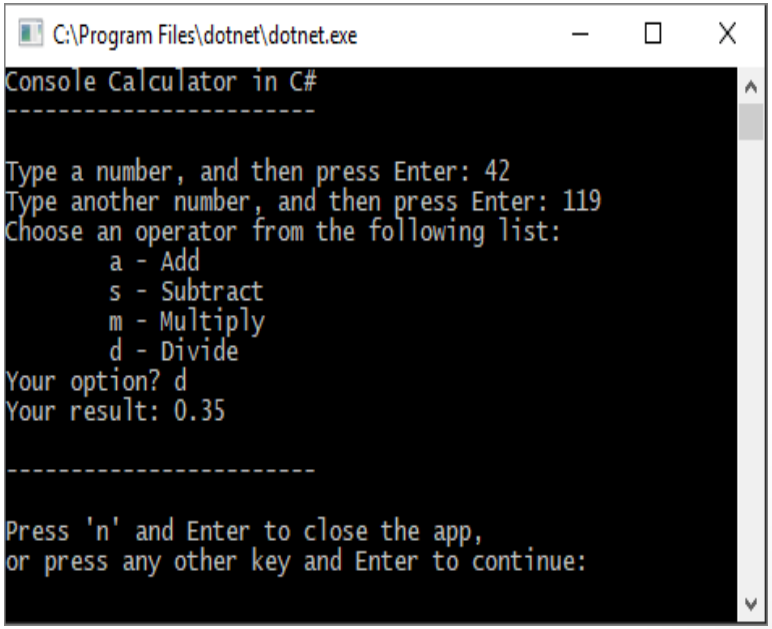
Introduction to Windows Forms Applications

GUI Introduction

- *GUI* = Graphical User Interface
- Allows the user to interact visually with a program

C# Windows Application

C# Console Application



Building GUI

GUI's are built from *GUI controls*

- Also known as *components*
- They are objects that can:
  - ✓ Display information on the screen, or
  - ✓ Enable users to interact with an application via the mouse, keyboard or other form of input
- Examples - commonly used *types of GUI controls*
  - ✓ Label, TextBox, Button, CheckBox ComboBox, ListBox

GUI in MS Visual Studio for C#: Windows Forms

- Windows Forms (or “Forms”) - used to create GUI's for C# programs

  - Create graphical elements that appear on the desktop (dialog, window, etc.)

