Topic 2B

Storing Data
And
Performing Math Operations
Part I

Topics

- ❑ Storing Data
- ❑ Data Types
- ❑ Arithmetic Operators

Objectives:

- ❑ Be able to understand what are variables used for
- ❑ Be able to understand simple data types
- ❑ Be able to perform simple arithmetic operations
- ❑ Introduce Debugger

Storing data

- ❑ We store data in the computer using variables. What exactly are variables?

Variables are **containers** assigned by you, the programmer, to store data in your program.

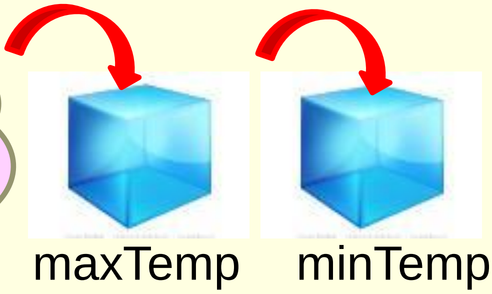
- ❑ Let’s look at this example:

Examples of variables (Pseudo code):

```
Read maxTemp, minTemp
```

The pseudo code above will read in 2 values from the keyboard, and store them into the variables maxTemp and minTemp.

Variables are containers for storing data



Storing data

- When we give variable a name, we should make it meaningful. Example:

Meaningful variable names	Not meaningful variable names	Remarks
age	a	Age of a person
marks	mk	Marks of a student

- In addition, C# variable names must conform to the following rules:
  1. Must begin with a letter or an underscore
  2. Can be followed by letters, underscore or digits.
  3. Case sensitive
  4. Cannot be a keyword
  5. Cannot have special characters (such as control chars, space)

IT1753 Principles of Computing PCMP

Storing data

- Examples of valid and invalid variable names

Rule	Valid Variable	Invalid Variable	Remarks
1	_name	@name	Begin with @
2	name3	name 3	Has space between name and 3
3	Name name	-	Name and name are treated as separate variables
4	IF	if	if is a keyword (see next slide)
5	name3	name 3	Space between name and 3

- Rules for C# variable names
  1. Must begin with a letter or an underscore
  2. Can be followed by letters, underscore or digits.
  3. Case sensitive
  4. Cannot be a keyword
  5. Cannot have special characters

IT1753 Principles of Computing PCMP

List of C# Keywords

abstract	as	base	bool	break
byte	case	catch	char	checked
class	const	continue	decimal	default
delegate	do	double	else	enum
event	explicit	extern	false	finally
fixed	float	for	foreach	goto
if	implicit	in	int	interface
internal	is	lock	long	namespace
new	null	object	operator	out
override	params	private	protected	public
readonly	ref	return	sbyte	sealed
short	sizeof	stackalloc	static	string
struct	switch	this	throw	true
try	typeof	uint	ulong	unchecked
unsafe	ushort	using	virtual	void
while				

Cannot use these for variable names as they’re reserved by Visual Studio

IT1753 Principles of Computing PCMP

Data Types

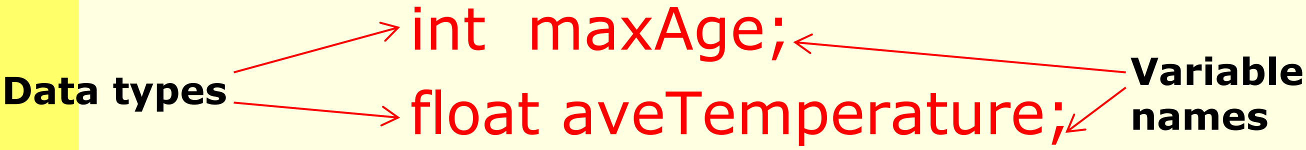
- ❑ The variable name given by you is designed to store specific type of data.

- ❑ Let's look at 2 data types for now:

Data Type	**Range**	**Remarks**
int	2,147,483,648 to 2,147,483,647	Represents whole numbers
float	between $1.5 \ast 10^{-45}$ and $3.4 \ast 10^{38}$	Represents numbers with decimal points

Data Types

- Before using variables, we must create them.
- We do this by preceding the variable name with its data type.
- Example :



```
int  maxAge;
float aveTemperature;
```

- These statements tell the computer to reserve storage areas for these variables.

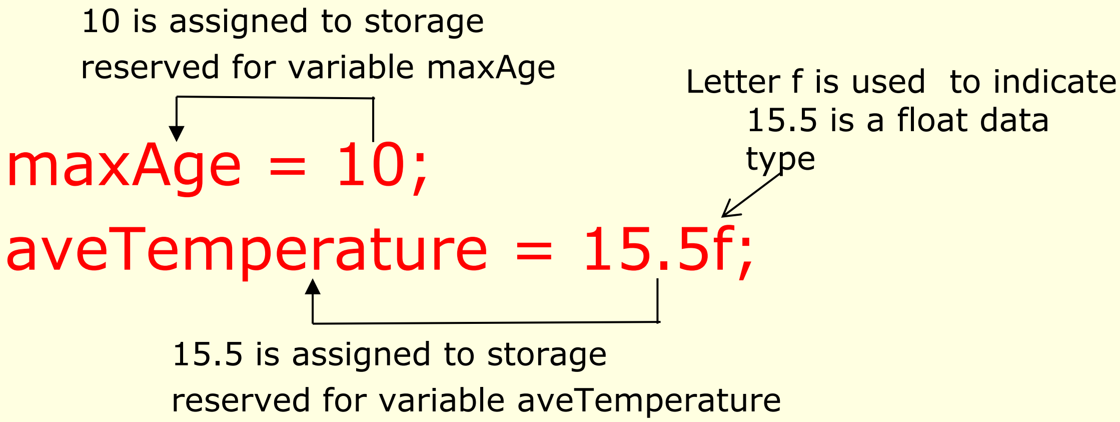
Address	Storage Area	Variable names
0001		maxAge
0002		aveTemperature

IT1753 Principles of Computing PCMP

Data Types

- If we want to store data into the variable name, we do it as follows:

```
int  maxAge;
float aveTemperature;
```



10 is assigned to storage reserved for variable maxAge

```
maxAge = 10;
aveTemperature = 15.5f;
```

Letter f is used to indicate 15.5 is a float data type

15.5 is assigned to storage reserved for variable aveTemperature

Address	Storage Area	Variable names
0001	**10**	maxAge
0002	**15.5**	aveTemperature

❗ We can create a variable and assign it a value together. Example:

```
int  maxAge = 10;
```

Default Value

- When you create a variable and set a value at the same time, it is called the **default** value

Example:

int Age = 17;

Note: the default value is 17

Example:

int Age;

What is the default value?

