

C# Tutorial Create a text encryption app

Welcome to this tutorial for creating an encryption app in C#. We will be using Microsoft Visual Studio 2010 and C# language. This is a windows form application so we can use buttons, labels and text boxes.

We are going to recreate an ancient encryption technology in our program, the technology is called Caesar Cipher its been used by the Great Julius Caesar of Rome who used to hide or encrypt his messages so when the messenger is caught in battle no one but himself can read those messages thus his victory was assured or so it was presumed then.

Let's look a closer exercise on the base principles of Cipher text. Below we have the 26 letters from the alphabet. If we wanted to hide the message and confuse any unauthorised person for reading what can we do?

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

We can disguise the letters with another. For example if we Shift the letter **A** 2 steps ahead it becomes C and B becomes D and C becomes E and so on. So when we want to hide a message let's look at the table below.



C	A	K	E
E	C	M	G

Table above is shifting 2 steps ahead.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

It's easier when saw in the table format? Now if we want to shift the letters to 5.

Now A becomes F, B becomes G, C becomes H and so on.



C	A	K	E
H	F	P	J

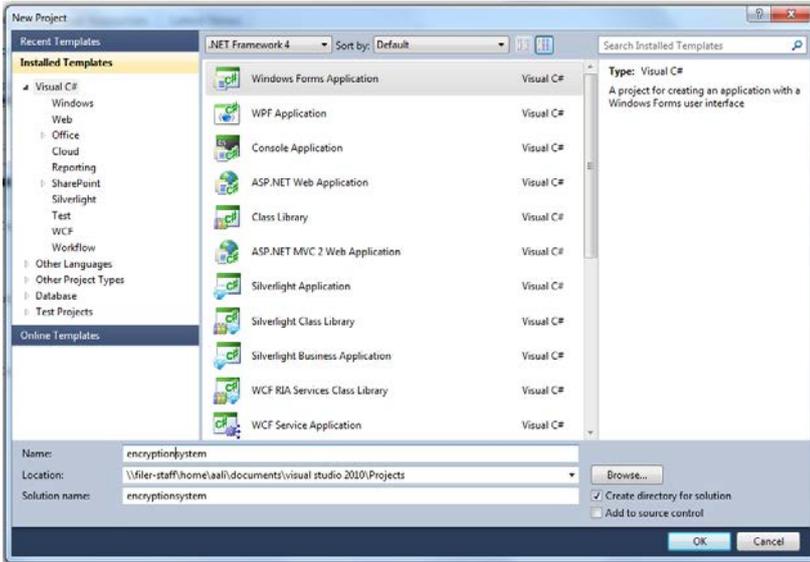
See how it changed from what it was before.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E

Look at both of the alphabets table when the bottom line ends in Z we restart it back from A. This enables this technique to make the encryption stronger.

Now here is the proposal for this project. As you can see instead of making the text from the table what if we had a program that encrypts the words for us and gives us the final result. It should also give us a function to choose how many letters we want to shift over.

