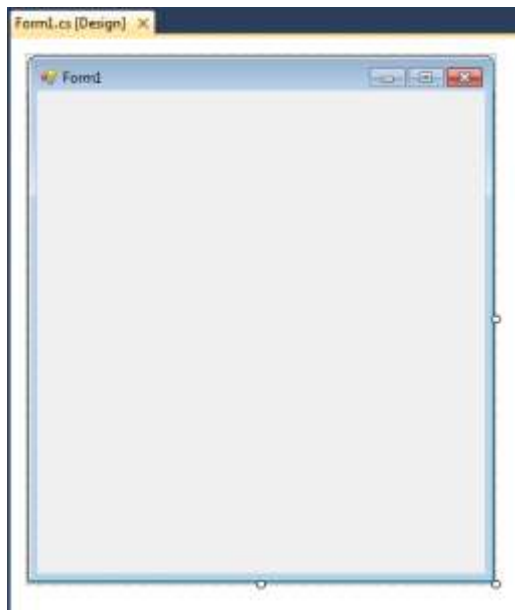


Create an Interest Calculator with C#

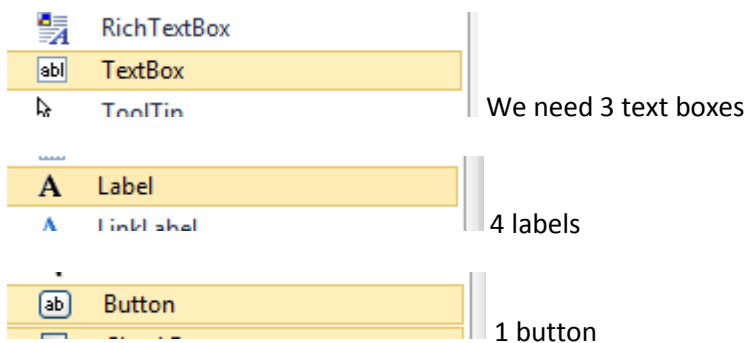
In This tutorial we will create an interest calculator in Visual Studio using C# programming Language. Programming is all about maths now we don't need to know every single algebra to learn programming what we need to do is translate the maths into understandable chunks in the code and run it for result.

This tutorial is created to bridge that gap. By doing this tutorial you will learn to create a fully working interest calculator that can work in decimal points as well as whole numbers.

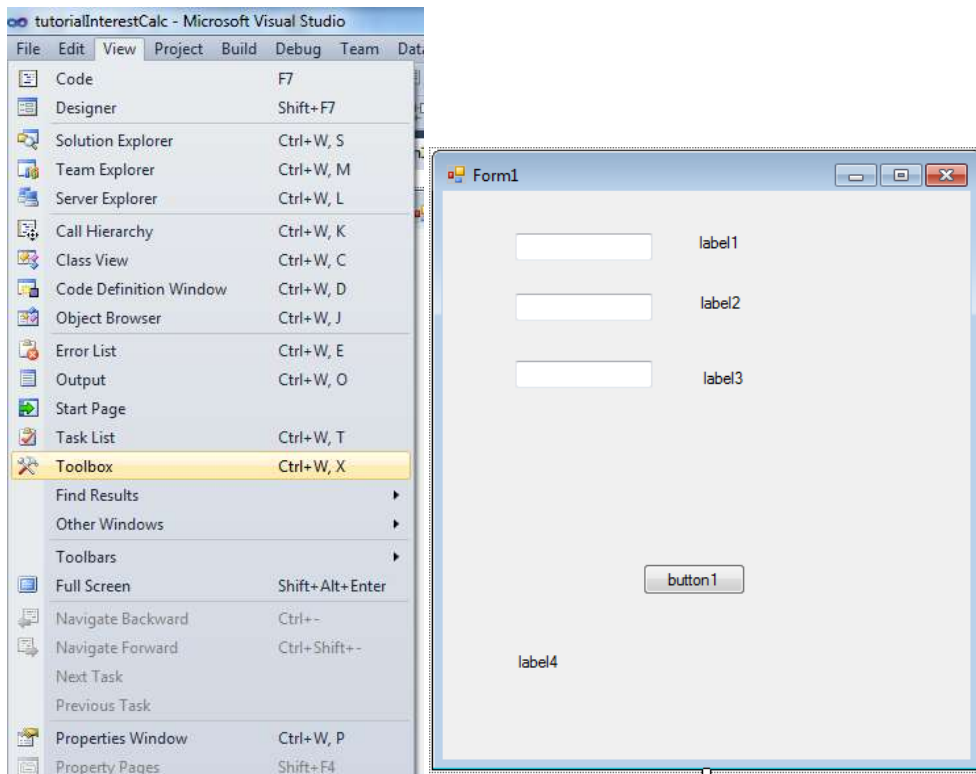
First start a new **Windows Form Application from C#** and name it **Interest Calculator**.



This is our first empty screen.

