**# 1\_Basic Instructions for Microsoft Visual Studio C#**

**Created by : Yogendra Patil**

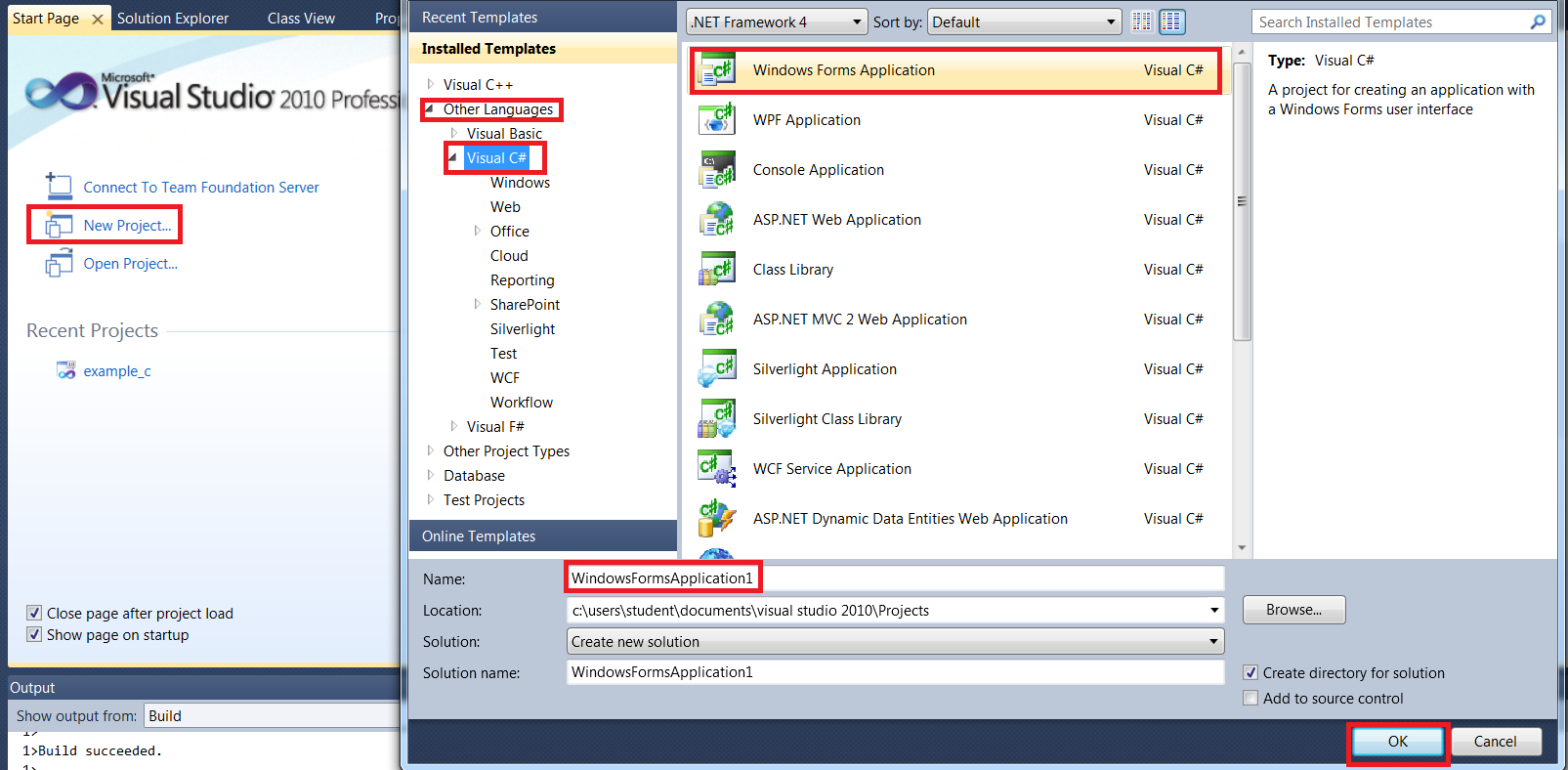
**PhD, University of Alabama**

\* LHS – Left Hand Side, M – Middle, RHS – Right Hand Side.

\* T – Top, C – Center, B – Bottom.

Start -> All Programs -> Microsoft Visual Studio 2010 -> Microsoft Visual Studio 2010.

Click -> New Project… -> Other Languages-> Visual C# -> Windows Form Application -> (Enter the file/script name) <Enter\_name> -> OK



**Video 1**:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

//Basically bunch of library files that allows proper coding

**Code2**: “Controls, Comments, Variables and Data Types”

namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void Form1\_Load(object sender, EventArgs e)

{

try

{

int var1 = 233;

//lblOutput.Text = "Hello World!";//Part I

lblOutput.Text = 5.ToString();//Convert to string for text boxes//Part II

lblOutput.Text = var1.ToString();

}

catch

{

MessageBox.Show("This is the main text", "Oops!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

finally { //The code that always executes, No point in using this

}

}

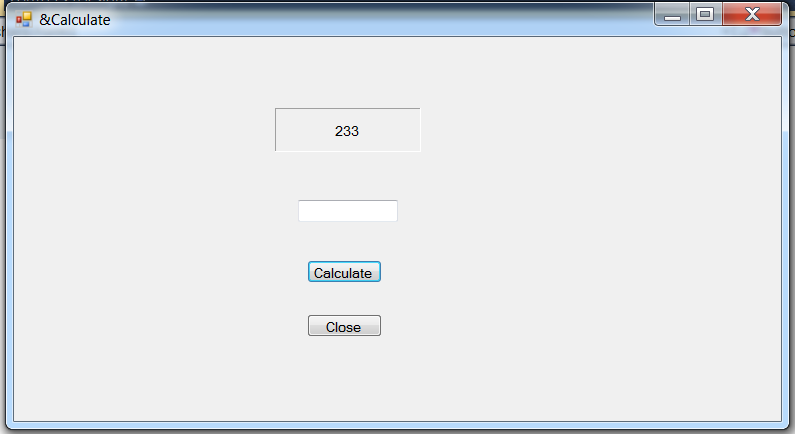
}

}

**Code 3**: “Buttons, Textboxes, Arithmetic Operators”

In “Properties” for Buttons, go to “Text” use ‘&Calculate’ in order to use shortcut ‘Alt+C’ to press the button during the code run.

In order to see code in Editor, double click on the button or text box, or whatever you have created on the GUI.



