

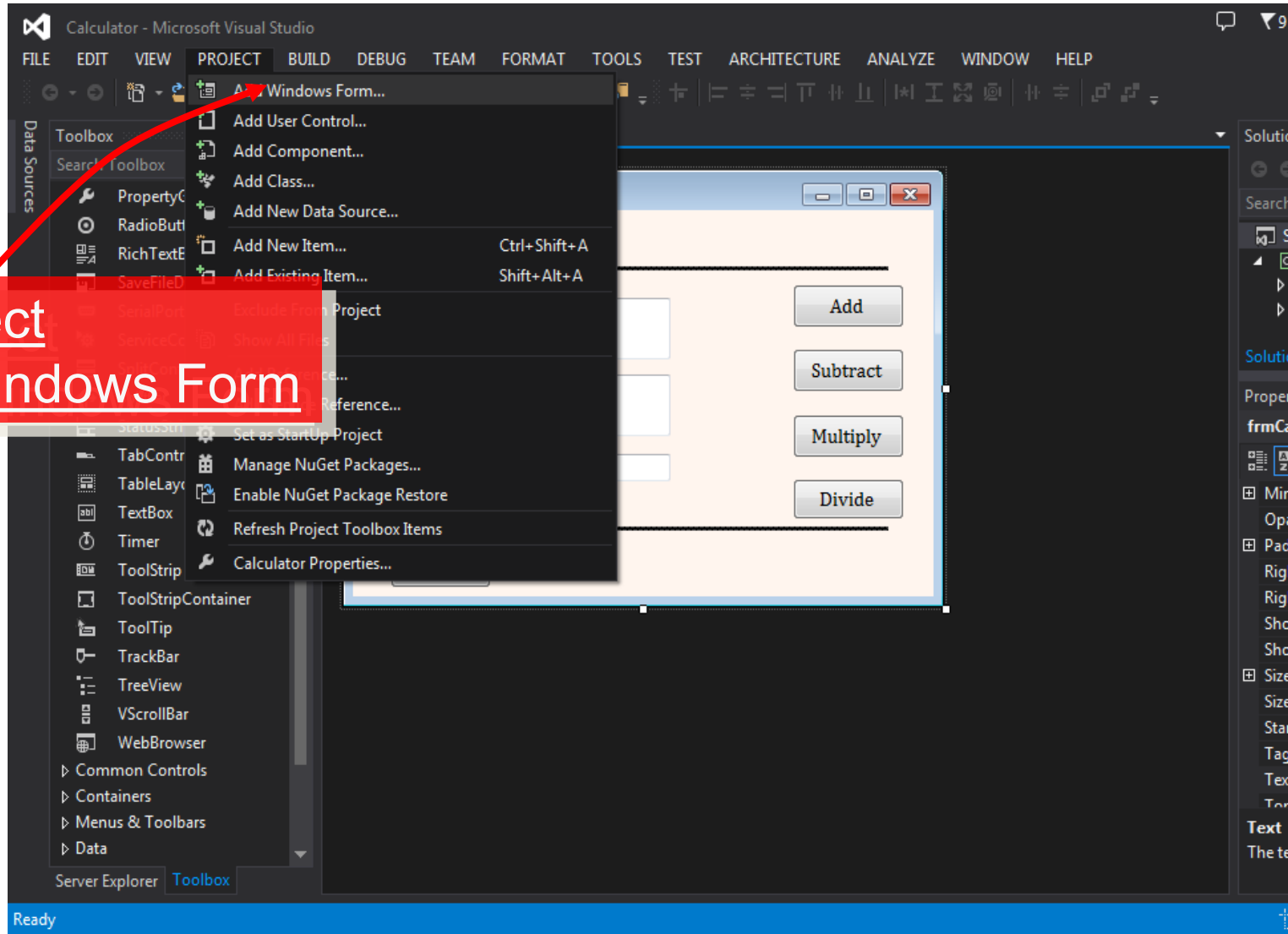
# CO453 Application Programming

Week 8 – multiform projects

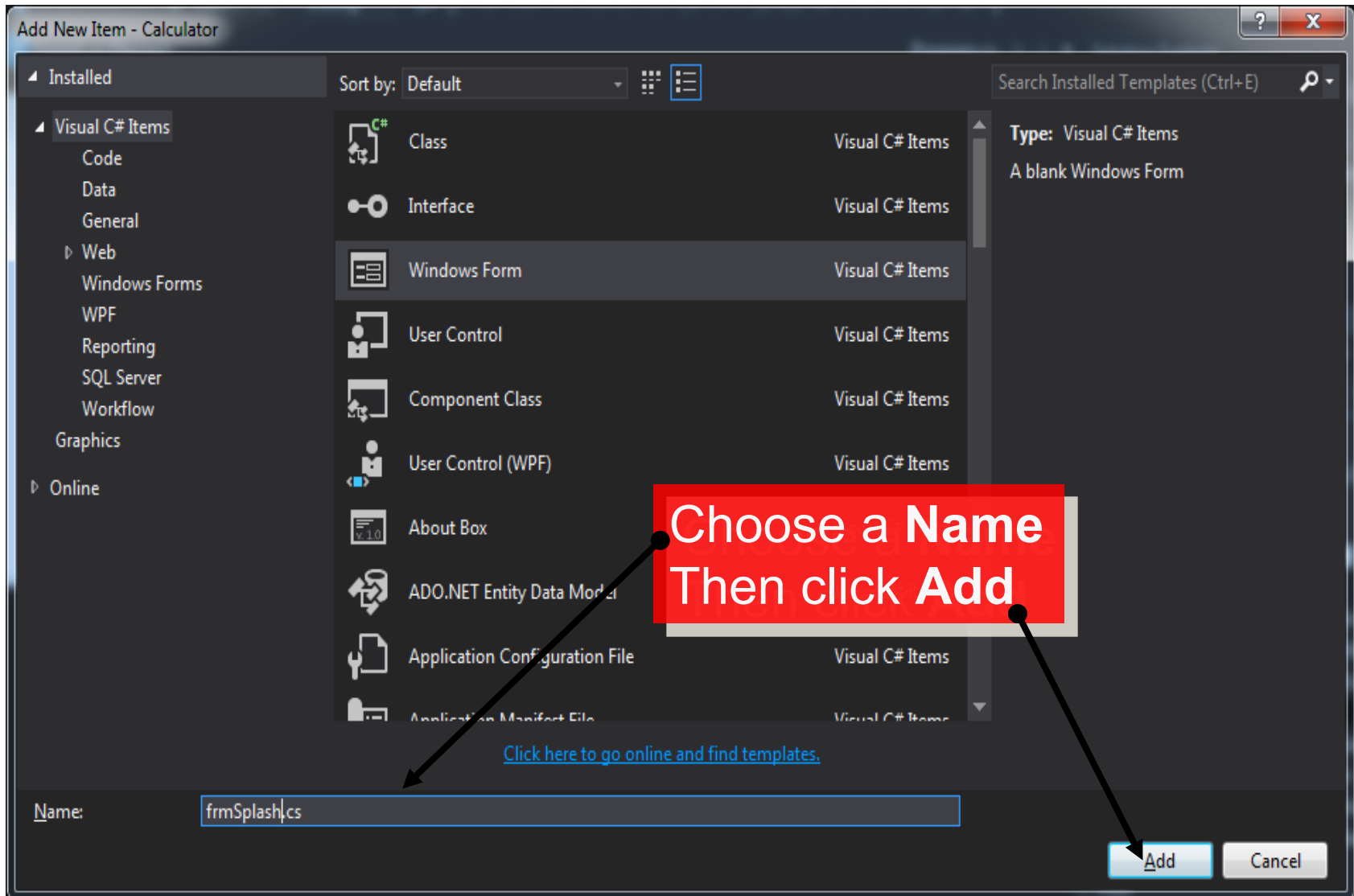
.NET part 4

# **Adding a Form to a project (reminder)**

# 1: Choose Add Windows Form

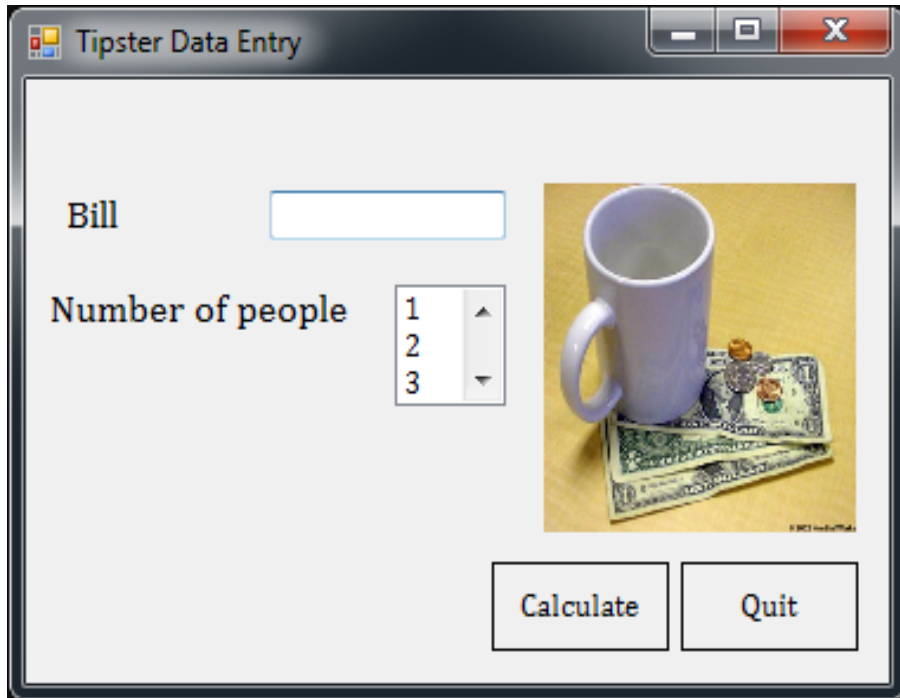


## 2: Name the new Form

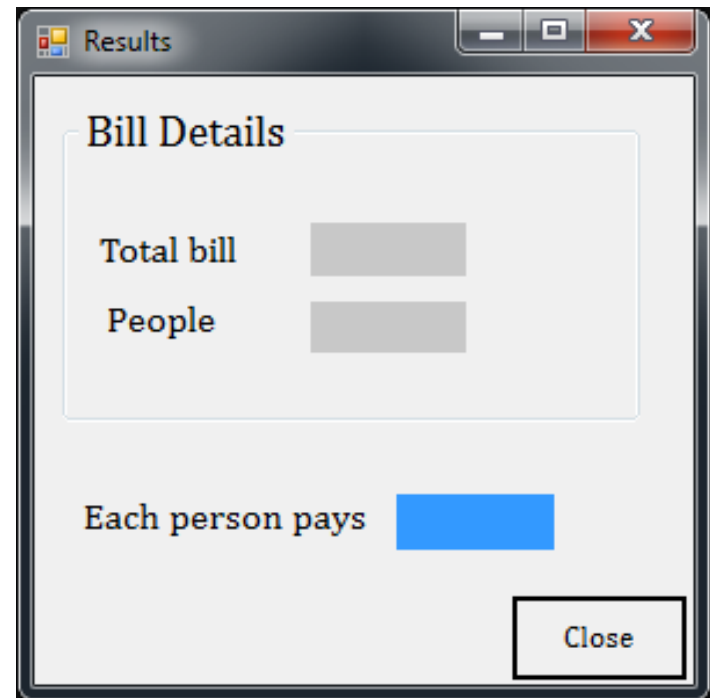


# Activity

Work on the forms in 'Tipster' (Task 