Using Toolbox in Visual Studio to Create GUIs

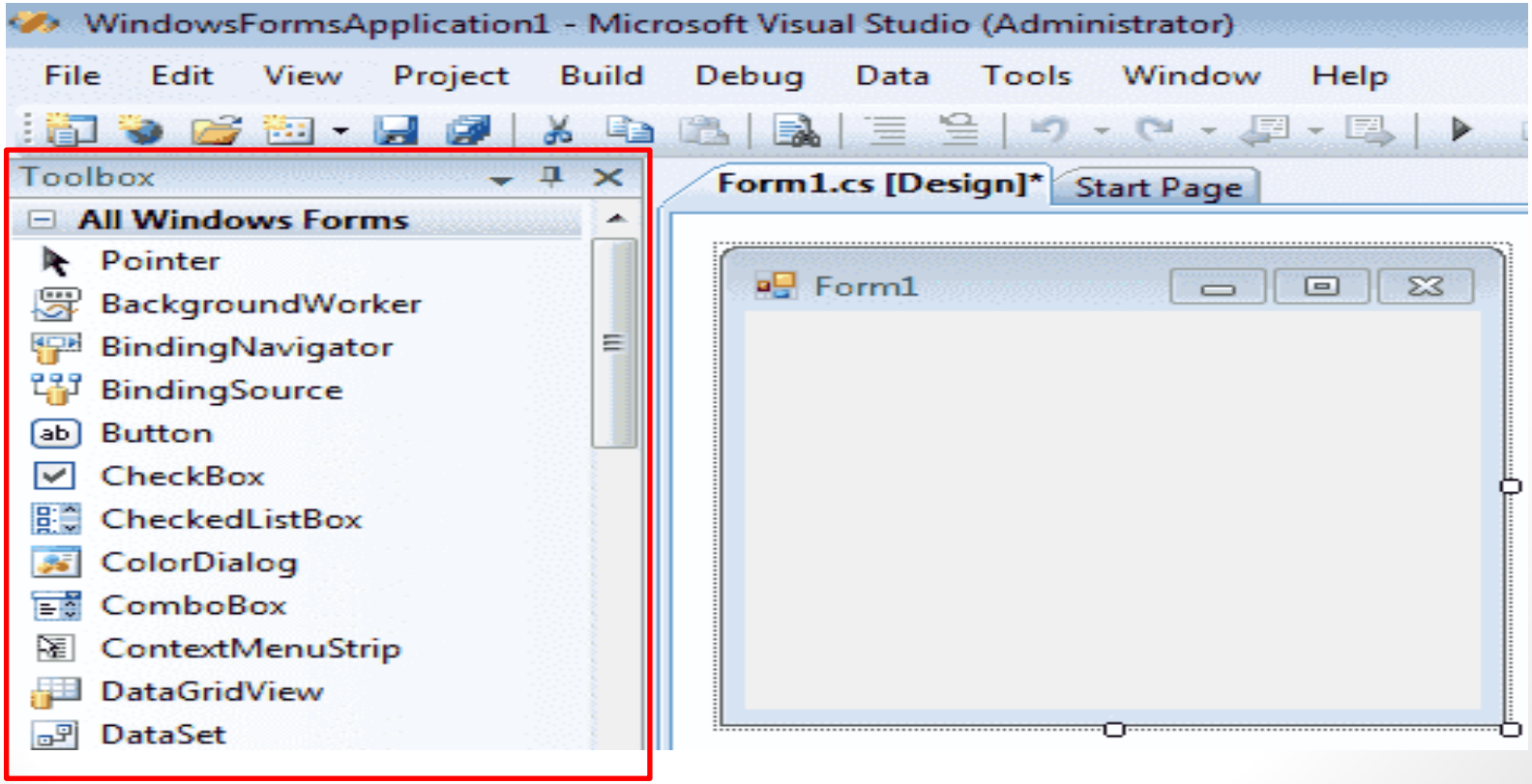
- The controls and components of C# are found in the C# Toolbox in Visual Studio
  - Organized by functionality

- To open Toolbox (takes time!):
  - Menu/Item: View>>Toolbox:

- To add a component to a Form:
  - Select that component in Toolbox
  - Drag it onto the Form

Using the Toolbox - Basics

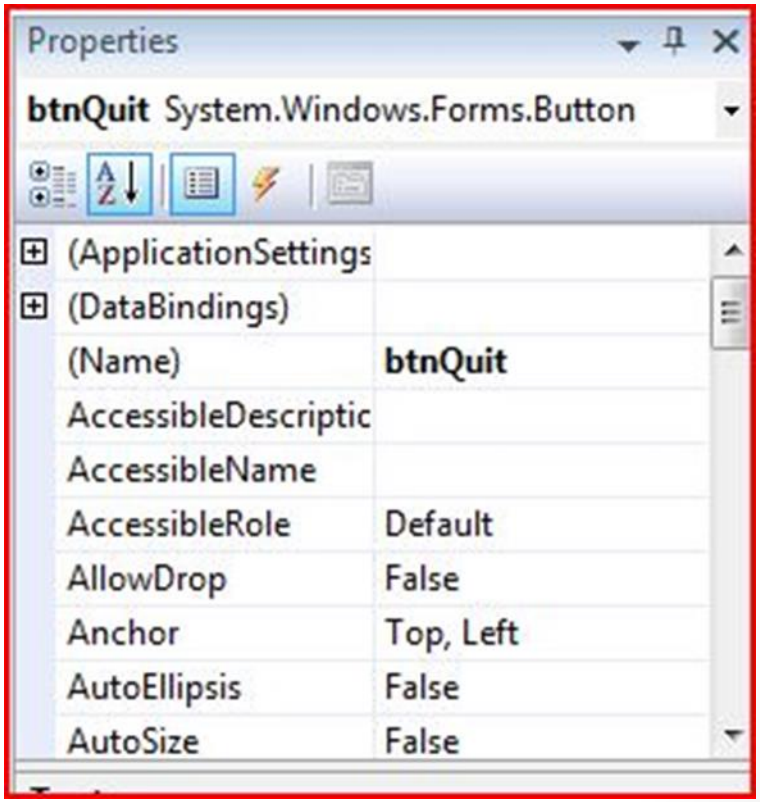
- (After you open it with: View>>Toolbox), the toolbox is located on the left-hand side of the VS screen
- Click on the control you want to add and drag it to the form