Arithmetic Operators

- ❑ The computer can perform arithmetic calculations much faster than you, so let’s just learn to do that now.
- ❑ The basic operators are:

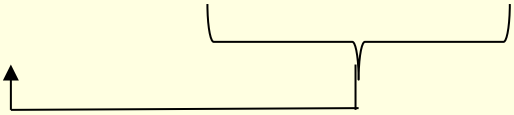
Operator	Name
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (used to obtain remainder from division)

IT1753 Principles of Computing PCMP

Arithmetic Operators

❑ Example:

```
int  totalAge, age1, age2;
age1 = 12;
age2 = 8;
totalAge = age1 + age2; //  20
```



a) **Data stored in age1 and age2 are added**

b) **Result of addition is then stored in totalAge**

Arithmetic Operators

- ❑ Additional Examples

Assume both variables a and b are integers

Operator	Expression	Result of a if b=8
Addition +	a= b+5	13
Subtraction -	a=b-5	3
Multiplication *	a=b*5	40
Division /	a=b/5	1
Modulus %	a=b%5	3

Arithmetic Operators

❑ Additional Examples

Assume both variables a and b are float

Operator	**Expression**	**Result of a if b=8.0f**
Addition +	a= b+5.0f	13
Subtraction -	a=b-5.0f	3
Multiplication *	a=b*5.0f	40
Division /	a=b/5.0f	1.6
Modulus %	a=b%5.0f	C# now can take all types of numeric value for modulus. Also can return non-integer result

A Quick Review

- Let’s summarize what we have learnt with a complete example:

```
//create 3 variables of integer data type
int  totalMarks, intEnglish, intMath;

intEnglish = 75;  //store 75 into variable intEnglish
intMath = 82;    // store 82 into variable  intMath

// 75 and 82 are added first
// their result is stored in variable totalMarks
totalMarks = intEnglish + intMath;
```

- We will use this example to create a Windows Form application next.

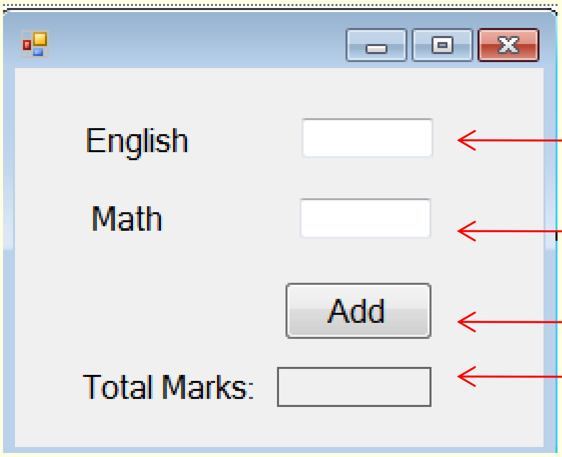
Example 1:
A Simple Add Calculator

- **Problem Statement**: Create a Form to accept marks for 2 subjects, English and Math. Add and display the total marks when the Add button is clicked.

- **Use Case Definition**
  1. User enters English Mark
  2. User enters Math Mark
  3. User clicks on Add button
  4. Total mark of English and Math marks will be displayed
  5. User repeats step 1-4
  6. User terminates the program by clicking the Exit button

Example 1:
A Simple Add Calculator

- **GUI Form Design** : The Form and the name of its controls are shown here:



Textbox : txtEnglish

Textbox : txtMath

Button : btnAdd

Label : lblTotal

Pseudo code

```
Read input English mark.
Read input Math mark.
Calculate Total mark = English mark + Math mark.
Display Total mark.
```

How many variables do you need?

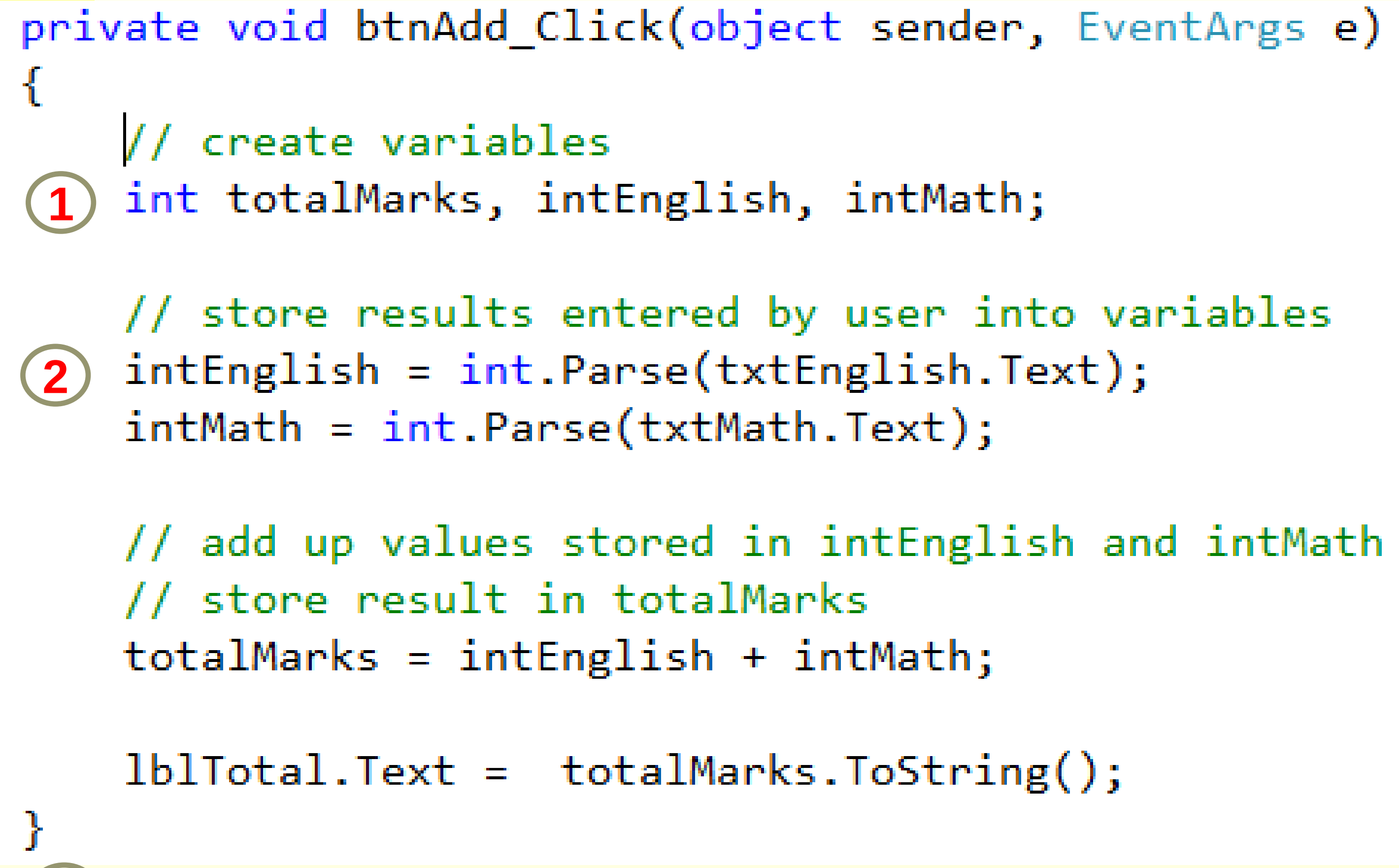
Example 1:
A Simple Add Calculator

- ❑ Open a new window application project in VS.
- ❑ Create the form and 7 controls for the Calculator
- ❑ They are labels, textbox and button
- ❑ Set the properties for all names
- ❑ Code in the "ADD" button click method (see the source in the next slide)

Very Important !!!
You must **DOUBLE CLICK** the **Add** button to jump to the programming screen.
This is the ONLY way to do it.