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void Form1\_Load(object sender, EventArgs e)

{

private void btnCalculate\_Click(object sender, EventArgs e)

{

try

{

double input = Convert.ToDouble(txtInput.Text);

//lblOutput.Text = input.ToString();//Part I: Code#3

lblOutput.Text = (input+2).ToString();//Part II: Arithematic Operator Code#3

}

catch {

MessageBox.Show("This is not a number", "Oops!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button1\_Click(object sender, EventArgs e)

{

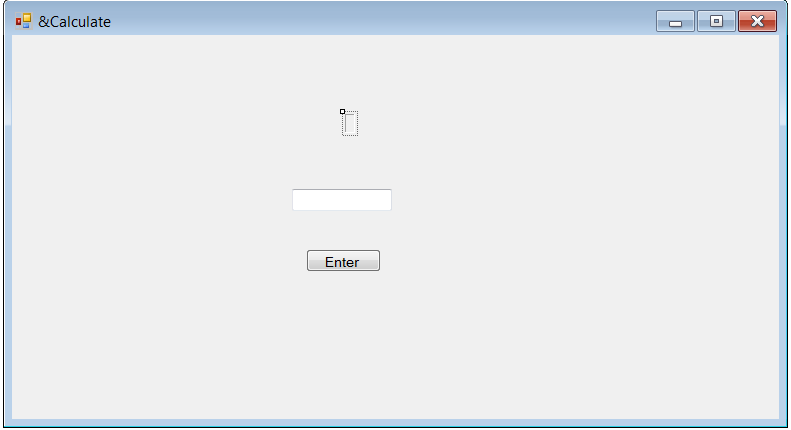
this.Close();

}

}

}

**Code4**: If Statements and Comparison and Logical Operators



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void Form1\_Load(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void btnEnter\_Click(object sender, EventArgs e)

{

int score = Convert.ToInt32(txtScore.Text);

if ((score <= 2400) && (score >= 2000)) {//Logical Operators check Boundary

lblOutput.Text = "Great Score!";

}

else {

lblOutput.Text = "Ohhh!";

}

}

private void txtScore\_TextChanged(object sender, EventArgs e)

{

}

}

}

**Code 5**: Constants, Strings, and String Functions

namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form

{

// Declare Global Parameters/Constants here

const double TAX\_FEDERAL = 15.3;//Capital letters for CONSTANTS

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btnEnter\_Click(object sender, EventArgs e)

{

//Constants ==> Work like vars but their values never change

/\* PART I: String Concatenation !!

//String Concatenation -> + +=

string myString = "Hello";

myString += " you";

lblOutput.Text = myString + " Adam"; \*/

//Part II: String Functions

// Length -> returns # of char in string

// ToUpper() -> converts all characters in string to upper case

// ToLower() -> converts all characters in string to lower case

// IndexOf('',optional start point) -> returns index number of first occurance of char or string

// Replace ('','') -> replace all occurances with given char or string

string myString = "Hello World";

lblOutput.Text = myString.Length.ToString();//output -> 11

string myString1 = "Hello World";

lblOutput.Text = myString1.ToLower();//output -> hello world

lblOutput.Text = myString.IndexOf('H').ToString();//output -> 0, -1 when the char is not there!!

lblOutput.Text = myString.IndexOf('o',5).ToString();//output -> 7

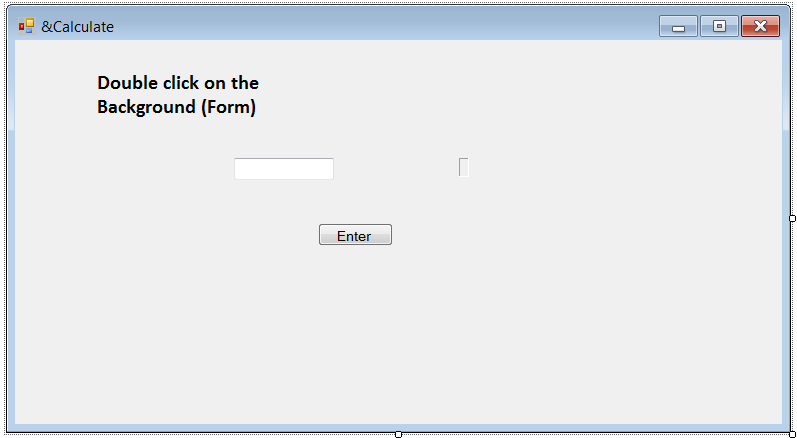
lblOutput.Text = myString.Replace('l', 'g');//output -> Heggo Worgd

}

}

}

**Code 6**: Formatting Numbers, Message Boxes, Focus and the Switch



Double click on Background (Form) to change the focus, blinking cursor! You will get following code, which should be modified. This is similar to MATLAB GUI, where at the beginning there is a short code to initialize the GUI appearance.

private void Form1\_Load(object sender, EventArgs e)

{

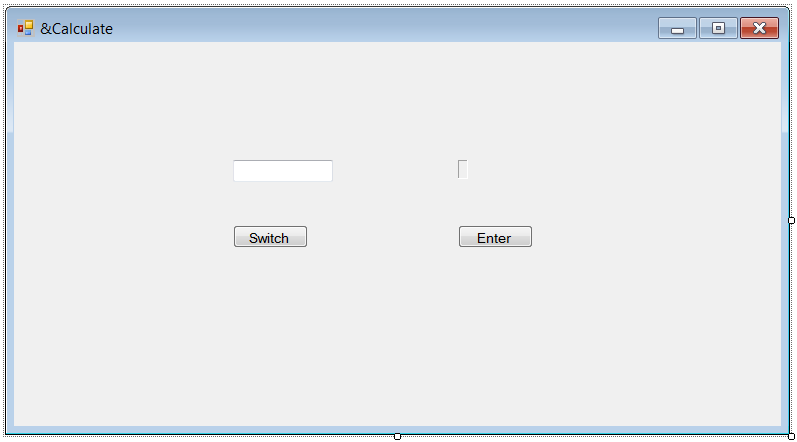
txtInput.Text = "Type Here";

//Focus -> Select() or Focus();

//txtInput.Select();//Type Here is highlighted and you can just type in

txtInput.Focus();//Type Here is highlighted and you can just type in

}



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void Form1\_Load(object sender, EventArgs e)

{

txtInput.Text = "Type Here";

//Focus -> Select() or Focus();

//txtInput.Select();//Type Here is highlighted and you can just type in

txtInput.Focus();//Type Here is highlighted and you can just type in

}

private void btnEnter\_Click(object sender, EventArgs e)

{

//Different ways of formating

//Formattiing: f(float), n(number), p(percentages), c(currency)

double input = Convert.ToDouble(txtInput.Text);

//lblOutput.Text = input.ToString("n");//input: 450000.311, output:450,000.31

//lblOutput.Text = input.ToString("n4");//input: 450000.56889, output:450,000.5689

//lblOutput.Text = input.ToString("p0");//input: 0.25, output: 25%

lblOutput.Text = input.ToString("c");//input: 4, output: $4.00

MessageBox.Show("User Input: " + input.ToString(),"Input",MessageBoxButtons.YesNo,

MessageBoxIcon.Information,MessageBoxDefaultButton.Button2);

//After you have type in the info, and click enter, what if you want to go the text box

txtInput.Text = "Type Here";

txtInput.Focus();

}

private void txtInput\_TextChanged(object sender, EventArgs e)

{

}

private void btnSwitch\_Click(object sender, EventArgs e)

{

//string userInput = txtInput.Text;//Part I

//int input = Convert.ToInt32(txtInput.Text);//Part II

char userInput1 = Convert.ToChar(txtInput.Text);//Part III: Char, check both upper case and lower case

