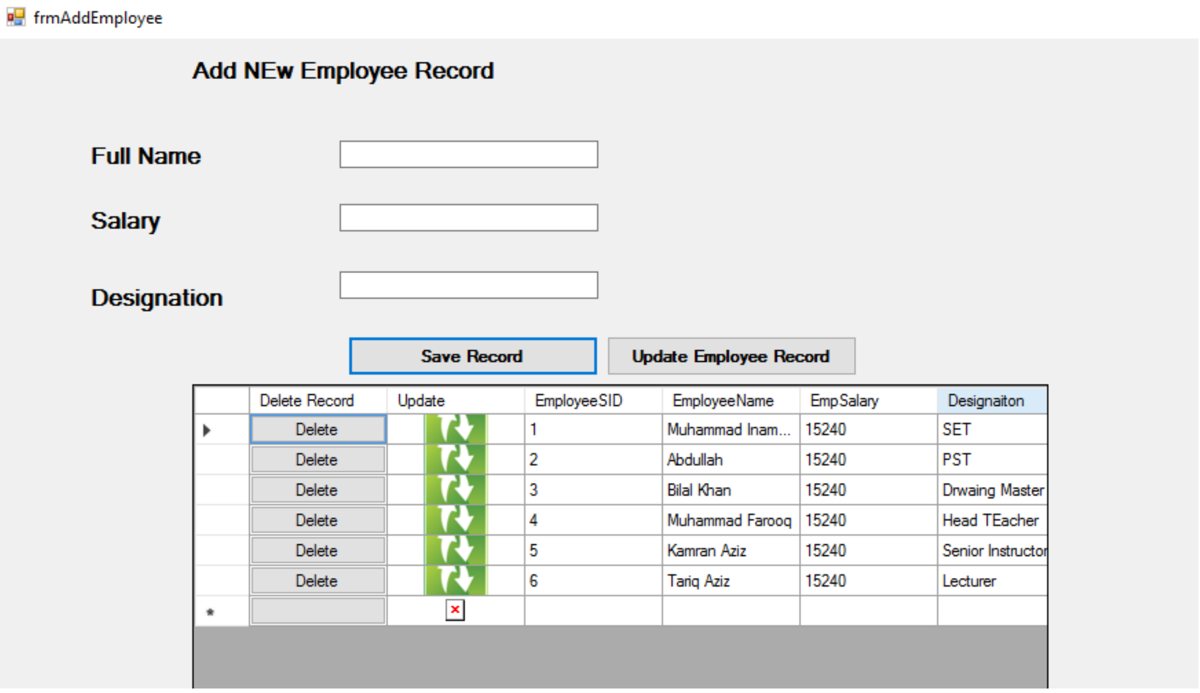
Full Name

Salary

Designation

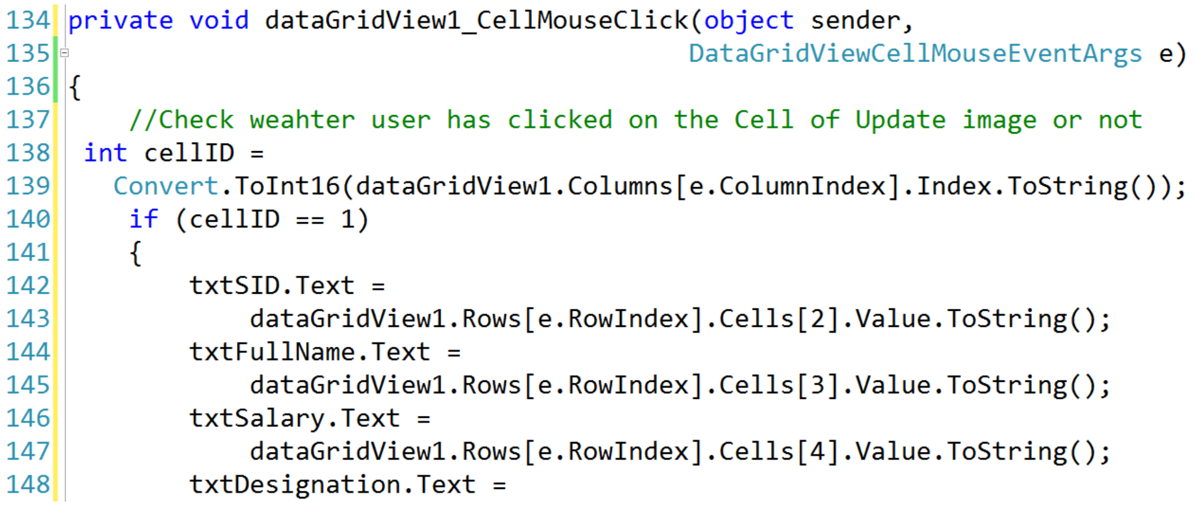
	Delete Record	Update
*		☒

5. Choose image for the Update **GridViewImageButton** as we did in the previous task
6. Run the form, you will see that dataGridView is shown with the Update image

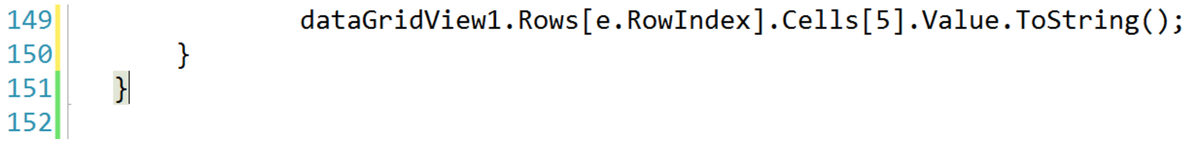


	Delete Record	Update	EmployeeSID	EmployeeName	EmpSalary	Designaiton
▶	Delete		1	Muhammad Inam...	15240	SET
	Delete		2	Abdullah	15240	PST
	Delete		3	Bilal Khan	15240	Drwaing Master
	Delete		4	Muhammad Farooq	15240	Head TEacher
	Delete		5	Kamran Aziz	15240	Senior Instructor
	Delete		6	Tariq Aziz	15240	Lecturer
*		☒				

7. Now, go to **<u>CellMouseClick</u>** event that we have created in the previous task. In this task, first we bring the data from dataGridView to text boxes. This is shown here