Editing the Properties

- Click on the control for which you want to change the properties

  E.g., click on form, button, label, etc.
- You can make these changes in the Properties window located on the bottom right



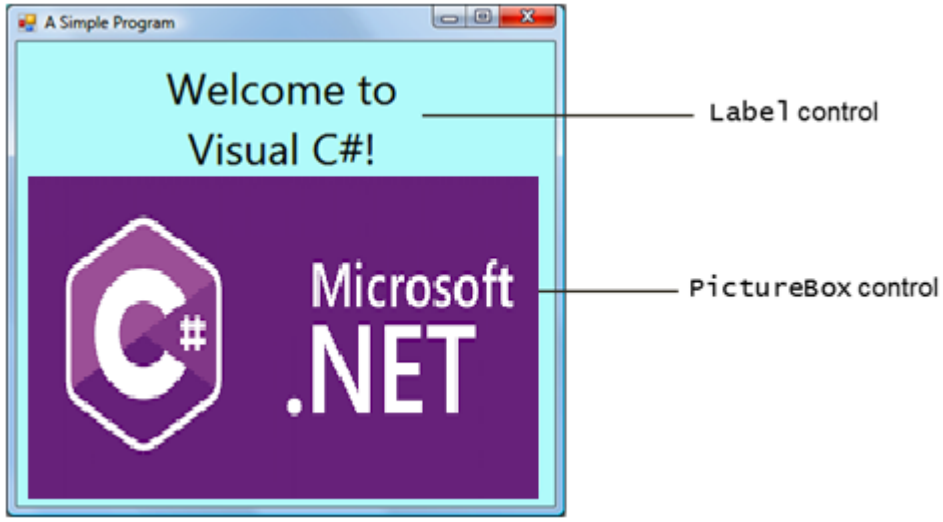
Naming Controls

- In C#, *default names* for controls/components are:
  - button1, label1, textbox1, etc.

    not very descriptive (“generic”)
- Use better, descriptive names
  - Names to have meanings that make sense

Using Visual Programming to Create a Simple Program that Displays Text and an Image

This program just displays such window:

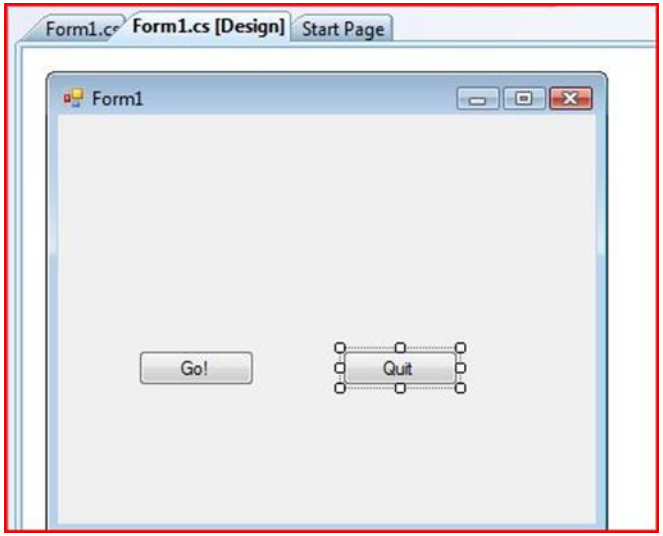


- Controls that will be used for this programs;
  - A `Label` — contains the descriptive text
  - A `PictureBox` — displays the image