Example 1:
A Simple Add Calculator

- ❑ Key in the source code



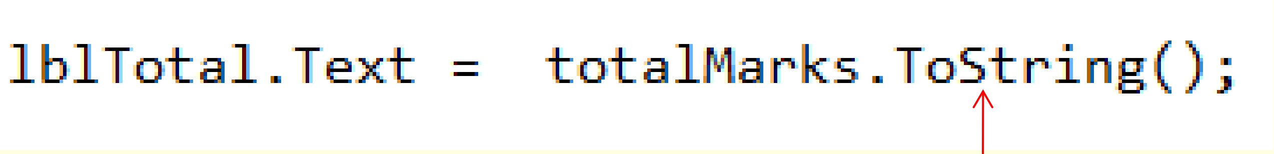
```
private void btnAdd_Click(object sender, EventArgs e)
{
    // create variables
    int totalMarks, intEnglish, intMath;

    // store results entered by user into variables
    intEnglish = int.Parse(txtEnglish.Text);
    intMath = int.Parse(txtMath.Text);

    // add up values stored in intEnglish and intMath
    // store result in totalMarks
    totalMarks = intEnglish + intMath;

    lblTotal.Text =  totalMarks.ToString();
}
```

1 Create 3 variables

2 The marks entered are stored in the Text property of the respective Text boxes. It is then converted to same data type of the variables using int.Parse. Converted value is stored in respective variables

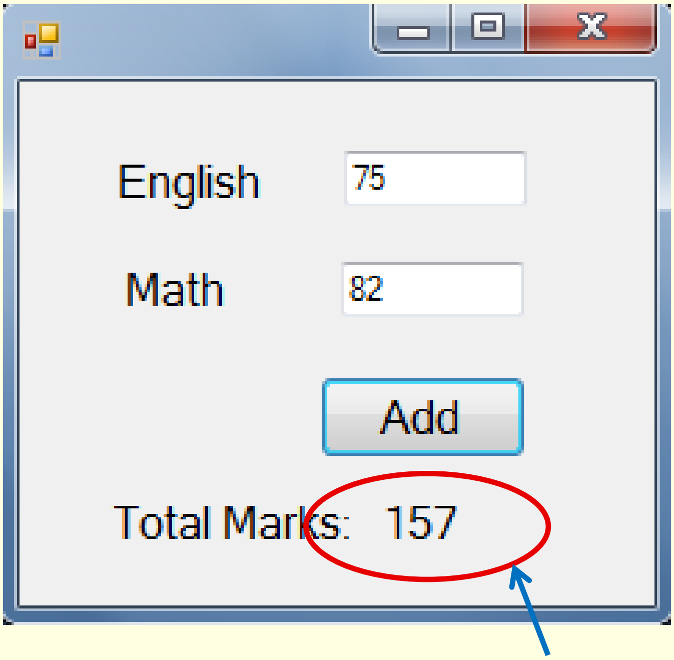
Example 1:
A Simple Add Calculator

- ❑ The last line of code needs further explanation:



```
lblTotal.Text =  totalMarks.ToString();
```

Integer totalMarks must be converted to a Text form (using the ToString() method) so that it can be assigned to the Text property of Label lblTotal.



English 75

Math 82

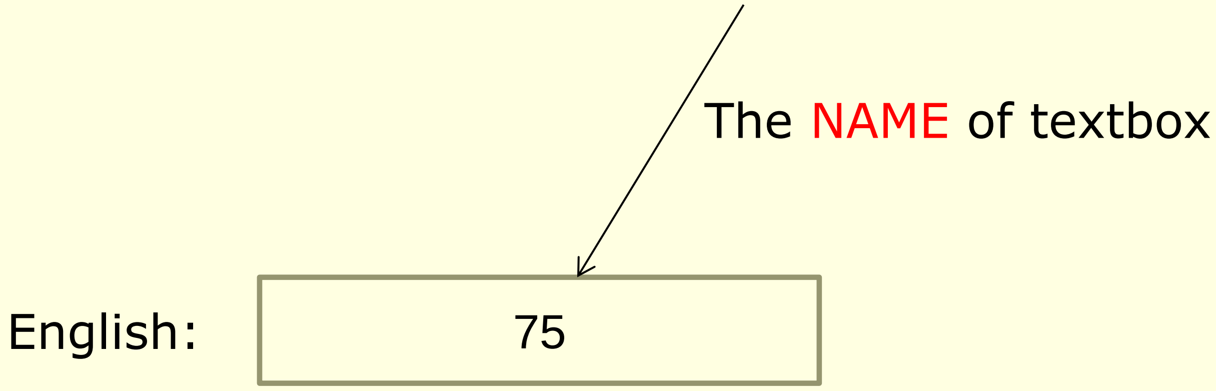
Total Marks: 157

157 is the value stored in totalMarks variable

IT1753 Principles of Computing PCMP

Quick Note:
User Input in C# code

- C# code to read an input from user
  - intEnglish = int.Parse (txtEnglish.Text)



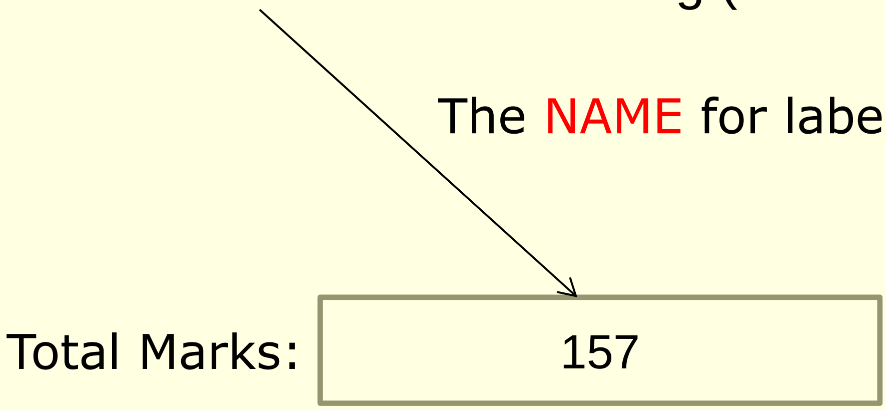
The NAME of textbox

English: 75

- *int.Parse*
  - It takes “75” and converts it to integer 75
- *intEnglish →*
  - intEnglish is assigned to 75
  - intEnglish is a variable, it stores 75

Quick Note: Program output in C# code

- C# code to display output to a label
  - lblTotal.Text = ToString (totalMarks)



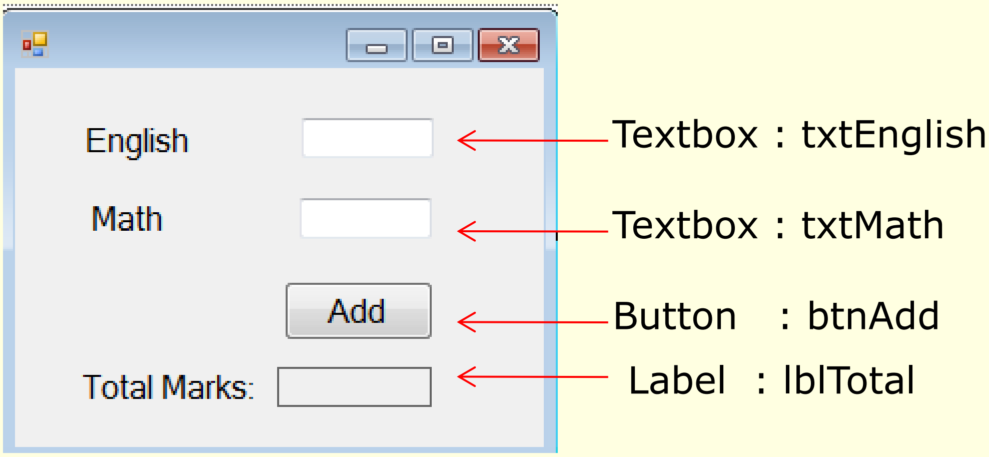
The NAME for label

Total Marks: 157

- *ToString*
  - It takes totalMarks content which is 157 and converts it to a string “157”
- lblTotal.Text →
  - It is assigned to “157”
  - It is then displayed on the label.