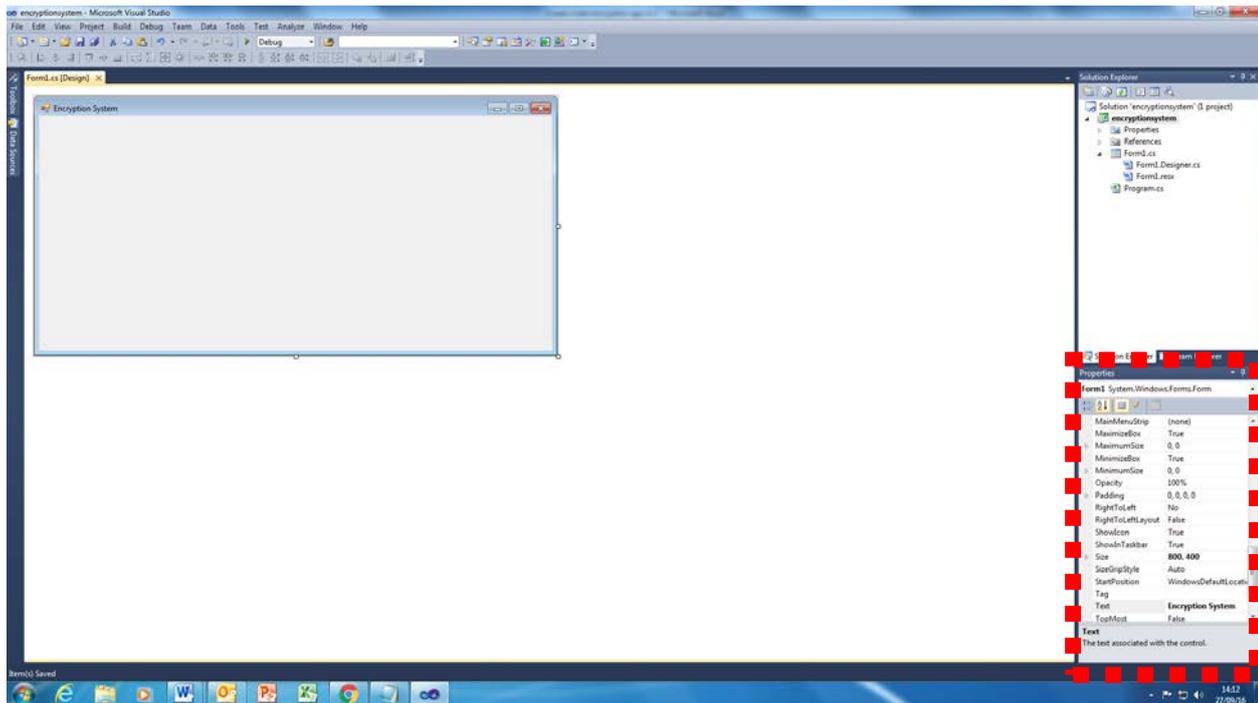
Start visual studio, make a new project (C#, Windows Form Application) and call it encryption system.



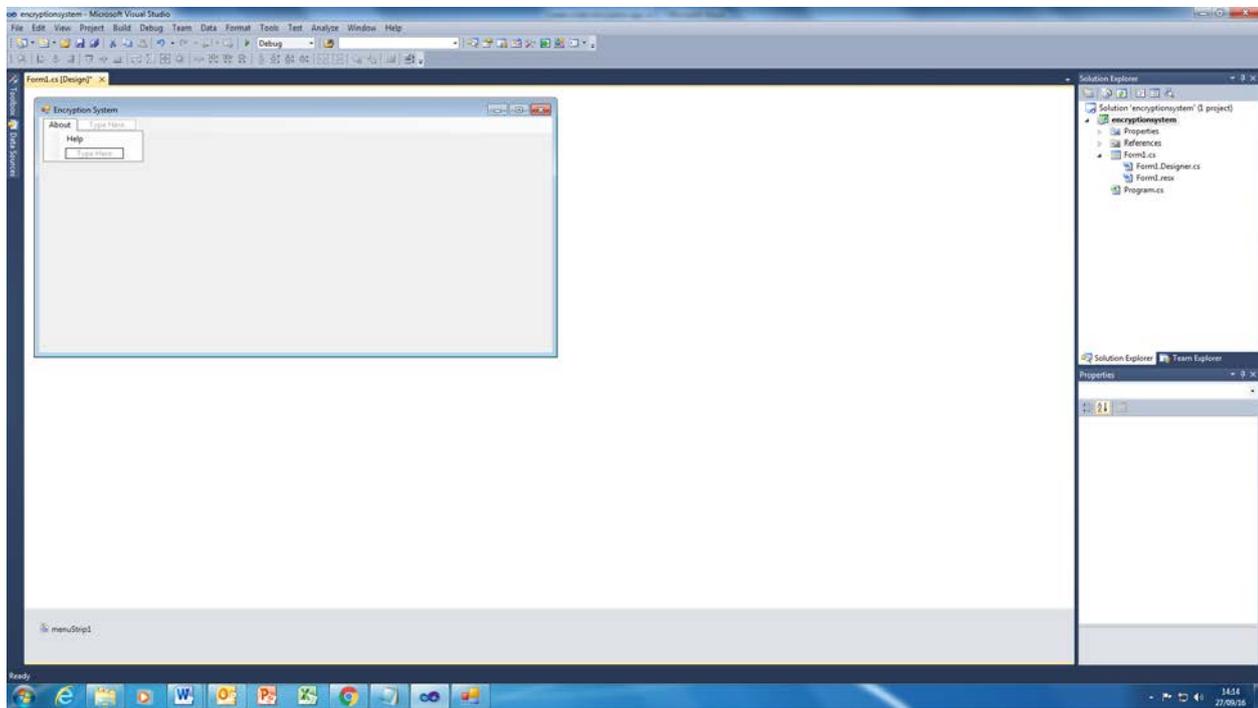
Click on the form go to the properties window and change the following