Event Handling

- When a user interacts with a form, this causes an *event* to occur
  - E.g., clicking a button, typing in a textbox, etc. are events
- Events signal that certain code should be run
  - To perform some actions
- *Event Handler* = method that runs after an event occurs
- *Event Handling* = the overall process of responding to events
- All GUI controls have associated events

Event Handler for Clicking a Button



- The following code is for a button named btnQuit
  - Function: When the button is clicked, the form closes

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

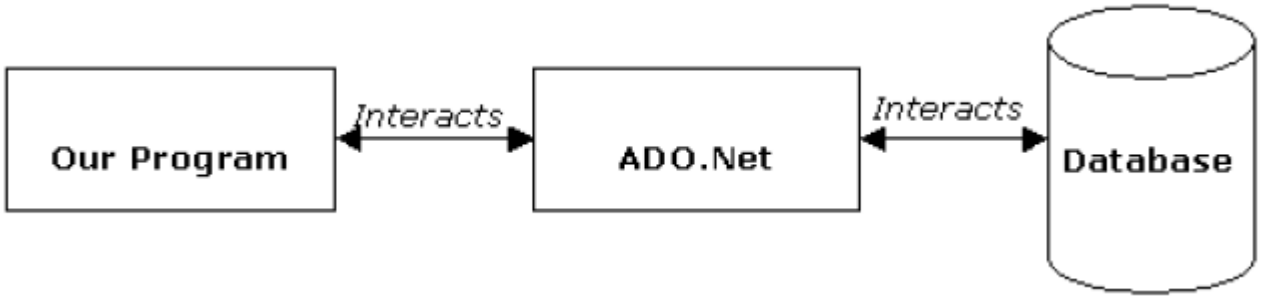
Create windows form application in C#

1. Open your Visual Studio
2. select File->New Project
3. from the new project dialog box select Other Languages->Visual C# and select Windows Forms Application.
4. Enter a project name at the bottom of the dialogue box and click OK button.