IT1753 Principles of Computing PCMP

Review



Textbox : txtEnglish

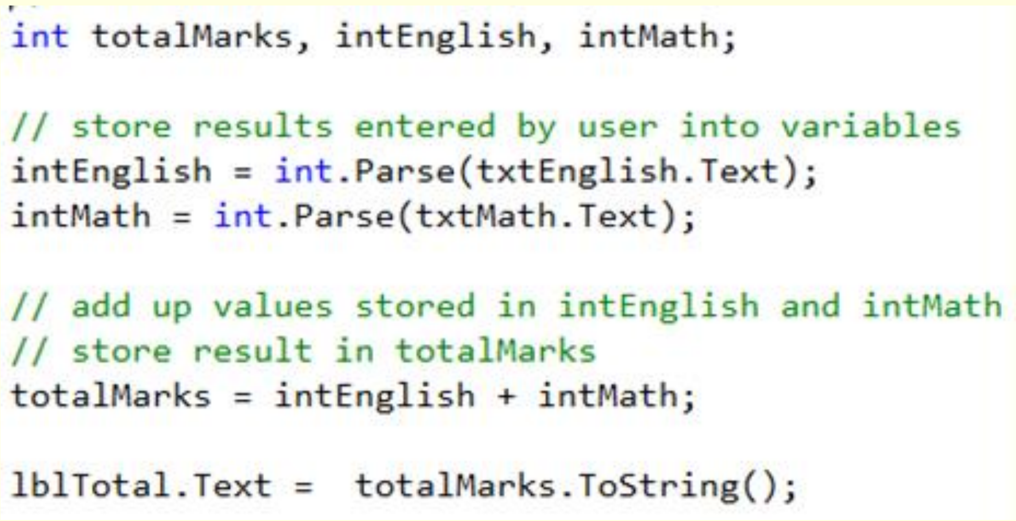
Textbox : txtMath

Button : btnAdd

Label : lblTotal

Pseudo code

```
Read input English mark.
Read input Math mark.
Calculate Total mark = English mark + Math mark.
Display Total mark.
```



```
int totalMarks, intEnglish, intMath;

// store results entered by user into variables
intEnglish = int.Parse(txtEnglish.Text);
intMath = int.Parse(txtMath.Text);

// add up values stored in intEnglish and intMath
// store result in totalMarks
totalMarks = intEnglish + intMath;

lblTotal.Text =  totalMarks.ToString();
```

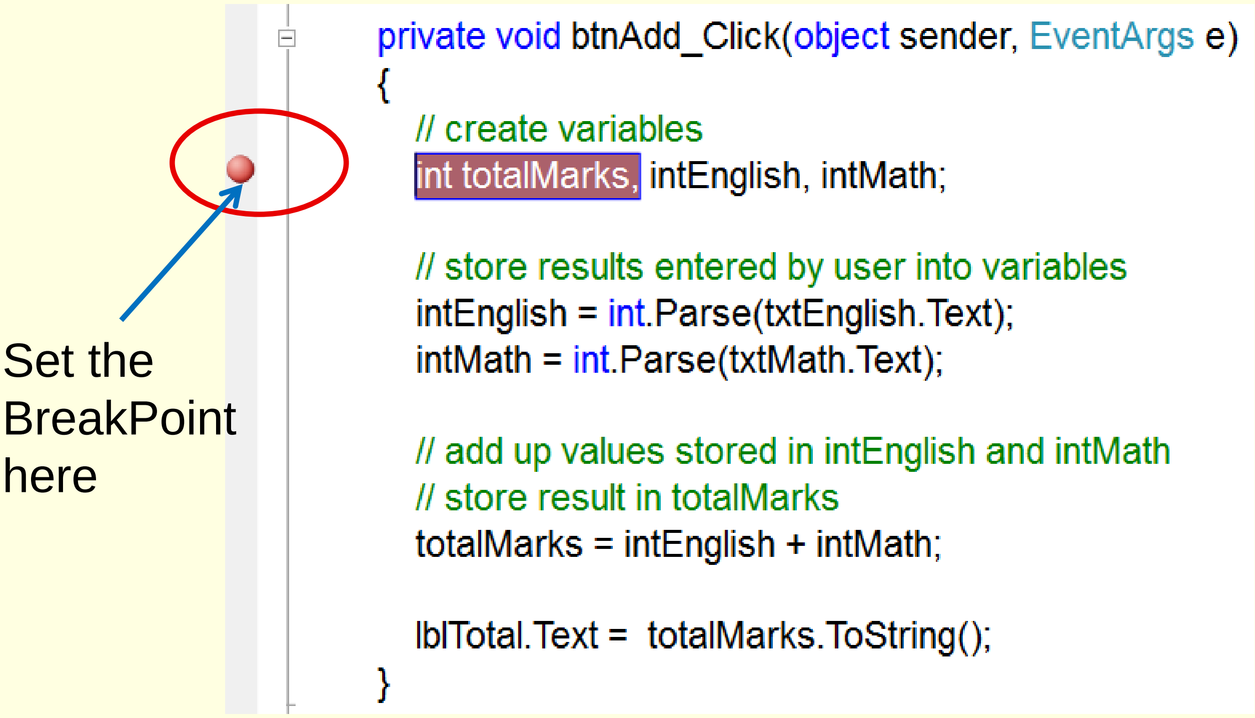
Example 1:
A Simple Add Calculator

- **Debugging Your Program**
  - With good Debugging skill, you can learn any programming languages easily
  - Debugger is a powerful tool that allows you to observe the run-time behaviour of your program
  - By tracing the program, it allows you to find the error (Logic) and understand the program better
  - Before you can debug, you program MUST not have any syntax compilation error
  - If you have any compilation error. You Must solve it first before debugging

Example 1: Debug your program – A Simple Add Calculator

- Break point allows the program to stop and programmer to check on the program status

- **Step 1: Set Breakpoint**
  - Open the source, click on the left margin to set a break point