4.1, 4.2) to resemble the below



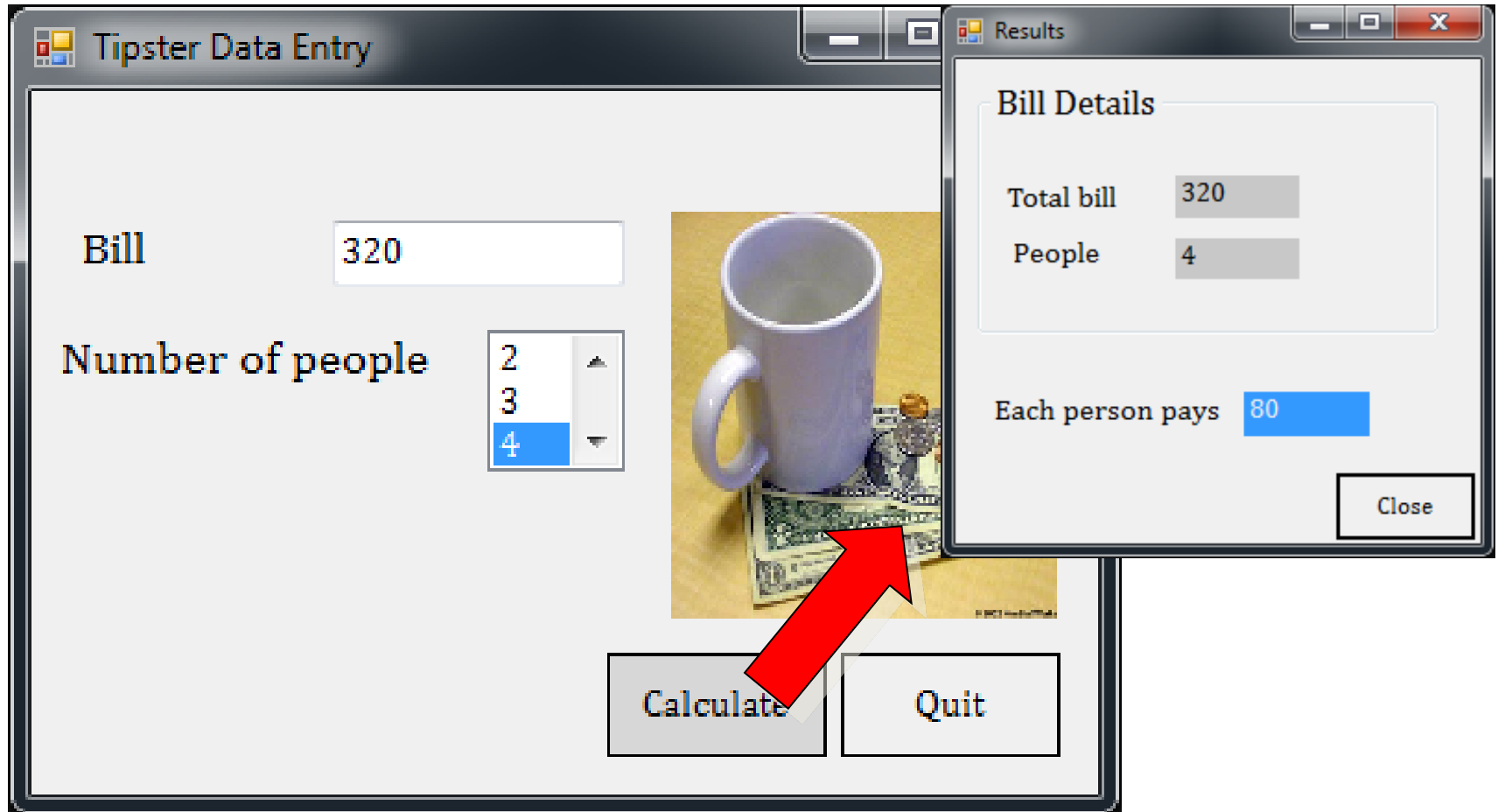
The screenshot shows a window titled "Tipster Data Entry". It features a text input field labeled "Bill", a dropdown menu labeled "Number of people" with options 1, 2, and 3, and a small image of a white mug with a dollar bill and coins. At the bottom, there are two buttons: "Calculate" and "Quit".