Size – 800,400 (800 pixels width and 400 pixels heights)

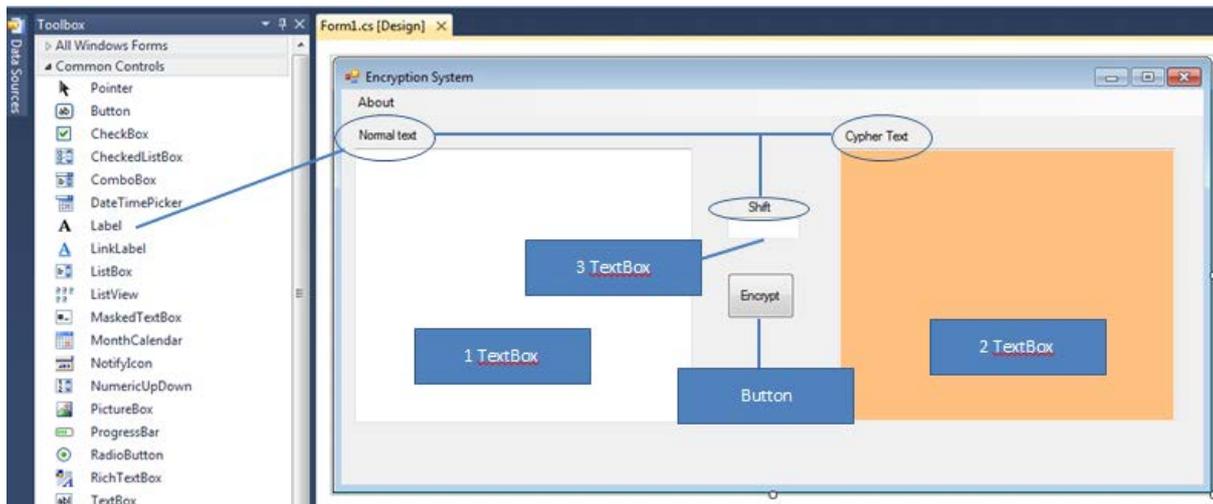
Name – Encryption system



Drag and drop a menu called MENUSTRIP from the tool bar and make a menu called About and the one under that called Help.



Lets add the rest of the components



3 Labels

1 – Change text to - Normal Text

2 – Change text to - Cypher Text

3 – Change text to - Shift

3 text boxes

1 – Going to be used for normal text, change the multi-line to true in the properties window, also change size to 307, 251 (307 pixels wide and 251 pixels tall.)

2 – going to be used to show the encrypted text. Same size and multi line as the normal text box.

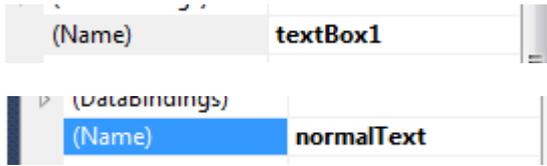
3 – going to be used for users to input the number which will shift the alphabets.

1 Button

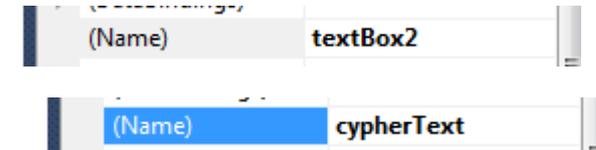
More tutorials on www.mooict.com

We will link an event to this button which will change the text and show it on the second text box.

We are going to call our components from the code so we might as well change the names of them to make sure we can remember them correctly.