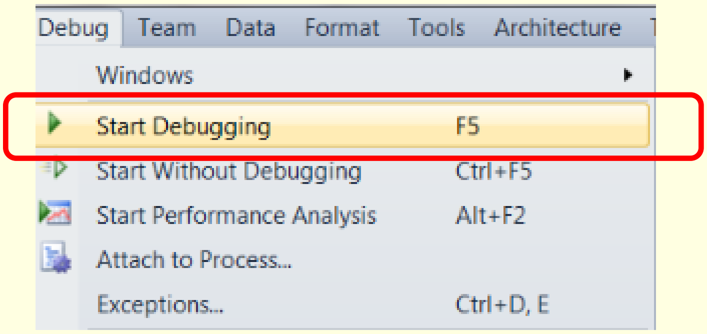


```
private void btnAdd_Click(object sender, EventArgs e)
{
    // create variables
    int totalMarks, intEnglish, intMath;

    // store results entered by user into variables
    intEnglish = int.Parse(txtEnglish.Text);
    intMath = int.Parse(txtMath.Text);

    // add up values stored in intEnglish and intMath
    // store result in totalMarks
    totalMarks = intEnglish + intMath;

    lblTotal.Text =  totalMarks.ToString();
}
```

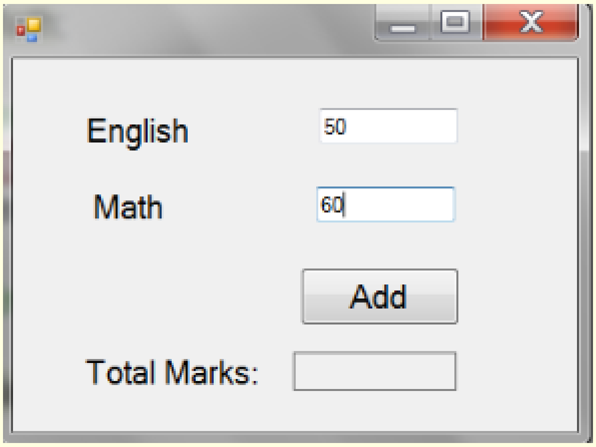
Set the BreakPoint here

Example 1:
Debug your program –
A Simple Add Calculator

- **Step 2: Launch the debugger** – F5 or



- Step 3: Program runs but stop at the form entry. Enter the following data and click Add.



English 50

Math 60

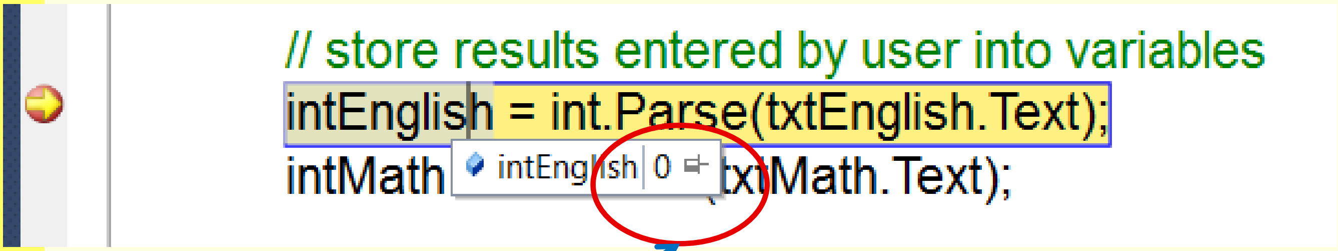
Total Marks:

- Step 4: Code and breakpoint will be displayed

IT1753 Principles of Computing PCMP

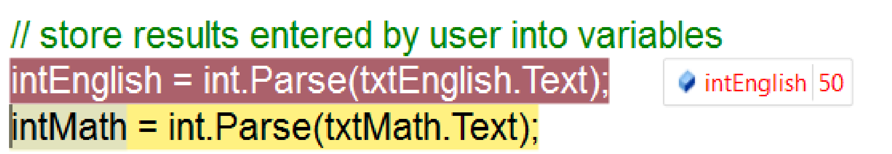
Example 1:
Debug your program –
A Simple Add Calculator

- **Step 5: Pin Watch window**
  - Place your cursor over intEnglish variable, a window will pop up.



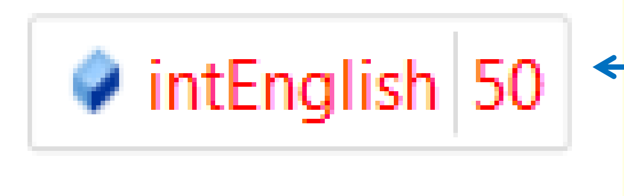
```
// store results entered by user into variables
intEnglish = int.Parse(txtEnglish.Text);
intMath = int.Parse(txtMath.Text);
```

  - Click on the Pin icon to pin the watch window on the screen
  - **F11** to trace the program



```
// store results entered by user into variables
intEnglish = int.Parse(txtEnglish.Text);
intMath = int.Parse(txtMath.Text);
```

  - The intEnglish watch window shows 50. It allows you to monitor the variable’s content

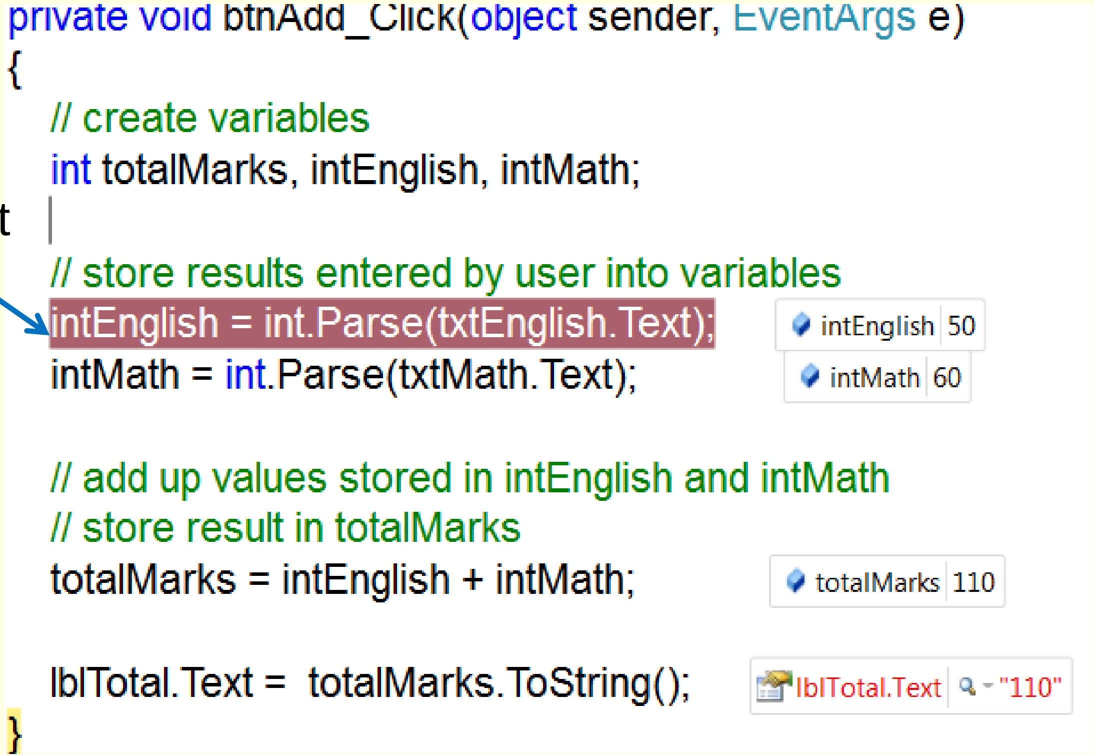


intEnglish stores 50

IT1753 Principles of Computing PCMP

Example 1: Debug your program – A Simple Add Calculator

- **Step 6: Pin intMath, TotalMarks and lblTotal.text**
  - Place your cursor over intMath and pin the watch window
  - Place your cursor over TotalMarks and pin the watch window
  - Place your cursor over lblTotal.Text and pin the watch window
  - Trace the program line by line by **F11**
  - Observe how the values are stored in the intMath, TotalMarks and lblTotal.Text