//PART I switch statements with Strings

/\*switch (userInput){

case "Hello":

lblOutput.Text = "Why Hello to you, too!";

break;

case "Bye":

lblOutput.Text = "See ya later!";

break;

default:

lblOutput.Text = "Not Expected";

break;\*/

//PART II switch statements with numbers

/\*switch (input){

case 100:

lblOutput.Text = "Perfect Score!";

break;

case 0:

lblOutput.Text = "Too Bad";

break;

default:

lblOutput.Text = "Not Expected";

break;

}\*/

//PART III switch statements with Chars, with upper and lower case

switch (userInput1)

{

case 'A':

case 'a':

lblOutput.Text = "Perfect Score!";

break;

case 'F':

case 'f':

lblOutput.Text = "Too Bad";

break;

default:

lblOutput.Text = "Not Expected";

break;

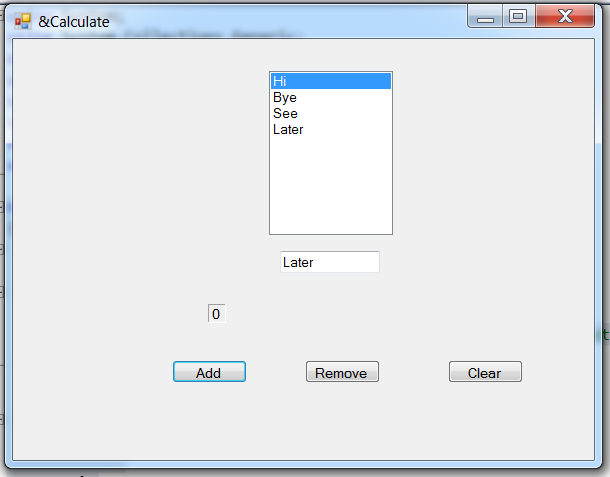
}

}

}

}

**Code 7**: GUI – List Boxes



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btnAdd\_Click(object sender, EventArgs e)

{

try

{

//List Box Functions

//name\_of\_lst.Items.Add(); -> returns an item to list box

//name\_of\_lst.Items.Count; -> returns # of elements in list box

//name\_of\_lst.SelectedItem; -> returns number/string in th elist box

//name\_of\_lst.SelectedText; -> returns index number to whatever is selected

//name\_of\_lst.Items.Remove(); -> removes items at selected index number

//name\_of\_lst.Items.Clear(); -> removes all items in list box

string item = txtInput.Text;

lstCollection.Items.Add(item);

lblOutput.Text = lstCollection.Items.Count.ToString();

}

catch

{

MessageBox.Show("Vital Error", "Oops!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void lstCollection\_SelectedIndexChanged(object sender, EventArgs e)

{

//lblOutput.Text = lstCollection.SelectedItem.ToString();//Show the selected string in the text output

lblOutput.Text = lstCollection.SelectedIndex.ToString();//Show the selected string serial # in the text output

}

private void btnRemove\_Click(object sender, EventArgs e)

{

lstCollection.Items.RemoveAt(lstCollection.SelectedIndex);

}

private void btnClear\_Click(object sender, EventArgs e)

{

lstCollection.Items.Clear();

}

}

}

**Code 8**: Loops

namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

//Tutorial::8 when you wnat to execute certain piece of code again and again //

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btnAdd\_Click(object sender, EventArgs e)

{

try

{

//Loops - while, do while, for

string name = txtInput.Text;

//Part II do-while loop

/\*int i = 0;

do{ //PART I: while(i < 10)

lstCollection.Items.Add(name);

i+=3;

} while (i > 10);// <--- for do while use semi-colon,//Also the code is executed only once??\*/

for (int i = 0; i < 5; i++) {

lstCollection.Items.Add((i + 1) + " " + name);

}

}

catch

{

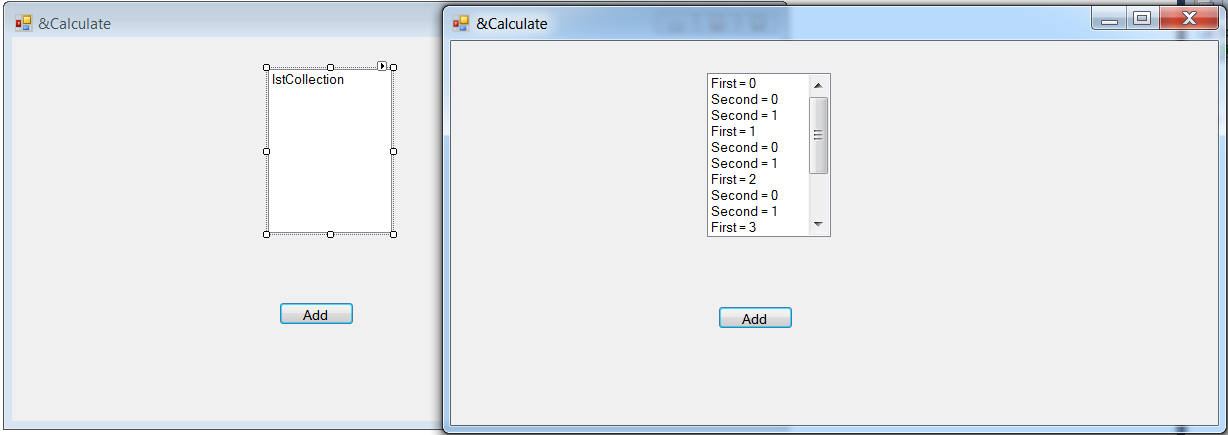
MessageBox.Show("Vital Error", "Oops!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

**Code 9**: Nesting, Existing and Continuing Loops



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btnAdd\_Click(object sender, EventArgs e)

{

try

{

int i = 0;

while (i < 5) {

lstCollection.Items.Add("First = " + i);

for (int j = 0; j < 5; j++) {

if (j == 2) {

//continue;//PART I: will skip the following lines for the given loop for the given condition

break;//PART II: break will permanently get out of the loop after j equals to 2,

}

lstCollection.Items.Add("Second = " + j);

}

i++;

}

}

catch

{

MessageBox.Show("Vital Error", "Oops!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void lstCollection\_SelectedIndexChanged(object sender, EventArgs e)

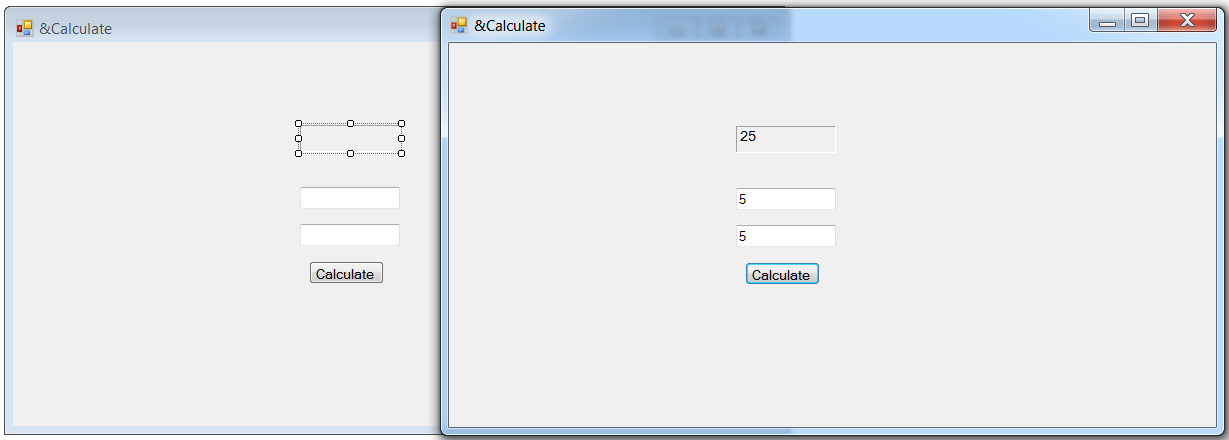
{

}

}

}

**Code 10**: Functions, Parameters, and Variable Scope



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form

{

//Here you define CONSTANTS !!