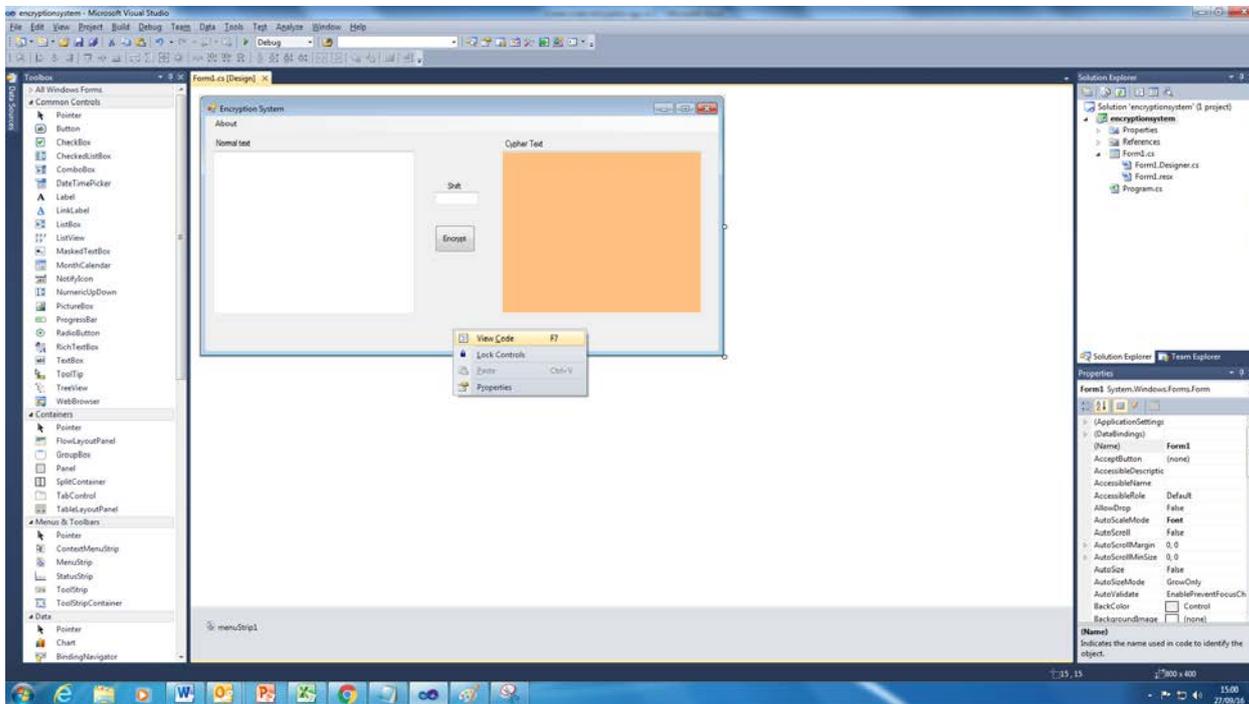


We have changed the name of the text box 1 to normal text.



We have changed the name of the text box 2 to cypher text.

Now let's proceed to the code view of the project.



Right click on the form and click on view code.

This is the view by default

```
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;

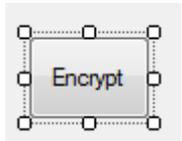
namespace encryptionssystem
{
    public partial class Form1 : Form
```

```

{
public Form1()
{
InitializeComponent();
}
}
}

```

Okay then now lets get back to the design view double click on the button.



<< double click this

```

privatevoid button1_Click(object sender, EventArgs e)
{
}

```

This code is added automatically when you double click it. This is a default click event linked to this button. Everything we want the button to do we will put it in between this function the curly brackets.

```

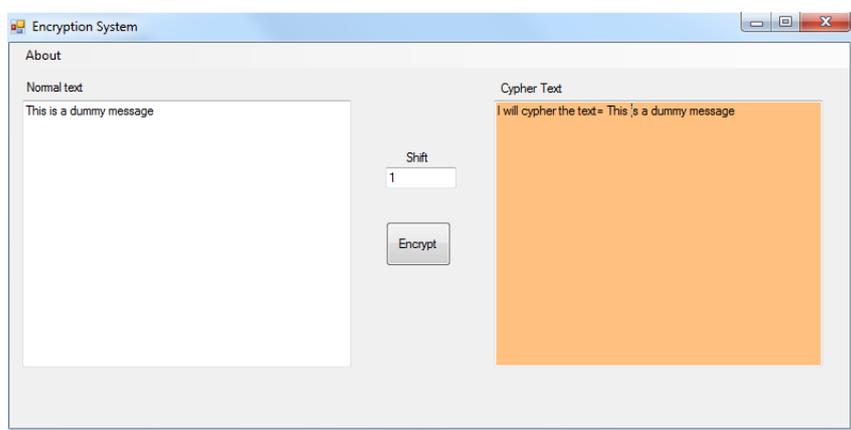
privatevoid button1_Click(object sender, EventArgs e)
{
stringorigMessage = normalText.Text;
intshiftNum = Int32.Parse(textBox3.Text);
cypherText.Text = "I will cypher the text= " + origMessage;
}