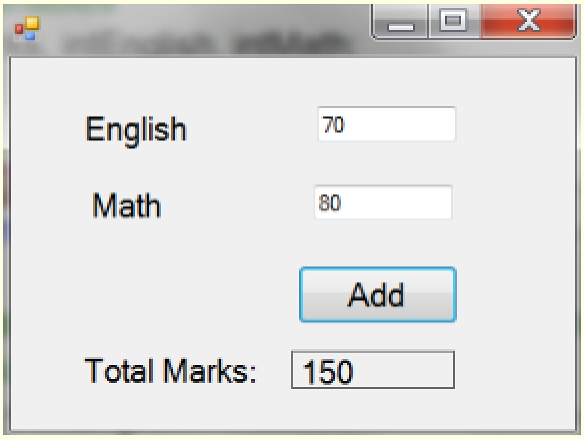


Breakpoint

```
private void btnAdd_Click(object sender, EventArgs e)
{
    // create variables
    int totalMarks, intEnglish, intMath;

    // store results entered by user into variables
    intEnglish = int.Parse(txtEnglish.Text);      intEnglish 50
    intMath = int.Parse(txtMath.Text);            intMath 60

    // add up values stored in intEnglish and intMath
    // store result in totalMarks
    totalMarks = intEnglish + intMath;            totalMarks 110

    lblTotal.Text =  totalMarks.ToString();       lblTotal.Text "110"
}
```

Example 1: Debug your program – A Simple Add Calculator

- **Step 7: Run it again and Observe it**
  - When you have completed tracing the program, it loops back to the form entry
  - Enter another set of values

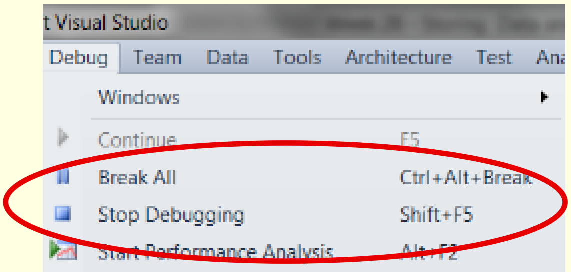


English 70

Math 80

Total Marks: 150

  - Observe the watch windows
- To stop the debugger,



- Do you find the watch windows useful? Does it help you to understand the program better?

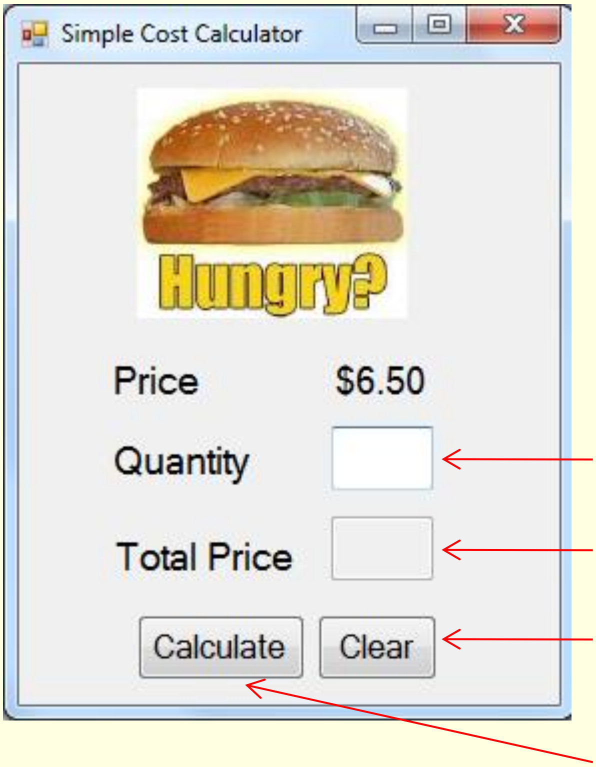
Example 2: A Simple Cost Calculator

- **Problem Statement**:
  - Create a Form to accept the quantity of Burgers to be purchased.
  - Compute the total price when the Calculate button is clicked, given one Burger cost $6.50.
  - The Textbox to enter the quantity is cleared when the Clear button is clicked.

- **Use Case Definition:**
  1. User enters Quantity
  2. User clicks on Calculate Button
  3. Total price is displayed
  4. User clicks on Clear Button to clear the entries
  5. User repeats step 1-4
  6. User terminates the program by clicking the Exit button

Example 2:
A Simple Cost Calculator

❑GUI Form Design :



Textbox : txtQuantity

Label : lblTotalPrice

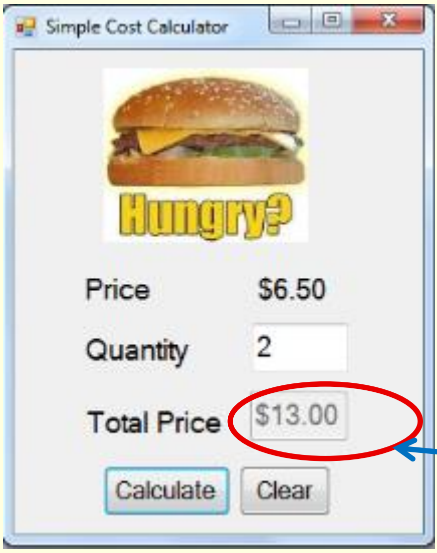
Button : btnClear

Button : btnCalculate

Write the pseudo code first if you are not sure how to start coding.