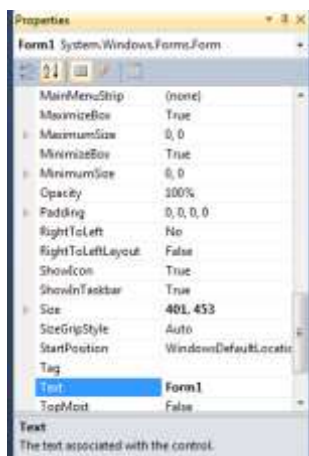
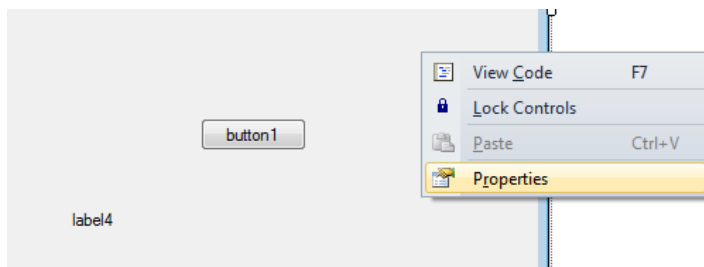


All of these components can be accessed from the Tool Bar. If you cannot see your tool bar then go to View -> Toolbar

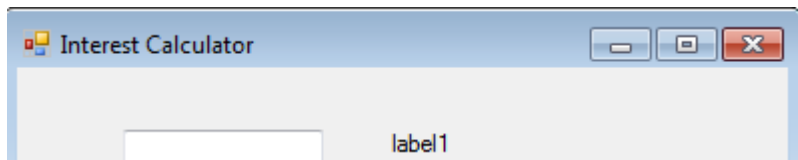


This is what our form now looks like after adding all of the components.

Now right click on the form and go to properties.



it will bring us to this little window within visual studio. Right now we are editing the properties of the form. We want to change the form text to Interest Calculator. That changes the title bar of the application.



Back to the form and start editing the other components properties.

Now click on label1 -> change text to Interest Rate

Click on label 2 -> change text to Months

Click on Label 3 -> change text to Amount

Click on button -> change text to Calculate

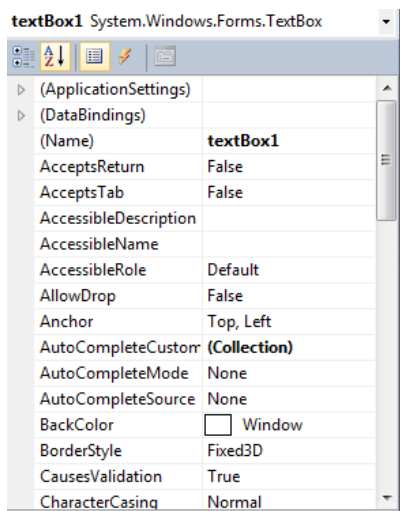
Label 4 texts leave it as is. This label will be used to show the result.



Final view of the Form.

Before we move on to the code view we need to change the NAME of those 3 text boxes so we can call them in the code easily. NAME is NOT the text property it's the (Name) which can be called through the code view.

Click on the first text box and look for this option in the properties menu.



See the name option says textBox1 that's what we need to change for all 3.

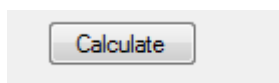
For the first text box change name to txtInterest

For the second text box change name to txtMonths

For the third text box change name to txtAmount

Once that is done. Let's go to code now.

Double click on the button which will automatically add an event linking it to that button and take us to the code view at the same time. Thank you visual studio©.



```

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 tutorialInterestCalc
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            |
        }
    }
}

```

This is the code view. Now we will need to declare a few variables above the public form1 function. Then we will return back to the button1_click function.

```
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 tutorialInterestCalc
{
    public partial class Form1 : Form
    {
        double amount;
        double months;
        double interest;
        double totalInterest;

        public Form1()
        {
            InitializeComponent();
        }

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

We will be using these 4 variables. Those are all numbers so instead of using integers we are using double WHY? Because integers only hold whole numbers for example 1, 2, 30, 36, 50 etc double on the other hand can calculate decimal numbers 6.9 100.99 etc.

Now lets capture the texts from those boxes and start the calculation.

```
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 tutorialInterestCalc
{
    public partial class Form1 : Form
    {
        double amount;
        double months;
```

```

double interest;
double totalInterest;

public Form1()
{
    InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
    interest = Convert.ToDouble(txtInterest.Text);
    months = Convert.ToDouble(txtMonths.Text);
    amount = Convert.ToDouble(txtAmount.Text);
}
}
}

```

We have declared those variables before and now we will add some values to them. We need to capture the numbers from those text boxes we added earlier. Can you see what why we gave each of them unique names. Its much easier to find the text boxes once you named them properly.

What are we doing in those 3 lines?

All three lines are doing the same thing they are converting the text box values to numbers and storing them in the variables. Now we have to convert them to numbers first because otherwise it will save them as string and we cannot do calculation with string because they are technically just text files.

We also need to convert our interest rate to decimal numbers for example if the user puts in 2% what they mean is 0.2 in decimal so we need to convert this to decimal numbers. How we are going to do it.