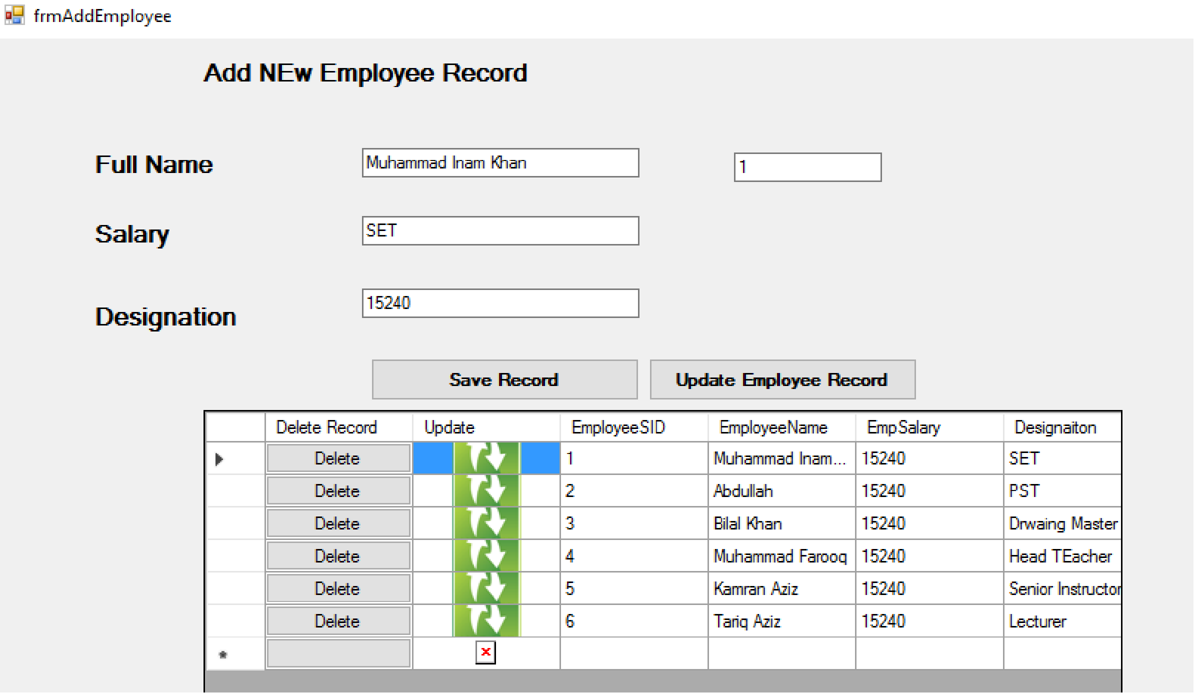
```
private void dataGridView1_CellMouseClick(object sender,
                                       DataGridViewCellMouseEventArgs e)
{
    //Check weahter user has clicked on the Cell of Update image or not
 int cellID =
   Convert.ToInt16(dataGridView1.Columns[e.ColumnIndex].Index.ToString());
    if (cellID == 1)
    {
        txtSID.Text =
            dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
        txtFullName.Text =
            dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
        txtSalary.Text =
            dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
        txtDesignation.Text =
```



```
            dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();
    }
}
```