Example 2: A Simple Cost Calculator

- The source code within the Click method for btnCalculate and corresponding output:

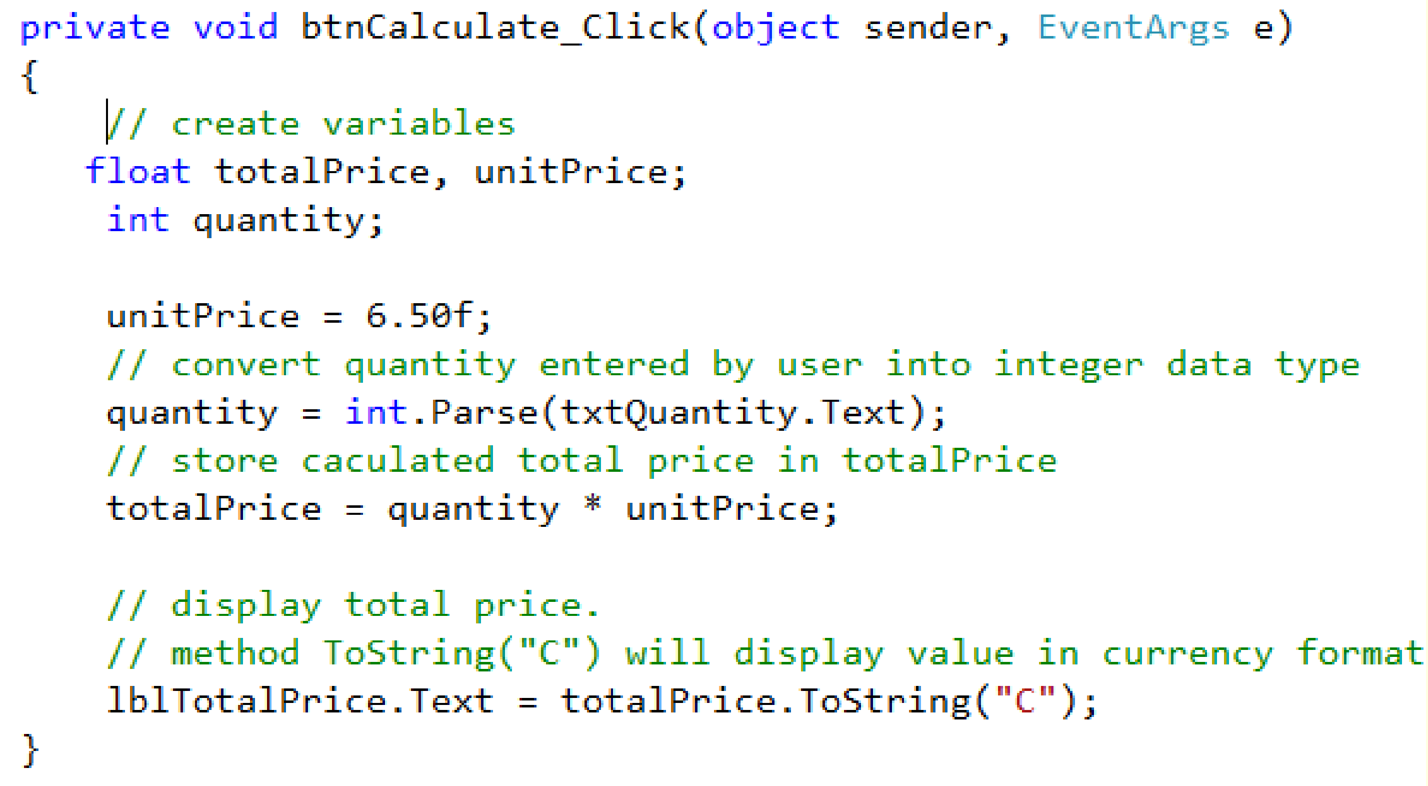


Price $6.50

Quantity 2

Total Price $13.00

13 is the value stored in totalPrice variable

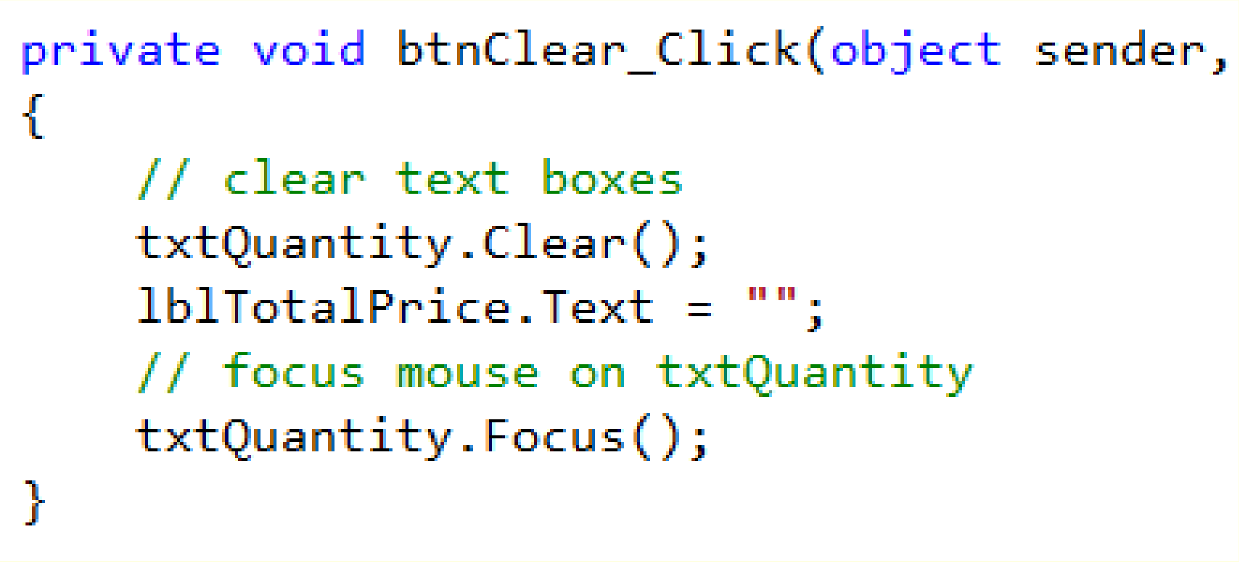


```
private void btnCalculate_Click(object sender, EventArgs e)
{
    // create variables
   float totalPrice, unitPrice;
    int quantity;

    unitPrice = 6.50f;
    // convert quantity entered by user into integer data type
    quantity = int.Parse(txtQuantity.Text);
    // store caculated total price in totalPrice
    totalPrice = quantity * unitPrice;

    // display total price.
    // method ToString("C") will display value in currency format
    lblTotalPrice.Text = totalPrice.ToString("C");
}
```