//Also here you define GLOBAL VARIABLES !!

//Here you dont have to specify public var\_name...because it is global/public by default

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

//Functions

//DataType ==> does returns value equivalent to the datatype

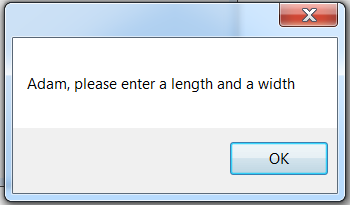
//void ==> donot return value

//Format: private datatype name\_of\_func() {}, look below they all are functions

private void print(string var1)

{ //private variables that can be declared in a given function cant be seen else where

MessageBox.Show(var1 + ", please enter a length and a width");



}

private double calculateArea(double length,double width) {

return(length \* width);

}

private void printArea(double Area) {

lblArea.Text = Area.ToString();

}

private void Form1\_Load(object sender, EventArgs e)

{

string name = "Adam";

print(name);

}

private void btnCalc\_Click(object sender, EventArgs e)

{

try

{

double userLength = Convert.ToDouble(txtLength.Text);

double userWidth = Convert.ToDouble(txtWidth.Text);

double Area = calculateArea(userLength, userWidth);

printArea(Area);

}

catch

{

MessageBox.Show("Vital Error", "Oops!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void lblArea\_Click(object sender, EventArgs e)

{

}

}

}

**Code 11**: Passing by Reference and Passing function as Parameters

// PART I :: Everything same as above add a new button called “Passing” and then add the following code//

//--------Ignore all code above this Tutorial 11 ------------------//

private void value(int num1) {

num1 += 5;

MessageBox.Show(num1.ToString());

}

private void reference(ref int num1)//type ref before datatype!!

{

num1 += 5;

MessageBox.Show(num1.ToString());

}

private void btnPassing\_Click(object sender, EventArgs e)

{

int z = 0;

value(z);

MessageBox.Show(z.ToString());

reference(ref z);//type ref to pass by reference//reference the address, so value at that address is changed

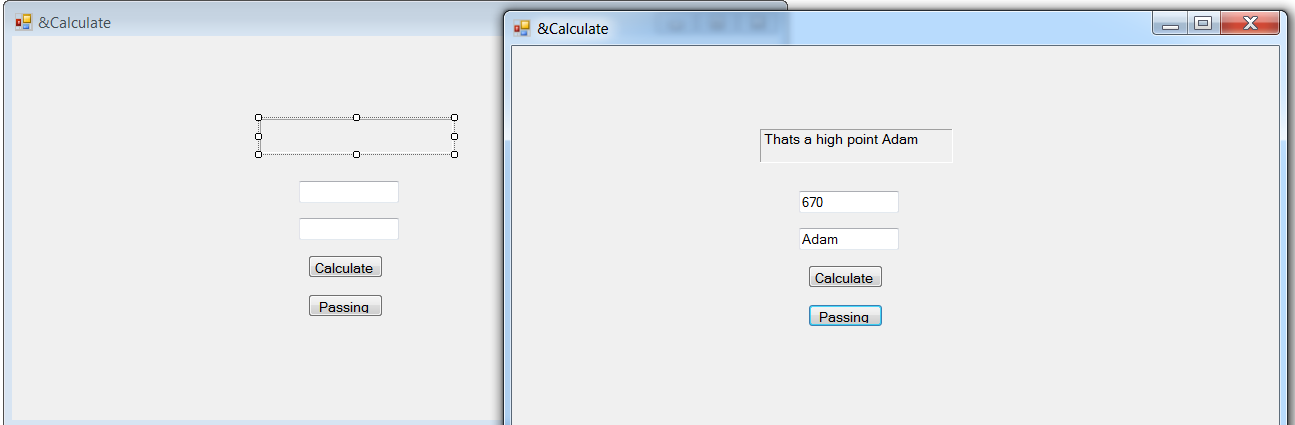
MessageBox.Show(z.ToString());

}

}

}

// PART II::



//--------Ignore all code above this Tutorial 11 ------------------//

private int mountain(int height)

{

return (height\*2);

}

private void calculateActivity(int newHeight, string name){

if (newHeight > 1000)

{

lblArea.Text = "Thats a high point " + name;

}

else {

lblArea.Text = "Not so much " + name;

}

}

private void btnPassing\_Click(object sender, EventArgs e)

{

int height = Convert.ToInt32(txtLength.Text);

string name = txtWidth.Text;

calculateActivity(mountain(height), name);//Passing a FUNCTION by reference!!

}

**Code 12**: Static Variables and Multiple Forms

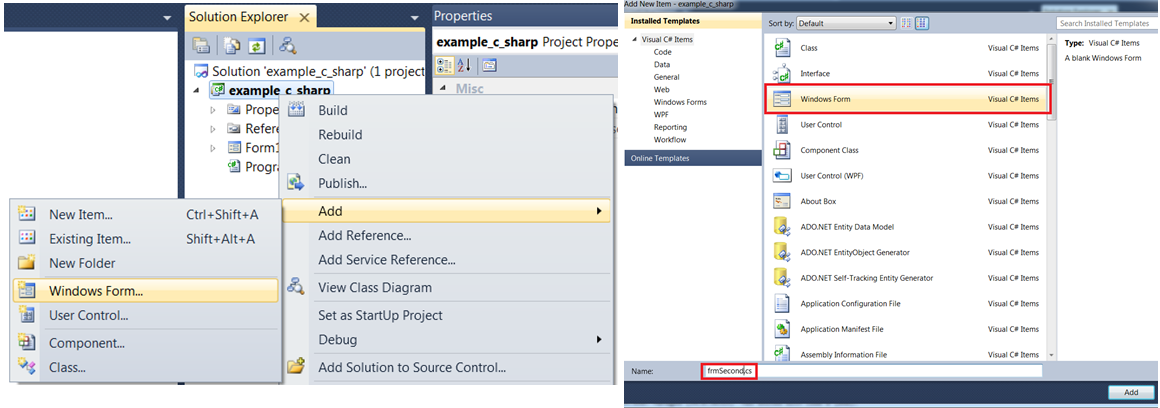
namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form //\*\* <------This is a class for Form1, you can create multiple forms (classes)