```

Lets go over these 3 lines of new code. First we are creating a local variable string called origMessage which will contain the contents of normal text box. Then we created an integer called shift num which stands for shift number. This will control how much are going to shift the letters around. Third line contains the cypher text box content. Once the button is click now it should just say I will cypher the text = "what's inside the normal text box". Remember the program is yet to be function this is only a test stage.

Before you click on the Encrypt button make sure you have a number in the shift text box, if it's empty it will crash the program.

Enter the highlighted code inside the function and run the program. .



Nice right. We know it's working so far.

One problem and you might be perfectly fine with it however I am not. It's the font size of both of the boxes. I want the text size to be bigger.

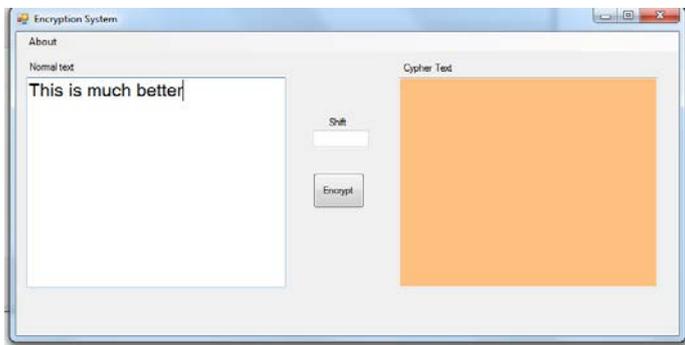
Click on the Original text box and find the font option in the properties window.



Click on the ... for the custom window to open. Change the font size to 16



Click ok.



Do the same for the Cypher text box only this one you can change it to BOLD.

I can now read the text without squinting. Good for everyone. Let's move on.

We are going to create a custom function. The custom function we will use if the one below. Take a good look at it first before we break it down. We will explain the code line by line. When you input the code please don't put the line numbers in there it will give an error.

Note - Left column is for explanation and right column is the actual code without the line numbers.

```
1) private static string doEncryption(string words, int shiftNo)
2) {
3) char[] buffer = words.ToCharArray();
4) for (int i = 0; i < buffer.Length; i++)
5) {
6) // each letter will be seperated and then
   changed.
7) char letter = buffer[i];
8) // shift the letters according to the shift
   no variable
9) letter = (char)(letter + shiftNo);
10) // Subtract 26 on overflow.
11) // Add 26 on underflow.
12) if (letter > 'z')
```

```
private static string doEncryption(string words, int shiftNo)
{
    char[] buffer = words.ToCharArray();

    for (int i = 0; i < buffer.Length; i++)
    {
        // each letter will be seperated and then changed.
        char letter = buffer[i];
        // shift the letters according to the shift no variable
        letter = (char)(letter + shiftNo);
        // Subtract 26 on overflow.
        // Add 26 on underflow.

        if (letter > 'z')
        {
            letter = (char)(letter - 26);
        }
        elseif (letter < 'a')
        {
            letter = (char)(letter + 26);
        }
        // Store.
```

```

13) {
14) letter = (char)(letter - 26);
15) }
16) elseif (letter <'a')
17) {
18) letter = (char)(letter + 26);
19) }
20) // Store.

21) buffer[i] = letter;
22) }

23) returnnewstring(buffer);
24) }

```

```

buffer[i] = letter;
}

returnnewstring(buffer);

}

```

1 – This is the line where we are declaring the custom function. First we start with **private**, this function will not be accessible from outside this class of document, then we call **static** means this function doesn't belong to any other object but it's only to itself. **String** is the data type of this function. Inside the brackets we have another string called words and an integer called.