The code given in the above figure is similar to one used for deleting the database record. Here. First we check whether the Edit Image Cell is clicked and then we retrieve the text from the dataGridView to corresponding text boxes and then make changes to the retrieved values and finally clicked the Update button to update the details. The Code for the Update button is shown in the next pages.

Now, run the application and you will see that text boxes are filled with the data of the row you clicked on. This is shown in the next figure



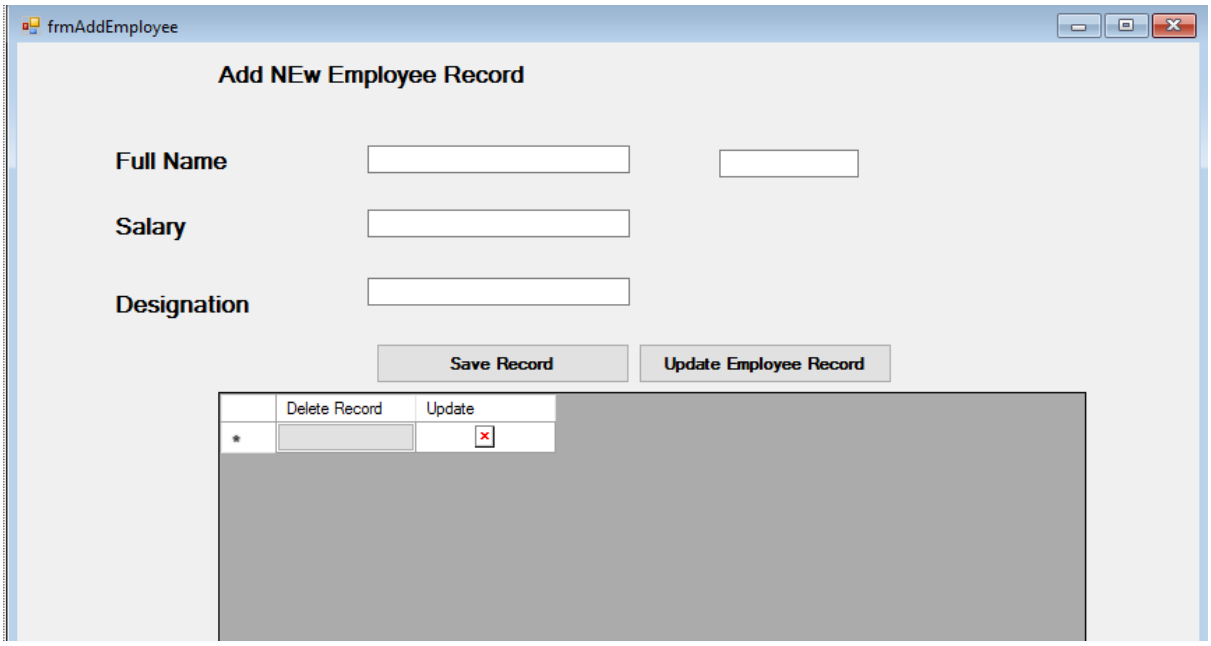
Full Name Muhammad Inam Khan 1

Salary SET

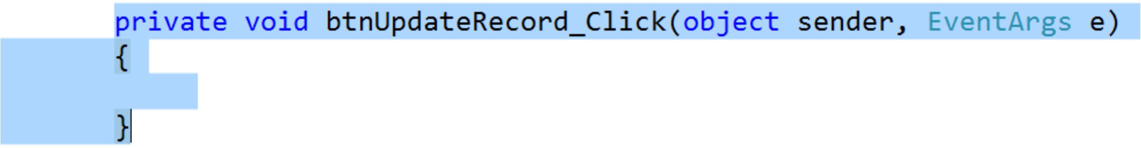