{

Right Click(example\_c\_sharp) -> Click ‘Add >’ -> Click ‘Windows Form…’



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form //\*\*This is a class for Form1

{

//Declare you static variable here

static int x;

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void Form1\_Load(object sender, EventArgs e)

{

}

private void lblArea\_Click(object sender, EventArgs e)

{

}

private void btnEnter\_Click(object sender, EventArgs e)

{

try

{

//int x = 0;//If you decalare a variable as static then its value cannot be destroyed!!

x += 10;

MessageBox.Show(x.ToString());

}

catch {

MessageBox.Show("Vital Error", "Oops!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btnOpen\_Click(object sender, EventArgs e)

{

//Since we are acessing class, we have to instantiate an object

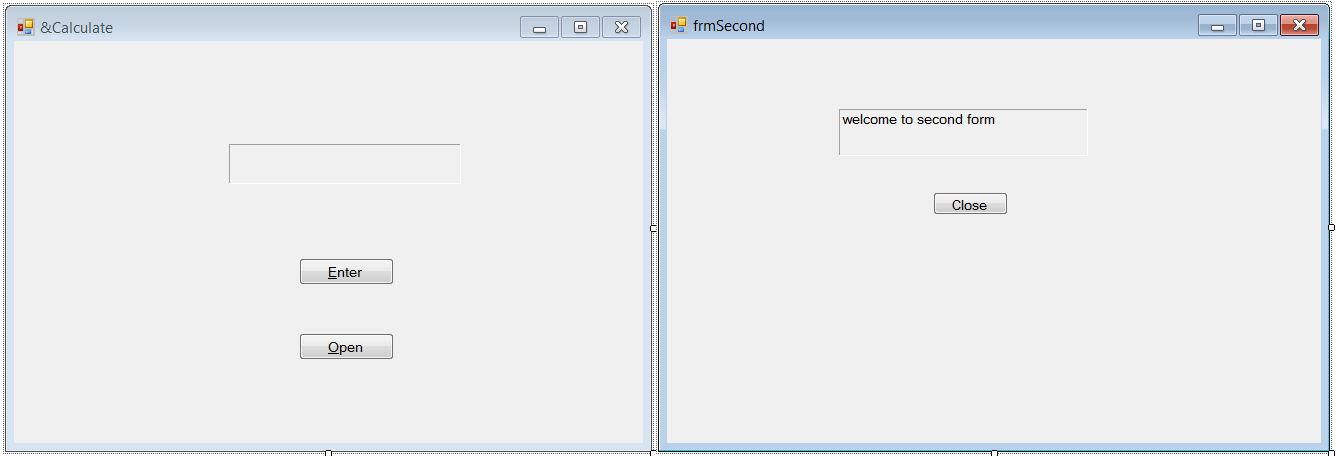
frmSecond Second = new frmSecond();

Second.ShowDialog();//By using showDialog you cannot use previous class window

}

}

}



namespace example\_c\_sharp

{

public partial class frmSecond : Form

{

public frmSecond()

{

InitializeComponent();

}

private void frmSecond\_Load(object sender, EventArgs e)

{

}

private void btnClose\_Click(object sender, EventArgs e)

{

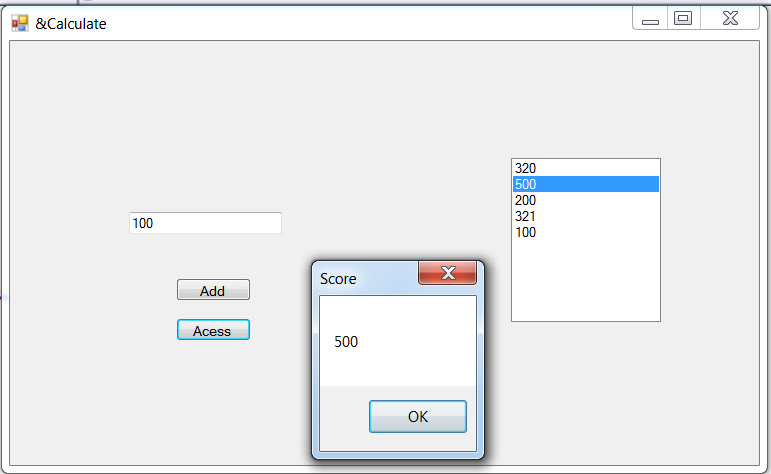
this.Close();

}

}

}

**Code 13**: Arrays



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form //\*\*This is a class for Form1

{

//Format: dataType [] nameofArray = {};

string[] countries = {"United States","Norway", "Japan", "Canada"};

//Format: dataType [] nameofArray = new dataType[5];

int[] scores = new int[5];

static int i = 0;

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void Form1\_Load(object sender, EventArgs e)

{

}

private void btnAdd\_Click(object sender, EventArgs e)

{

//MessageBox.Show(countries[0]); // to pop up a msg box, as soon as you click "Add"

int newScore = Convert.ToInt32(txtInput.Text);

if (i <= 4)

{

scores[i]= newScore;

lstStudents.Items.Add(scores[i]);

i++;

}

else {

MessageBox.Show("All Students Have Been Entered!");

}

}

private void btnAccess\_Click(object sender, EventArgs e)

{

MessageBox.Show(scores[lstStudents.SelectedIndex].ToString(),"Score"); //\*\*\*\*\*\*\*\*A new way to show highlighted string in the list box

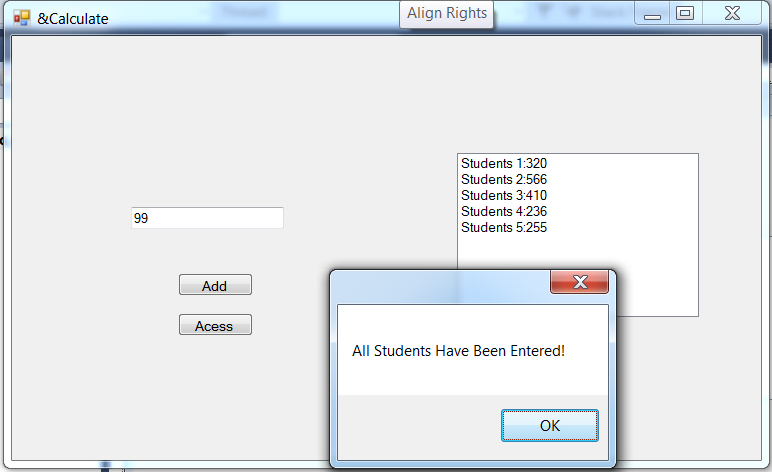
}

}

}

**Code 14**: Arrays as Parameters and Multi-Dimensional Array

//\*\*\* PART I: Arrays as Parameters\*\*\*//



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form //\*\*This is a class for Form1

{

static int i = 0;//global var

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}

//Start your coding from below

private void list(int[] scores) {

lstStudents.Items.Add("Students " + (i+1) + ":" + scores[i]);

}

//Create a function to fill Array

private void fillArray(int[] scores, int newScore)

{

if(i < 5){

scores[i] = newScore;

list(scores);//\*\*\*\*\*\*No brackets required when passing array to a function

i++;