The screenshot shows a window titled "Results". It features a section titled "Bill Details" containing two labels, "Total bill" and "People", each followed by a grey rectangular placeholder. Below this section, there is a label "Each person pays" followed by a blue rectangular placeholder. At the bottom right, there is a "Close" button.

# **Passing Data from one Form to another**

# What we want to do



## When Calculate is Clicked

- newForm is created and displayed
- Text is passed from txtBill to lblTotal (on newForm)
- Text is passed from lstPeople to lblPeople (on newForm)
- A calculation is done and displayed

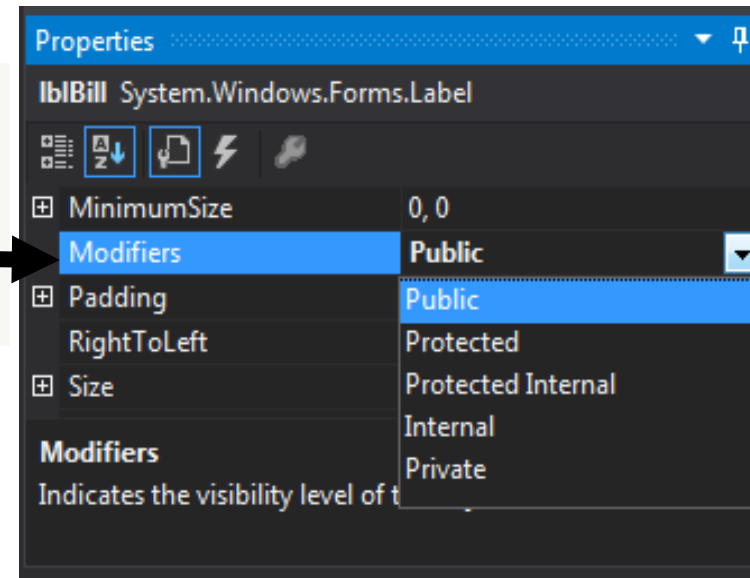
# Problem!

When you add objects to a form, they are automatically set to Private

So their data is NOT available to other forms!

## Solution

- Set the Modifiers property to Public for all these objects



Information hiding is also known as 'encapsulation'



# Programming the Calculate Button *(double-click it first)*

# 1. Copy stuff to a new Form

```
private void btnCalculate_Click (object sender, EventArgs e)
{
    // create a new form object from frmResults
    frmResults newForm = new frmResults();

    // copy text to lblTotal (on newForm) from txtBill (on this form)
    newForm.lblTotal.Text = this.txtBill.Text;

    // show the new form object
    newForm.Show();
}
```