Designation 15240

	Delete Record	Update	EmployeeSID	EmployeeName	EmpSalary	Designaiton
▸	Delete		1	Muhammad Inam...	15240	SET
	Delete		2	Abdullah	15240	PST
	Delete		3	Bilal Khan	15240	Drwaing Master
	Delete		4	Muhammad Farooq	15240	Head TEacher
	Delete		5	Kamran Aziz	15240	Senior Instructor
	Delete		6	Tariq Aziz	15240	Lecturer
*		☒				

8. Next, add a button to the form, change its Text to Update and double click on button to create its Click event as shown



The Click Event is shown



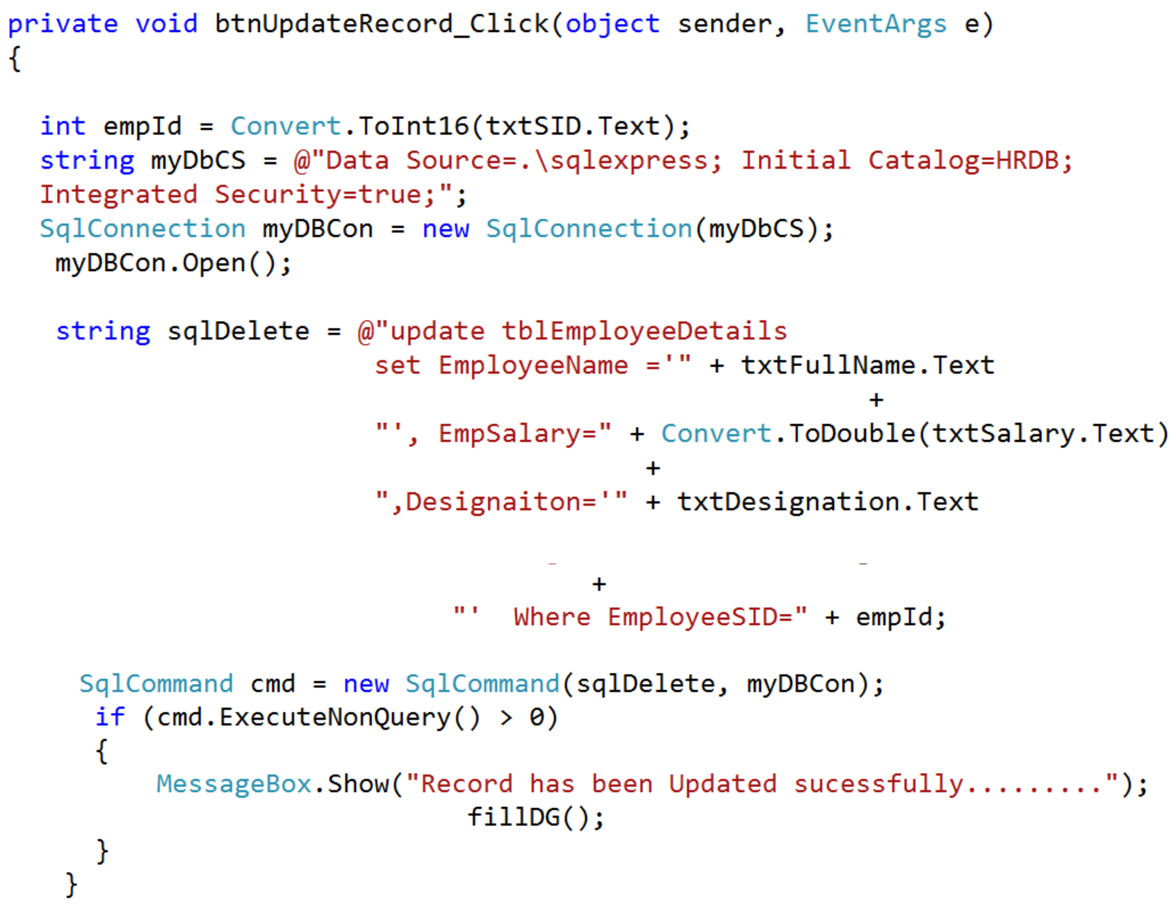
```
        private void btnUpdateRecord_Click(object sender, EventArgs e)
        {

        }
```