After the function line each function starts and ends with the curly brackets { }

char[] buffer = words.ToCharArray(); in this line we are creating a character array called buffer. Char is the data type [] is the symbol for array and buffer is the name of the array. We are going to give it the value words.ToCharArray() this means the value inside the words variable will be separated into each character for example if the words CAT it will be separated to ["C", "A", "T"];

for (inti = 0; i<buffer.Length; i++)

In the line above we are running the for loop. It's a standard for loop except we are running it with the total value of the buffer array. Inside this loop we will change the characters for the encryption.

Each loop starts and stops with the curly brackets – { ... }

char letter = buffer[i];

line above we are creating a character variable letter and giving it the value from the buffer array.

letter = (char)(letter + shiftNo); On this line we are changing that letter character to a character from the shift no. So if the letter is A and shift number is 2 then this loop will change A to C since it goes from 2 up from A.

```

if (letter >'z')
{
letter = (char)(letter - 26);
}

```

In this if statement we are resetting the letters when they go to Z we want start again from A when they reach Z.

```

elseif (letter <'a')
{
letter = (char)(letter + 26);
}

```

In the else if stamen we are checking if the letter goes below A then we start back to A again.

buffer[i] = letter; In this line we are going to change the array buffer to our newly encrypted letters.

returnnewstring(buffer);

After all of it has been done we will return a newly converted letters to the output. Since this is a string type function we will return a string to the program.

How to use this function?

So the idea behind using this function is to call it when we need and then just use some values in it to test it out.

Find the function below

```
private void button1_Click(object sender, EventArgs e)
{
    string origMessage = normalText.Text;
    int shiftNum = Int32.Parse(textBox3.Text);

    cypherText.Text = "I will cypher the text= " + origMessage;
}
}
```

Make this change to it.

```
private void button1_Click(object sender, EventArgs e)
{
    string origMessage = normalText.Text;
    int shiftNum = Int32.Parse(textBox3.Text);

    cypherText.Text = doEncryption(origMessage, shiftNum);
}
}
```

So each time the button is clicked we will capture the original message, capture the shift number and then change the cypher text box with the custom function.

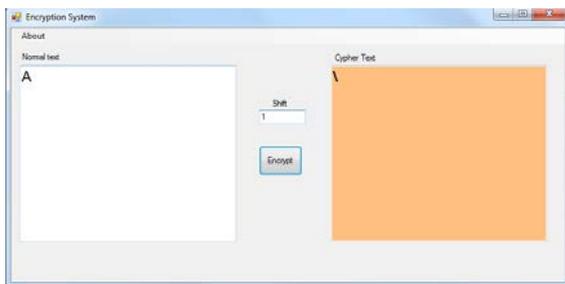
Let's test this

Original Message = A

Shift = 1

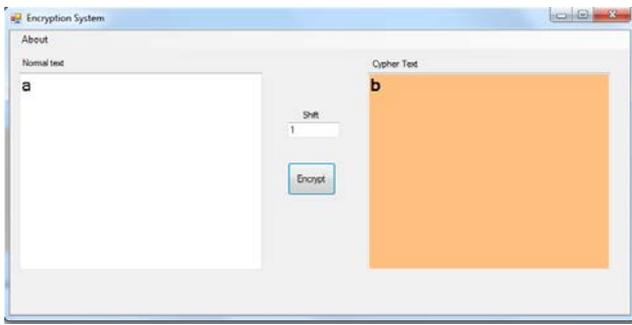
Result should be = B

Result was instead /



What?

Try again with lower case "a" this time.



Okay worked this time.

Problem is simple – computers see upper case and lower case letters differently since they all have a different ASCII code linked to them they are seen as separate characters rather than the same.

Solution to this problem – When we click encrypt we also want to change the whole thing to lower case by FORCE and then encrypt it.

How do we do this?

Very simple, remember the line below

```
cypherText.Text = doEncryption(origMessage, shiftNum);
```

Make this small change to it.

```
cypherText.Text = doEncryption(origMessage.ToLower(), shiftNum);
```

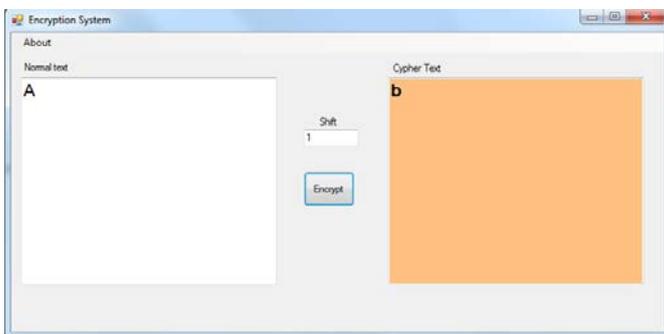
Add that ToLower() command next to the original message. This will force the words since that variable to be converted to lower case.