```

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 tutorialInterestCalc
{
    public partial class Form1 : Form
    {
        double amount;
        double months;
        double interest;
        double totalInterest;

        public Form1()
        {
            InitializeComponent();

```

```

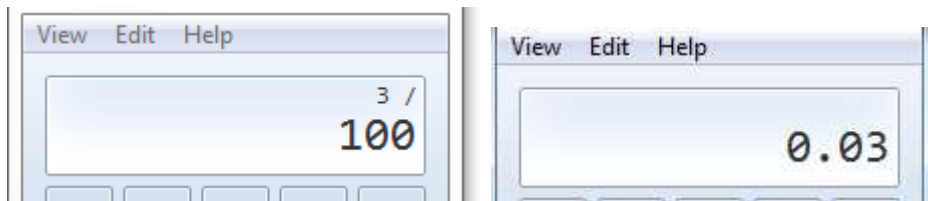
    }

    private void button1_Click(object sender, EventArgs e)
    {
        interest = Convert.ToDouble(txtInterest.Text);
        months = Convert.ToDouble(txtMonths.Text);
        amount = Convert.ToDouble(txtAmount.Text);

        interest = interest / 100;
    }
}

```

So basically we have declared the interest variable again this time we will divide it by 100. So if we get 3% interest rate inputted by the user it should come up as 0.3. Try this on the windows calculator 3 divided by 100 = 0.03



Once we saved all three values in the three variables we can do our secret maths formula to make it work.

Here is the maths equation we will need to use for this

$$C = P * ((1+r)^n - 1)$$

$$C = P [(1 + r)^n - 1]$$

C = Compound Interest

P = principal(original balance)

r = rate per period

n = number of periods

Ref [http://www.financeformulas.net/Compound_Interest.html]

Total interest rate = amount times ((1 plus interest rate) by power of months) minus 1.

Follow the link for the detailed description of this calculation.

This will give us the total month's interest in one.

So how will we do the calculations in C#? C# has some maths functions built in to already which we can use to calculate the POWER OF calculations.

More tutorials on www.mooict.com

We can do this in one line here is how

```
totalInterest = amount * (Math.Pow(1 + interest, months) - 1);
```

Note to self on this one; in the future always use variable names that are relevant to the project. Don't go for typical name like NumA numB or c. Look at the line above you can read through it and it explains the task it's conducting.

We are literally doing the calculation formula in C# we are using the Math.Pow() function which is built into C# main framework and works like a charm for this issue.

If you want to learn more about how Math.Pow() works go here

<http://www.dotnetperls.com/math-pow>

Now lets look at our code thus far

```
private void button1_Click(object sender, EventArgs e)
{
    interest = Convert.ToDouble(txtInterest.Text);
    months = Convert.ToDouble(txtMonths.Text);
    amount = Convert.ToDouble(txtAmount.Text);

    interest = interest / 100;

    totalInterest = amount * (Math.Pow(1 + interest, months) - 1);
}
```

There is our total interest calculation formula now let's display that result on label 4.

```
private void button1_Click(object sender, EventArgs e)
{
    interest = Convert.ToDouble(txtInterest.Text);
    months = Convert.ToDouble(txtMonths.Text);
    amount = Convert.ToDouble(txtAmount.Text);

    interest = interest / 100;

    totalInterest = amount * (Math.Pow(1 + interest, months) - 1);
    label4.Text = totalInterest.ToString();
}
```

In the line highlighted above you can see that we are displaying the total interest value inside the text of label 4. You notice that we are using toString() end of the total interest. This line of code is basically converting the number to a text so we can display it on screen. Remember how we converted text to double before we are reversing this right now.

Run the program now.

Input 2 for interest, 12 for months and 1000 for amount

More tutorials on www.mooict.com

2 Interest Rate

12 Months

1000 Amount

Calculate

268.241794562545

Okay that's a lot of decimal numbers. We want that number to show 268.24 not the whole line of them.

Add the following in you code.

```
private void button1_Click(object sender, EventArgs e)
{
    interest = Convert.ToDouble(txtInterest.Text);
    months = Convert.ToDouble(txtMonths.Text);
    amount = Convert.ToDouble(txtAmount.Text);

    interest = interest / 100;

    totalInterest = amount * (Math.Pow(1 + interest, months) - 1);
    label4.Text = totalInterest.ToString("#.##");
}
```

In the same line where we are displaying the final result to label 4 we are going to add `"#.##"` inside the double brackets of ToString function. This little hack allows us to format the text before displaying to the screen. Now the string will be counting only two decimal numbers after and display the results on screen.

Lets run the application again now

A screenshot of a web form with a light gray background. It contains three input fields on the left, each with a label to its right: 'Interest Rate' with the value '2', 'Months' with the value '12', and 'Amount' with the value '1000'. Below these fields is a button labeled 'Calculate' with a blue border. At the bottom left of the form area, the number '268.24' is displayed.

Happy days.

So we learned to use double variable, used actual maths formula and converted to c#, converting string to double and double back to string, not bad for a day's practise.

Well done you.

Moo on.