this is optional

**Note:** assume that our added form was named frmResults

## 2. Doing the Calculation

```
private void btnCalculate_Click (object sender, EventArgs e)
{
    // Do the stuff from the previous slide

    // Define 3 variables for the calculation
    double n1, n2, answer;

    // Convert text from a text box and a list box into numbers
    n1 = Convert.ToDouble(txtBill.Text);
    n2 = Convert.ToDouble(lstPeople.Text);

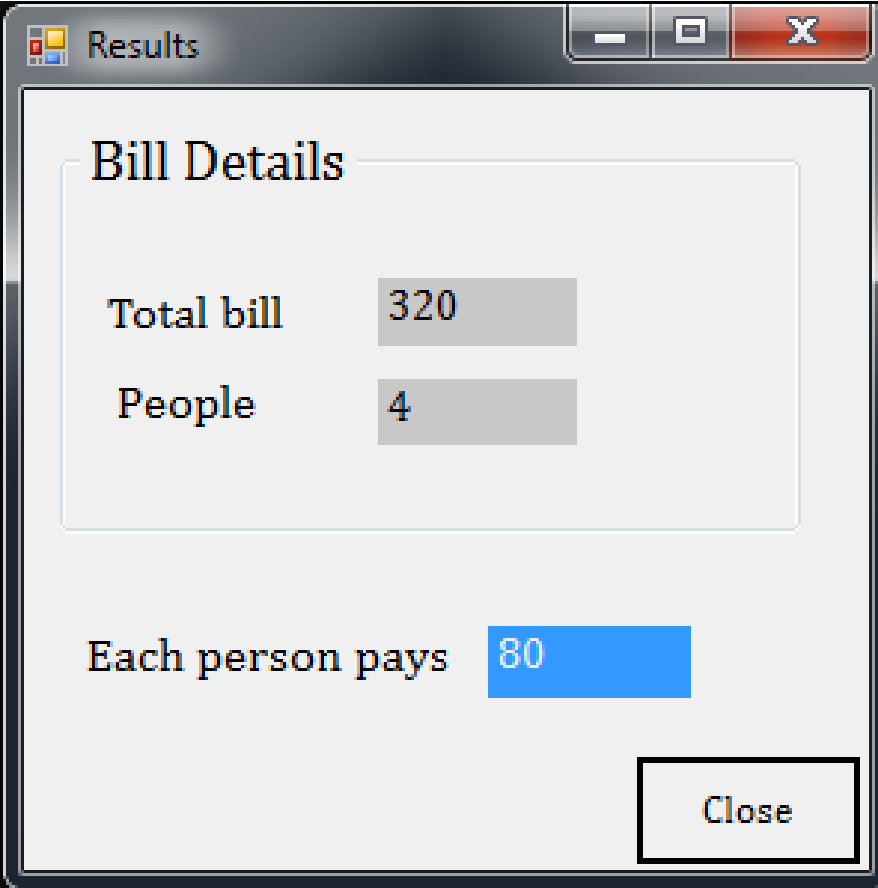
    // work out answer and display it in lblResult
    answer = n1/n2;
    newForm.lblResult.Text = answer.ToString();

    // show the form with all the results
    newForm.Show();
}
```

**Note:** assumed names are txtBill, lstPeople and lblResult

# Activity

Now try Task 4.3 and pass data from the Data Entry form to the Results form



The screenshot shows a window titled "Results" with a standard Windows-style title bar (minimize, maximize, close buttons). The window content is as follows:

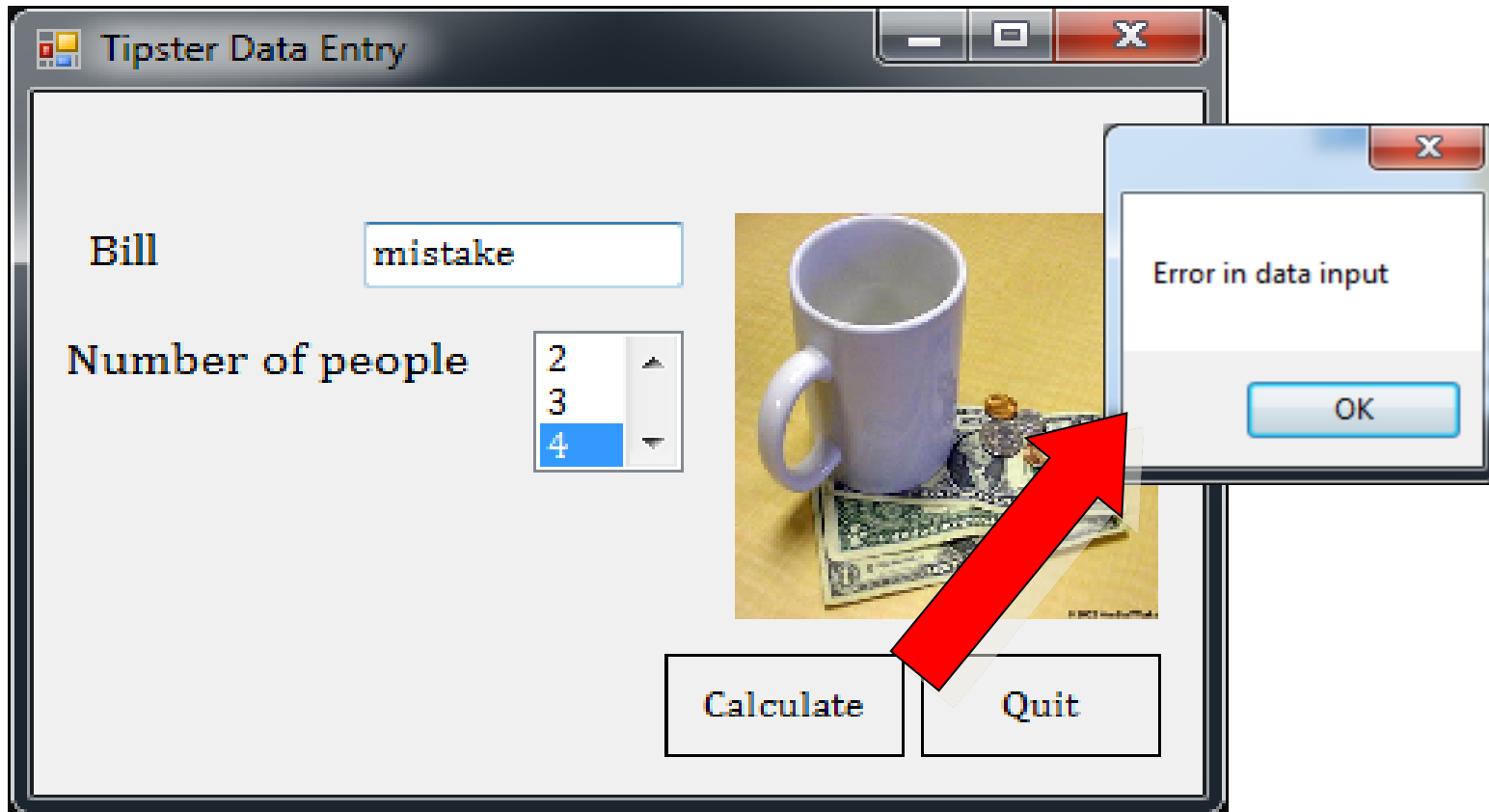
Bill Details	
Total bill	320
People	4

Each person pays 80

Close

**Catching Errors  
(or Exceptions)  
using try .. catch ..**

# What we want to do



- The program will crash if we don't enter a number for the Bill
- We prefer the program to recover and continue, like this

# How? Use try .. catch ..

```
try
```

```
{
```

```
    // put instructions here that you want to check  
n1 = Convert.ToDouble(txtBill.Text);  
    // put the rest of the calculations here
```