9. Add the following code inside the curly braces in the above **E**vent

{

//code

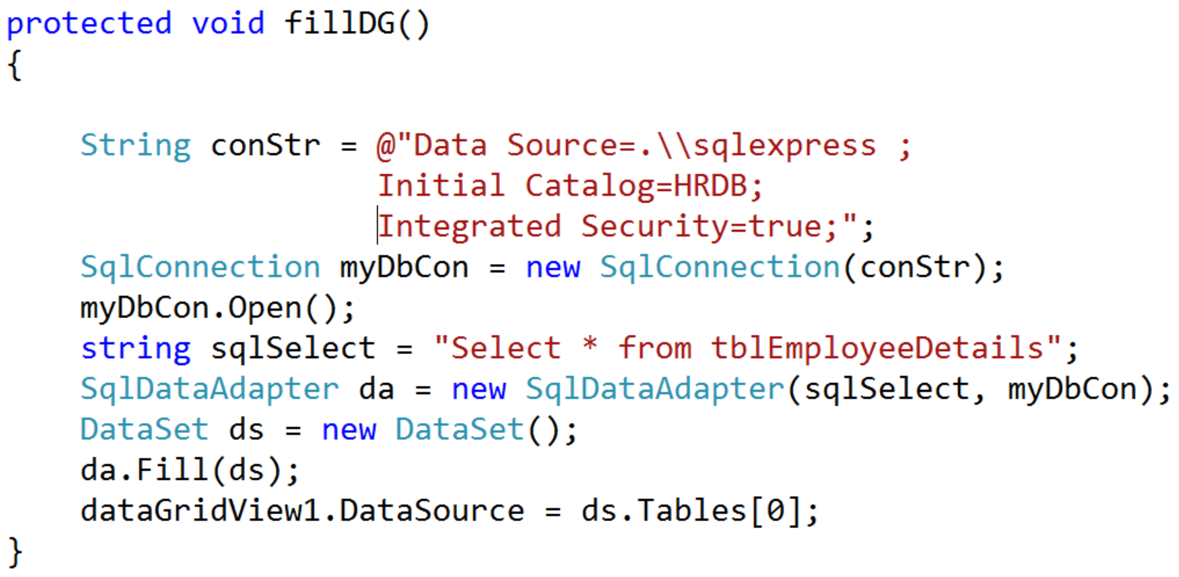
}



```
private void btnUpdateRecord_Click(object sender, EventArgs e)
{

  int empId = Convert.ToInt16(txtSID.Text);
  string myDbCS = @"Data Source=.\sqlexpress; Initial Catalog=HRDB;
  Integrated Security=true;";
  SqlConnection myDBCon = new SqlConnection(myDbCS);
   myDBCon.Open();

   string sqlDelete = @"update tblEmployeeDetails
                    set EmployeeName ='" + txtFullName.Text
                                              +
                   "', EmpSalary=" + Convert.ToDouble(txtSalary.Text)
                                 +
                   ",Designaiton='" + txtDesignation.Text

                             +
                        "'  Where EmployeeSID=" + empId;

     SqlCommand cmd = new SqlCommand(sqlDelete, myDBCon);
      if (cmd.ExecuteNonQuery() > 0)
      {
          MessageBox.Show("Record has been Updated sucessfully.........");
                         fillDG();
      }
    }
```

Here, **fillDG()** is method which is called to fill the dataGridView used on this form.