}

else{

MessageBox.Show("All Students Have Been Entered!");

}

}

private void btnAdd\_Click(object sender, EventArgs e)

{

int[] scores = new int[5];

int newScores = Convert.ToInt32(txtInput.Text);

fillArray(scores, newScores);//\*\*\*\*\*\*No brackets required when passing array to a function

}

private void btnAccess\_Click(object sender, EventArgs e)

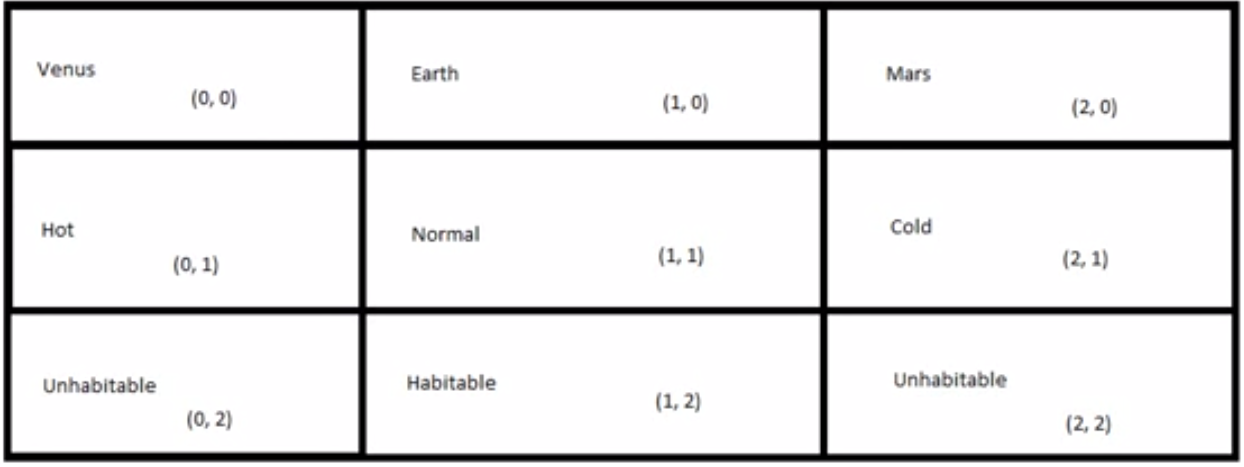
{

}

}

}

//\*\*\* PART II: Multi-Dim Arrays \*\*\*//



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form //\*\*This is a class for Form1

{

static int i = 0;//global var

//2D arrays initialization

//string[,] planets = { { "Venus", "Hot", "Unhabitatble" }, { "Earth", "Normal", "Habitable" }, { "Mars", "Cold", "Unhabitable" } };//PART 2.1 Initialize manually

string[,] planets = new string[3, 3];//PART 2.2 Different way to initialize multi-dimensional array

private void btnAccess\_Click(object sender, EventArgs e)

{

planets[0, 0] = "Venus";

planets[0, 1] = "Hot";

planets[0, 2] = "Unhabitable";

planets[1, 0] = "Earth";

planets[1, 1] = "Normal";

planets[1, 2] = "Habitable";

planets[2, 0] = "Mars";

planets[2, 1] = "Cold";

planets[2, 2] = "Unhabitable";

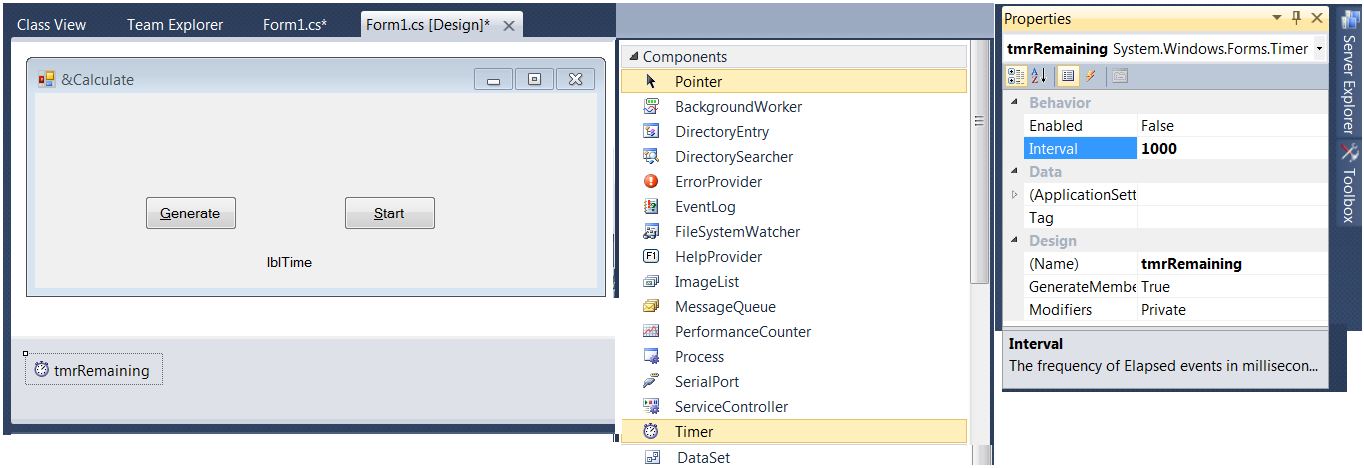
MessageBox.Show(planets[2,0]);//[0,0]-Venus,[0,1]-Hot

}

}

}

**Code 15**: Timers and Random Numbers



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form //\*\*This is a class for Form1

{

static int i = 30;

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}//------Start your coding from below-----//

private void btnStart\_Click(object sender, EventArgs e)

{

tmrRemaining.Start();

}

private void tmrRemaining\_Tick(object sender, EventArgs e)

{

if(i > 1){

lblTime.Text = i.ToString() + "Seconds Remaining";

}

else if (i == 1)

{

lblTime.Text = i.ToString() + "Seconds Remaining";

}

else {

tmrRemaining.Stop();

lblTime.Text = "You have 30 seconds";

MessageBox.Show("You are out of Time!","Too Bad", MessageBoxButtons.OKCancel,MessageBoxIcon.Exclamation);

i = 30;

}

i--;

}

private void btnGenerate\_Click(object sender, EventArgs e)

{

Random myRand = new Random();//Access random class

//MessageBox.Show(myRand.Next(0,100).ToString(), "Output");//Next(min,max)-,NextDouble-0.0 1.0

//MessageBox.Show(myRand.NextDouble().ToString(), "Output");//Next(min,max)-,NextDouble-0.0 1.0

//\*\*\* For my bytes

/\*byte[] myBytes = new Byte[3];

myRand.NextBytes(myBytes);

MessageBox.Show(BitConverter.ToString(myBytes), "Output");\*/

//Pokemon Game: Chance of hiting

int chance = myRand.Next(0, 100);

if(chance > 60){

MessageBox.Show("Hit","Output");

}

else if(chance > 40 && chance < 50){

MessageBox.Show("Critical Hit","Output");

}

else{

MessageBox.Show("Your attacked missed");