Let's test again

Original Message = A

Shift Number = 1

Result should be = B



Nice.

Well done.

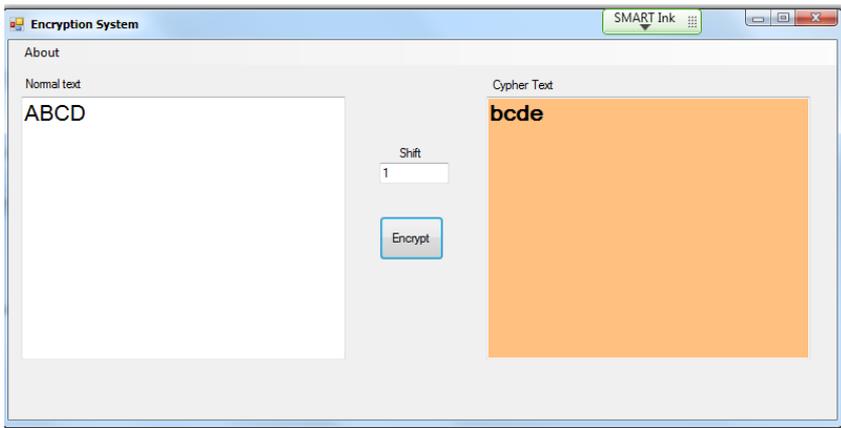
Lets test out one more

More tutorials on www.mooict.com

Original message ABCD

Shift no 1

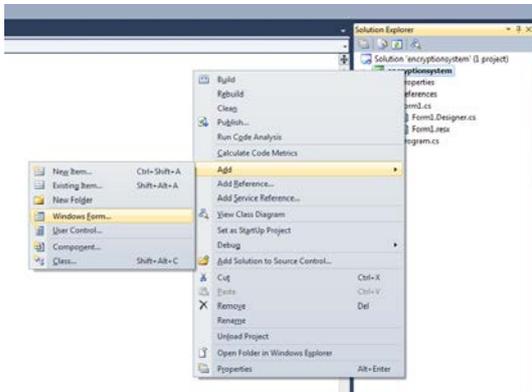
Cypher text should be = BCDE (lower case off course)



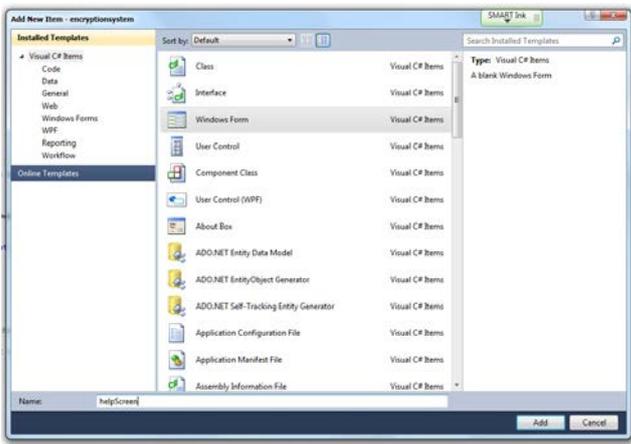
Good.

Now we need to add the help screen for user accessibility for the program.

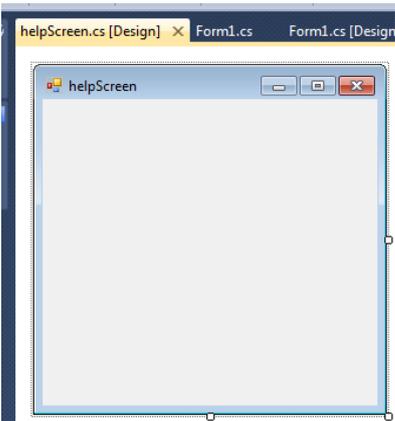
First right click on the solutions tab –



