Desktop Application (Windows Desktop Application) using Windows Form

* Desktop Application
  + Windows Form(s)

Word is an example of Windows Desktop Application

Window Form (A window)

* Popup Window
* Dialog Window
* Full Window
* Window has a size
* Window Application installs on Client / User PC
* No server resources will be used.
* If any Database driven application, then it will only use server resources for fetching data.
* Web Apps requires high performance server to process users request. (100KB for one user request, 1000 Users at a time, total 100,000/=KB, 98MB. But in Window App, all data stores in client PC.
* Its not simple to take data from one web page into another in Web App but in Window App all the data stores in local client machine so its easy to take from one Form into another Form.
* Window App is faster than Web App.

Dot Net Provides Windows Forms to build Windows Desktop Application but now It has introduced WPF (Windows Presentation Foundation) to build Window Desktop App.

Method:

Messagebox.Show()

Simple Calculator:

Enter First Number:

Enter Second Number:

Add Sub Mult Div Reset

Result Label

TextBox Control

* How to implement New Line in Text Box for Multi-Line Text Box
  + .Net Library: Environment.NewLine Property (Textbox1.Text = “Nabeel” + Environment.NewLine
  + Textbox1.text = “Nabeel\n\r” (Using Escape Sequence)
* Events
  + KeyDown (Whenever any key press)
* AutoComplete

Environment Class Practice

* GetFolderPath
* GetLogicalDrives
* Practice Environment Class Properties and Methods

Name, Father Name, Roll Number, Age, Total Marks.

Name: Accept (A-Za-z Space)

Roll Number: Only Aplhabets and Digits (A101)

Age and Total Marks: Only Digits

Form Show

Form Open

Create Exe File

Combo Box (Drop Down)

* Dot Net Provides uilt in Combo Box.
* It supports Maximum two Columns (One For Display and another for Value (Hidden)). Visible only Display Column.
* 3rd Party Combo Boxes provides multi Columns.

Combo Box with only One Column (Display)

Add Item into Drop Down

ComboBox1.Items.Add(object obj)

ComboBox1.Items.AddRange

Insert, InsertRange etc.

ComboBox Items can be accessed by its index position (Same as Array)

* Combobox1.items[index]

Get All Items

Text Property

* Get and Set the Combo Box Text.
* Set will also set the index position of the given Text in the Combo Box Item List if available.

SelectedText Property

* SelectedText Property will only work if Control has the focus other wise it will return Empty Text.

SelectedIndex Property

* Get and Set the SelectedIndex Item
* Returns -1 if no item is selected.
* If you set -1, means it will dis-select the selected item.

Get Selected Item Text using SelectedIndex Property

SelectedItem Property

* Returns Selected Object

Sort Property

* Boolean true = sort, false = no sort

FindString Method

* Find object in the combo box list and return its index position
* Combobox1.FindString(object obj)

ListBox Control

* Same as ComboBox Control
* ListBox supports Multiple Item Selection but Combo Box only Supports Single Item Selection.

How to Get Multiple Selected Items from ListBox

* SelectedItems //Returns Collection
* ClearSelected() Method will clear all the selection
* Clear() Method will clear (Remove) the Items from the listbox.
* SetSelected(index, bool) // Same as SelectedIndex but SetSelected can select/dis-select multiple Items.

Multiple Columns (Two Columns) in Combo Box and List Box.

* Column 1 used for Value (Hidden Text)
* Column 2 Used for Text (Display Text)

Add Method accepts Object for Additem Items.

String, Integer, Float etc. contains single Value

Class can supports multiple Fields/Properties , so we can use Class for Adding Items in the Combo and List Box.

CheckedListBox

Assignment:

Simple Point of Sale

Radio Button

* By Default all the Radio Button Controls will be grouped means only one can be selected.
* There is no Group/Ungroup property
* Use Container Control to create Multiple Radio Button Sets.
* Radio Button States
  + Checked Property which will return true or false.

Check Box

* Same as Radio Button but no grouping required
* Checked Property will return true or false
* CheckState Property will return the State out of Three States

DateTimePicker Control

* Text Property will return the selected DateTime as a string
* Value Property will also return the Selected Date Time as a DateTime Type.
* MinDate Property – Sets or get the Minimum Date for the Control
* MaxDate – Sets or get the Maximum Date for the Control.
* Format
* CustomFormat
* ShowCheckBox
* ShowUpDown
* ToString(CustomFormat)

MaskedTextBox

Windows Dialog Boxes

* MessageBox
* FolderBrowser
* OpenFile
* SaveFile
* ColorDialog
* FontDialog
* PrintDalog

FolderBrowser Dialog

* Selected Path – returns selected folder path
* Dialog Result – enum

OpenFile

* FileName
* FileNames
* Create instance in code using its class.
* Multi select prop
* Initial directory prop
* Title prop
* Filter – Creates a list of extension to filter out the result
* Filter index – to default select the extension from the list
* Checkfileexist prop
* Checkpathexist
* Showreadonly
* Readonlychecked

Save Dialog

* Same prop/methods as fileopen

Color dialog

* Color class – returns class
* Allow full open

Font dialog

* Font prop
* Color prop

Print dialog

Progress bar

Timer

* Tick event
* Interval
* Enabled