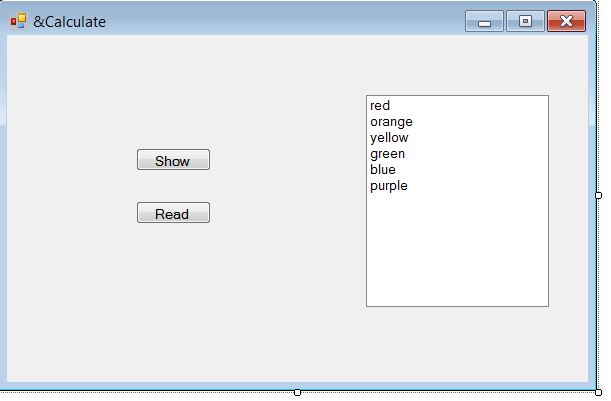
}

}

}

}

**Code 16**: Enumerations



namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form //\*\*This is a class for Form1

{

//Format Enum Data Type:: enum nameOfType {};

enum Colors : int {red,orange=2,yellow=7,green,blue,purple };//Reppresented by integers by default, e.g.red-0,blue-9, these are not strings but ints

//Also notice "" or '' are not used!!

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}//------Start your coding from below-----//

private void btnShow\_Click(object sender, EventArgs e)

{

//Colors myColor = Colors.red;//Part I shows the actual value

int myColor = (int)Colors.orange;//Part II shows the index number of red

MessageBox.Show(myColor.ToString());

}

private void btnRead\_Click(object sender, EventArgs e)

{

int selected = lstColors.SelectedIndex;//Get the selected index

Colors newColor;

switch (selected) {

case 0:

newColor = Colors.red;

MessageBox.Show("You Selected "+ newColor.ToString(), "Output", MessageBoxButtons.OK,

MessageBoxIcon.Information);

break;

case 1:

newColor = Colors.orange;

MessageBox.Show("You Selected " + newColor.ToString(), "Output", MessageBoxButtons.OK,

MessageBoxIcon.Information);

break;

case 2:

newColor = Colors.yellow;

MessageBox.Show("You Selected " + newColor.ToString(), "Output", MessageBoxButtons.OK,

MessageBoxIcon.Information);

break;

case 3:

newColor = Colors.green;

MessageBox.Show("You Selected " + newColor.ToString(), "Output", MessageBoxButtons.OK,

MessageBoxIcon.Information);

break;

case 4:

newColor = Colors.blue;

MessageBox.Show("You Selected " + newColor.ToString(), "Output", MessageBoxButtons.OK,

MessageBoxIcon.Information);

break;

case 5:

newColor = Colors.purple;

MessageBox.Show("You Selected " + newColor.ToString(), "Output", MessageBoxButtons.OK,

MessageBoxIcon.Information);

break;

default:

MessageBox.Show("Error::You Need to Select a listed Item ", "Output", MessageBoxButtons.OK,

MessageBoxIcon.Information);

break;

}

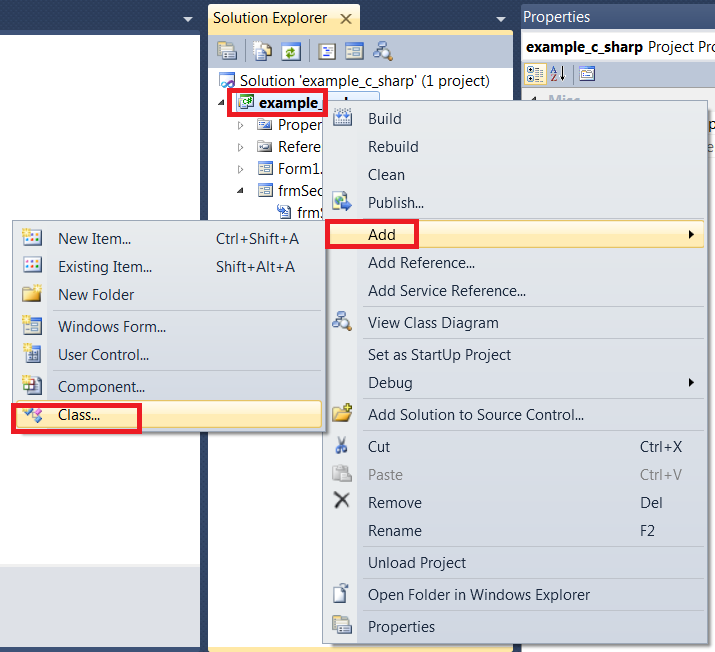
}

}

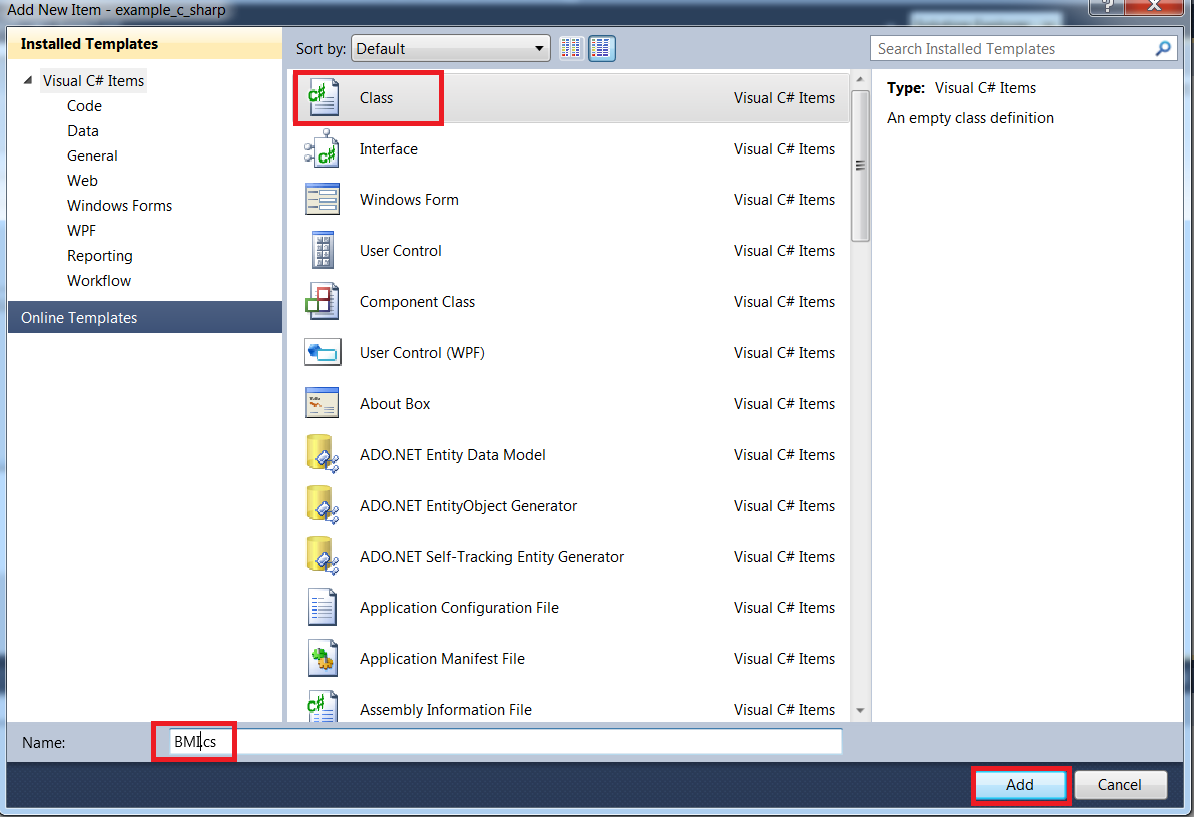
}

**Code 17-PART 1+2**: Classes and Object-Oriented Programming

Objects have characteristics and traits. Patient’s Ht. and Wt. are taken to calculate the BMI.