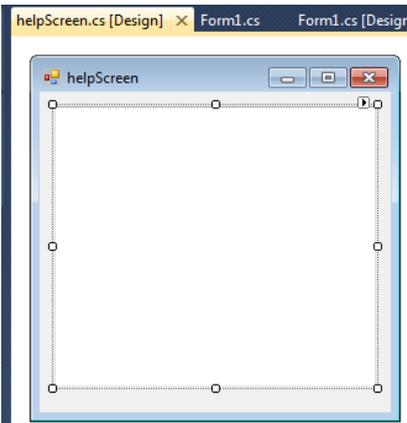
Add -> Windows Form



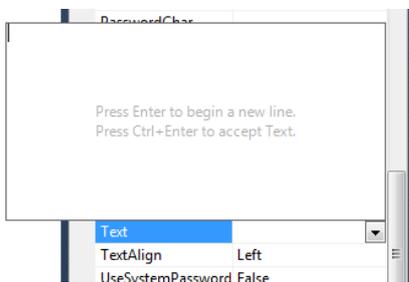
Click Add



This is the new help screen we created just now.



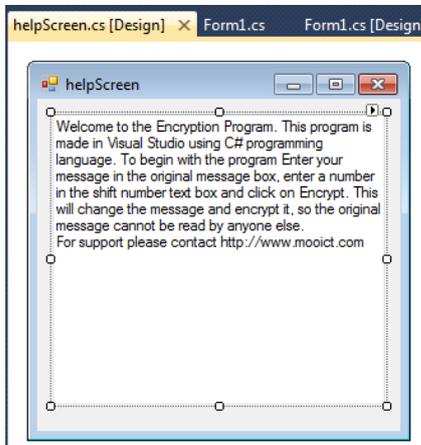
Add a text box to the form and size it to fill the whole form.



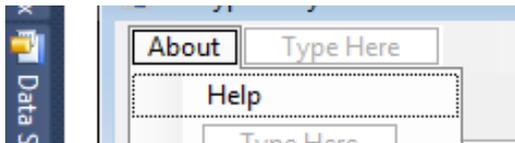
Add the following text in the properties window under the text column.

“Welcome to the Encryption Program. This program is made in Visual Studio using C# programming language. To begin with the program, enter your message in the original message box, enter a number in the shift number text box and click on Encrypt. This will change the message and encrypt it, so the original message cannot be read by anyone else.

For support please contact <http://www.mooict.com>”



Now lets go back to the form 1 and link this new form to the help button in the menu strip



Double click on the help button from the menu strip

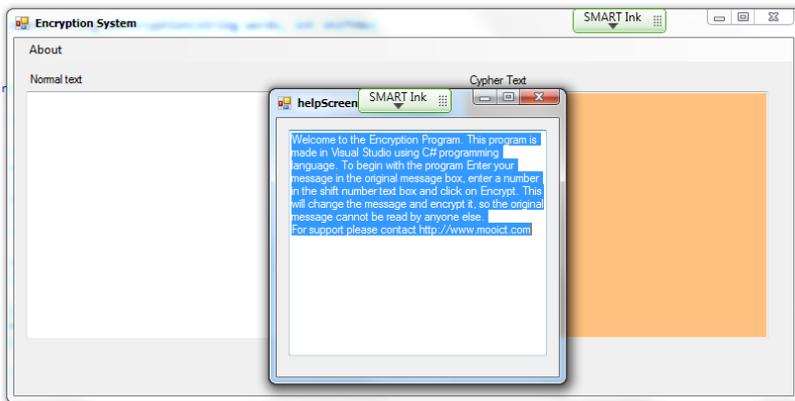
```
private void helpToolStripMenuItem_Click(object sender, EventArgs e)
{
}
```

Visual studio will automatically create and link an event to that button. What we want this button to do will go in between the curly brackets.

```
private void helpToolStripMenuItem_Click(object sender, EventArgs e)
{
    helpScreen helpForm = new helpScreen();
    helpForm.Show();
}
```

Enter those two lines of code in the function. The first line is linking the help screen we created before to this form and second line is bringing the help screen to the display of the computer.

Try it now.



Well done. Your program is now completed. However there are few other things you might want to look at.

In the shift number text box you can add numbers also you can add text so if you run it with text inside it, well it crashes. Find a way to fix that.

Reference

- 1) Dot net perls Cypher Text C# <https://www.dotnetperls.com/caesar>