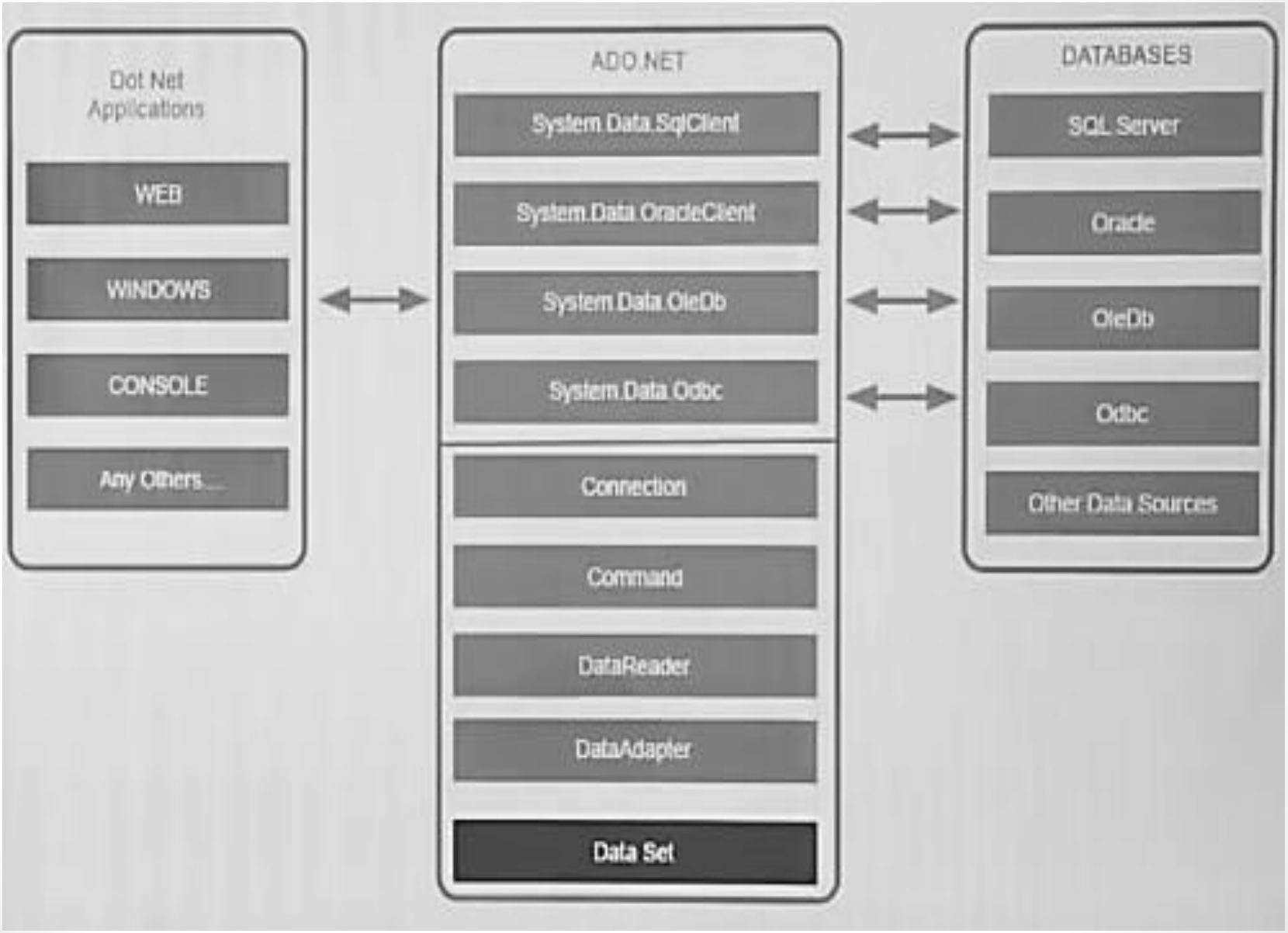
ADO.NET
Data Access and Manipulation

What is ADO.NET?

- A data-access technology that enables applications to connect to data stores and manipulate data contained in them in various ways
- Former version was ADO (ActiveX Data Object)



Understanding ADO.NET and it's class library



ADO.NET Framework Data Providers

.NET Framework data provider	Description
SQL Server	It provides data access for Microsoft SQL Server. It requires the System.Data.SqlClient namespace.
OLE DB	It is used to connect with OLE DB. It requires the System.Data.OleDb namespace.
ODBC	It is used to connect to data sources by using ODBC. It requires the System.Data.Odbc namespace.
Oracle	It is used for Oracle data sources. It uses the System.Data.OracleClient namespace.

.NET Framework Data Provider for SQL Server

- We can include this namespace in our C# application by using the following syntax.

using System.Data.SqlClient;

This namespace contains the following important classes.

Class	Description
SqlConnection	It is used to create SQL Server connection. This class cannot be inherited.
SqlCommand	It is used to execute database queries. This class cannot be inherited.
SqlDataAdapter	It represents a set of data commands and a database connection that are used to fill the DataSet. This class cannot be inherited.
SqlDataReader	It is used to read rows from a SQL Server database. This class cannot be inherited.

Steps of Data Acces : Connected Environment

1. Create connection
2. Create command (select-insert-update-delete)
3. Open connection
4. If SELECT -> use a **DataReader** to fetch data
5. If UPDATE,DELETE, INSERT -> use command object's methods
6. Close connection