To create a class, right click on the project file under “Solution Explorer” tab->click “Add” > click “Class…”



Click “Class” -> Name: BMI.cs -> Add

//------------- BMI class function ----------------------//

namespace example\_c\_sharp

{

class BMI

{

// Default Constructor

public BMI()//\*\*\*To create this,type (1)**ctor**->(2)"Tab"->(3)"Tab"

{

newName = "";

newHeight = 0;

newWeight = 0.0;

}

//Overloaded Constructor

public BMI(string name,int height, double weight)

{

newName = name;//This is encapsulation, assigning the orig to temp var

newHeight = height;//This is encapsulation, assigning the orig to temp var

newWeight = weight;//This is encapsulation, assigning the orig to temp var

}

//Accesor Methods

public string getName() {

return newName;

}

public int getHeight()

{

return newHeight;

}

public double getWeight()

{

return newWeight;

}

//Mutator Functions: to change the value of main function variables

// Mutator pass parameter to function, Accessor dont

public void setName(string name) {

newName = name;

}

public void setHeight(int height) {

newHeight = height;

}

public void setWeight(double weight)

{

newWeight = weight;

}

//Function for BMI calculation

public double calculateBMI() { //dont have to create new vars, because member vars are here

return((newWeight\*703)/(newHeight\*newHeight));

}

// Member Variables

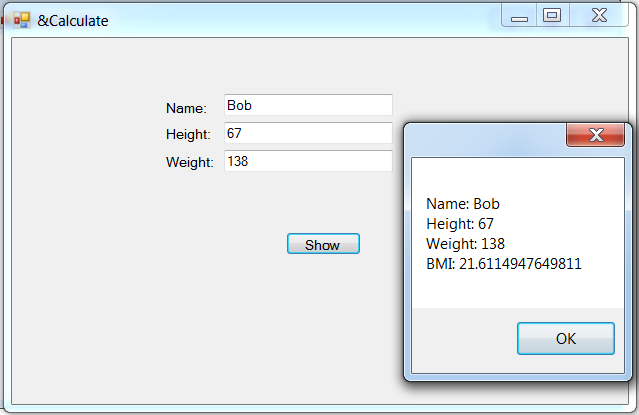
private string newName;

private int newHeight;

private double newWeight;

}

}



//------------- main function ----------------------//

namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form //\*\*This is a class for Form1

{

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}//------Start your coding from below-----//

private void btnShow\_Click(object sender, EventArgs e)

{

/\*

//Part I

BMI Patient\_1 = new BMI();//new is keyword to instantiate the class type

MessageBox.Show("Name: " + Patient\_1.getName() + Environment.NewLine +

"Height: " + Patient\_1.getHeight() + Environment.NewLine +

"Weight: " + Patient\_1.getWeight());//Name = "",Height = 0,Weight = 0.0 due to default constr func

\*/

//Part II + IV (cal BMI)

string name = txtName.Text;

int height = Convert.ToInt32(txtHeight.Text);

double weight = Convert.ToDouble(txtWeight.Text);

//Create Object

BMI Patient\_1 = new BMI(name,height,weight);//Automatically use default constructor, set member var to NULL

MessageBox.Show("Name: " + Patient\_1.getName() + Environment.NewLine +

"Height: " + Patient\_1.getHeight() + Environment.NewLine +

"Weight: " + Patient\_1.getWeight() + Environment.NewLine +

"BMI: "+Patient\_1.calculateBMI());

//Part III new object to showcase mutator function

/\*BMI Patient\_2 = new BMI();//Default Constr; Null values

Patient\_2.setName("John");

Patient\_2.setHeight(73);

Patient\_2.setWeight(175.5);

MessageBox.Show("Name: " + Patient\_2.getName() + Environment.NewLine +

"Height: " + Patient\_2.getHeight() + Environment.NewLine +

"Weight: " + Patient\_2.getWeight());

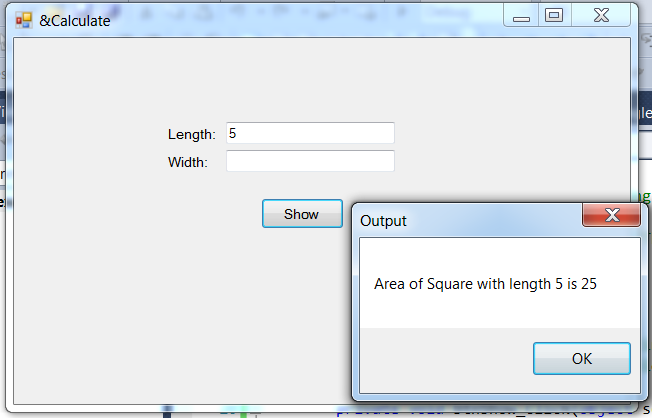
\*/

}

}

}

**Code 18**: Inheritance



//---------- Rectangle class function -----------//

namespace example\_c\_sharp

{

class Rectangle

{

//Default Constructor

public Rectangle()

{

newLength = 0;

newWidth = 0;

}

// Overload Constructor

public Rectangle(int length, int width)

{

newLength = length;

newWidth = width;

}

//Accesor

public int getLength()

{//public variable because square inherited it

return newLength;

}

private int getWidth()

{

return newWidth;

}

//Mutator Functions

public void setLength(int length) {

newLength = length;

}

private void setWidth(int width)

{

newWidth = width;

}

//cal Area

private int calculateArea() {

return (newLength \* newWidth);

}

//Member variable

public int newLength;//public variable because square inherited it

private int newWidth;

}

}

//---------- Square inherited Rectangle class function -----------//

namespace example\_c\_sharp

{

class Square : example\_c\_sharp.Rectangle

{

public int calculateArea() {//make it public to see in Form1 function

return (newLength \* newLength);

}

}

}

//---------- Main function -----------//

namespace example\_c\_sharp//Start everything below this

{

public partial class Form1 : Form //\*\*This is a class for Form1

{

public Form1()

{

InitializeComponent();// Creating initialization, donot touch this

}//------Start your coding from below-----//

private void btnShow\_Click(object sender, EventArgs e)

{

int length = Convert.ToInt32(txtLength.Text);

Square mySquare = new Square();

mySquare.setLength(length);

MessageBox.Show("Area of Square with length " + mySquare.getLength() + " is " +

mySquare.calculateArea(),"Output");

}

}

}

**Code 19**:

**Code 20**:

**Code 20**:

**Code 21**:

**Code 22**:

**Code 23**