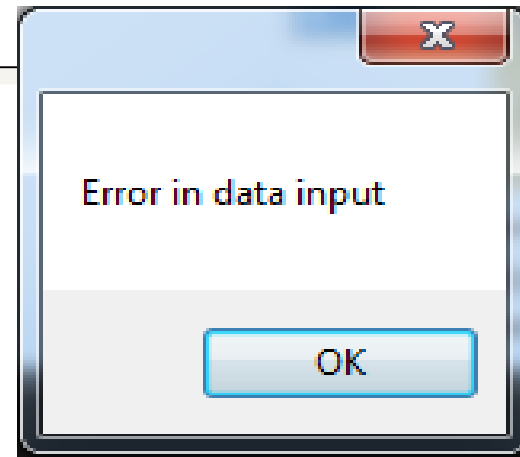
```
}
```

```
catch
```

```
{
```

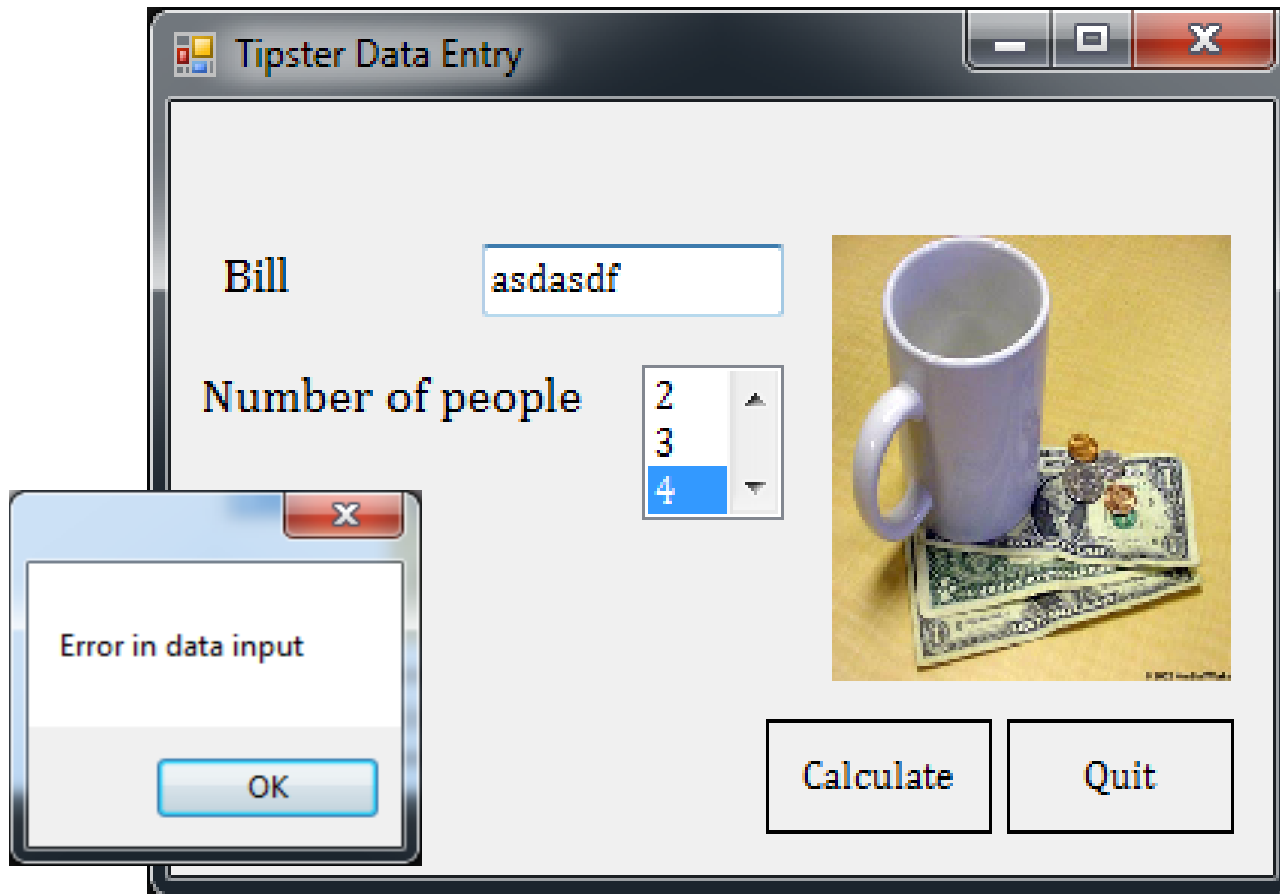
```
    // display an error message here  
MessageBox.Show("Error in Data Entry" );
```

```
}
```



# Activity

Implement error handling (Task 4.4)





# The Last Slide



# Extra Reading

# Some other versions of catch

```
catch (FormatException ex)
{
    MessageBox.Show ("Error in input : " + ex.Message);
}
```

```
catch (ArithmeticException ex)
{
    MessageBox.Show ("Calculation Error : " + ex.Message);
}
```

```
catch (Exception ex)
{
    MessageBox.Show ("Some other error : " + ex.Message);
}
```

- Multiple catch blocks can be used with a single try block
- The ex.Message is not always helpful! Try some.