The code for **fillDG()** is shown here

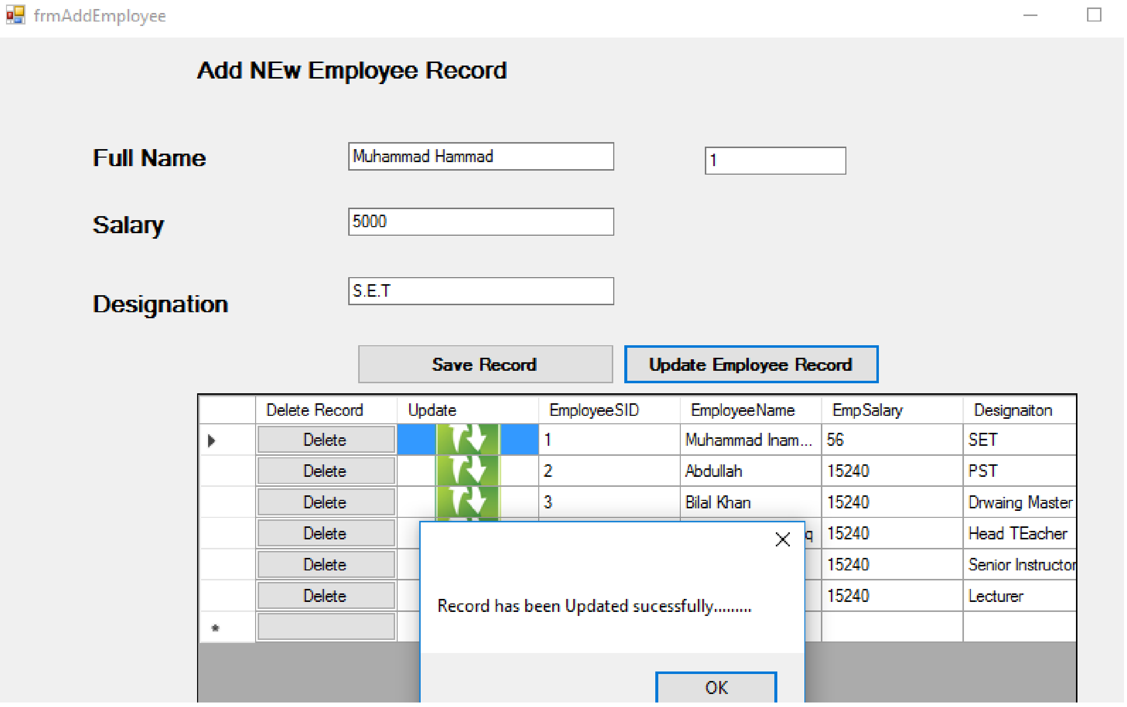


```
protected void fillDG()
{

    String conStr = @"Data Source=.\\sqlexpress ;
                      Initial Catalog=HRDB;
                      Integrated Security=true;";
    SqlConnection myDbCon = new SqlConnection(conStr);
    myDbCon.Open();
    string sqlSelect = "Select * from tblEmployeeDetails";
    SqlDataAdapter da = new SqlDataAdapter(sqlSelect, myDbCon);
    DataSet ds = new DataSet();
    da.Fill(ds);
    dataGridView1.DataSource = ds.Tables[0];
}
```

You can call the method fillDG() soon after you

a. Added a new record to database table
b. Deleted a record from database table
c. Updated a record into database table

10. Run the code



Add NEw Employee Record

Full Name: Muhammad Hammad 1

Salary: 5000

Designation: S.E.T

	Delete Record	Update	EmployeeSID	EmployeeName	EmpSalary	Designaiton
▶	Delete		1	Muhammad Inam...	56	SET
	Delete		2	Abdullah	15240	PST
	Delete		3	Bilal Khan	15240	Drwaing Master
	Delete			q	15240	Head TEacher
	Delete				15240	Senior Instructor
	Delete				15240	Lecturer
*						

Record has been Updated sucessfully.........

Hands out are prepared by

Saif Ur Rehman

Assistant Professor

University Institute of Information Technology

PMAS Arid Agriculture University

Murree Road, Rawalpindi

More Hands out on ASP.Net with C#, PHP with MySQL, Java, Programming Fundamentals and much more can be found at

www.aspuiit.blogspot.com

Suggestions are welcome at

Saif at uaar dot edu dot pk

References

[1] https://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.110).aspx

[2] http://www.academictutorials.com/microsoft.net/dotnet-framework.asp

[3]http://www.icodeguru.com/dotnet/core.c.sharp.and.dot.net/0131472275/ch01lev1sec2.html

[4] https://technet.microsoft.com/en-us/library/bb496996.aspx

[5] http://www.developerin.net/a/39-Intro-to-.Net-FrameWork/23-Components-of-.Net-Framework

[6] https://msdn.microsoft.com/en-us/library/bb822049(v=vs.100).aspx

[7] https://msdn.microsoft.com/en-us/library/aa292164(v=vs.71).aspx