IT1753 Principles of Computing PCMP

Example 2:
A Simple Cost Calculator

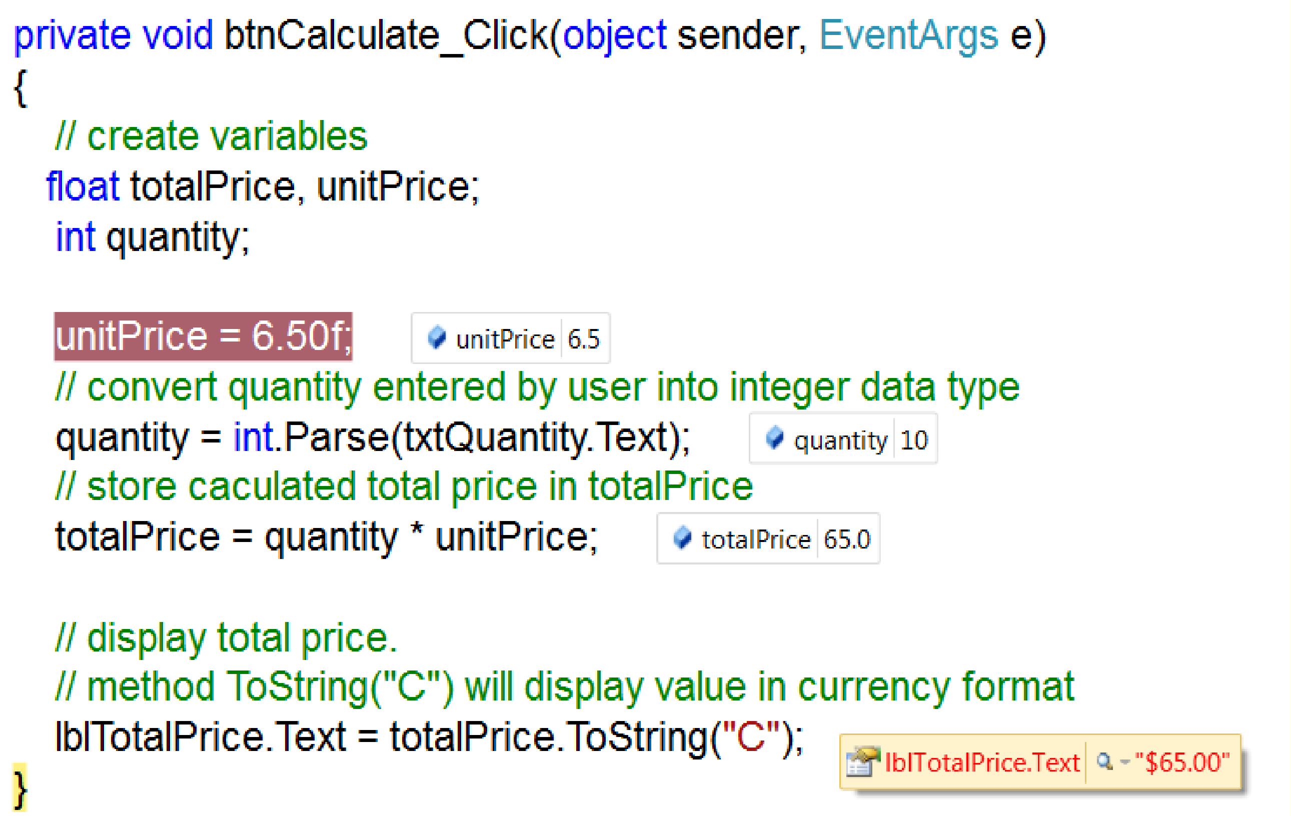
- ❑ When user clicks on Clear button, the text in Quantity will be cleared
- ❑ Mouse cursor stays at Quantity text
- ❑ The source code within the Click method of btnClear



```
private void btnClear_Click(object sender,
{
    // clear text boxes
    txtQuantity.Clear();
    lblTotalPrice.Text = "";
    // focus mouse on txtQuantity
    txtQuantity.Focus();
}
```

Example 2:
A Simple Cost Calculator – Do it!!

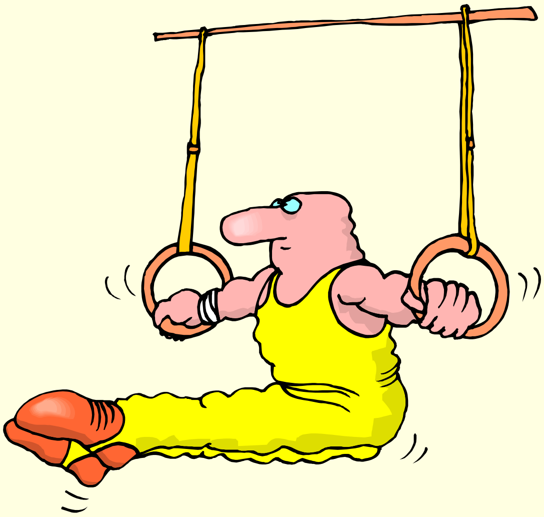
- Tutor will distribute **Topic2BExample2starter**
- Open project in VS.
- Form and controls are created
- Add in the Code for “Calculate” and “Clear” button click methods (see the source)
- Build and run the program
- Set a break point as shown
- Start the debugger
- Add in watch window for all variables



```
private void btnCalculate_Click(object sender, EventArgs e)
{
    // create variables
   float totalPrice, unitPrice;
    int quantity;

    unitPrice = 6.50f;        unitPrice 6.5
    // convert quantity entered by user into integer data type
    quantity = int.Parse(txtQuantity.Text);        quantity 10
    // store caculated total price in totalPrice
    totalPrice = quantity * unitPrice;        totalPrice 65.0

    // display total price.
    // method ToString("C") will display value in currency format
    lblTotalPrice.Text = totalPrice.ToString("C");
}                                         lblTotalPrice.Text "$65.00"
```

Summary

- ❑ Variables are created to store data
- ❑ Variable names have rules when creating them
- ❑ Variables are created to store specific data types
- ❑ The Arithmetic Operators are +, -, *, / and %
- ❑ Debugger helps to trace errors.



Practical 2B

Storing Data And Performing Math Operations Part I

1. Suggest what data type (int or float) and you would use for the following. Explain your choice

- Marks e.g. 88.5
- Age e.g. 17
- Price e.g. 15.60
- Temperature e.g. 23.5

Practical 2B

Storing Data And Performing Math Operations Part I

2. The following are the variable's name. Comment if it is a VALID or NON VALID name and explain.

a) _alpha
b) X_
c) FLOAT
d) maxValue
e) RED ROSE
f) 1_of_many
g) howmany
h) int
i) say_what?
j) Number
k) string
l) 2014_tax
m) 5_stars
n) stop!
o) price per item

Practical 2B

Storing Data And Performing Math Operations Part I

3. Calculate the Area of a Triangle given its base and height

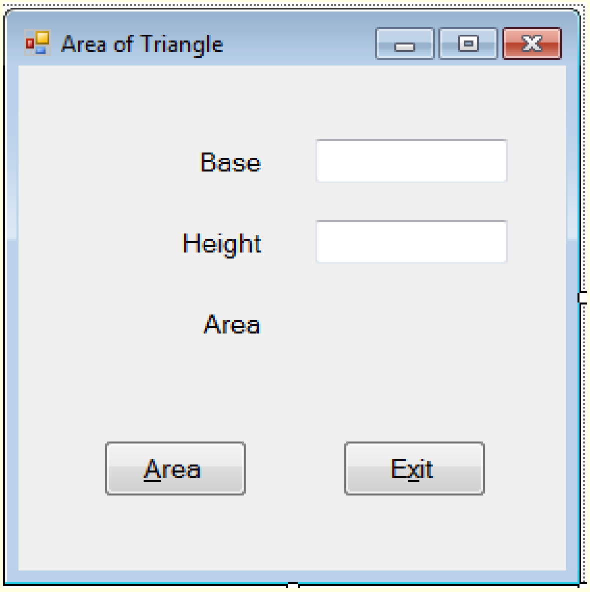
- **Problem Statement**: Create a Form to accept base and height of a Triangle. When the button is clicked, the area of the Triangle is calculated and displayed. Starter files are given to you.

$$\textbf{Area} = \frac{\textbf{base x height}}{\textbf{2}}$$

- **Use Case Definition**
  - 1. User enters Base value
  - 2.
  - 3. <and more>

- Write the **Pseudo code** (ie. algorithm)

- **GUI Form Design** :



Practical 2B

Storing Data And Performing Math Operations Part I

4. Calculate the total price of tickets

- **Problem Statement**: Create a Form to accept the number of tickets for child and adults to be purchased. Calculate the total price to be paid when the button is clicked. Starter files are given to you.

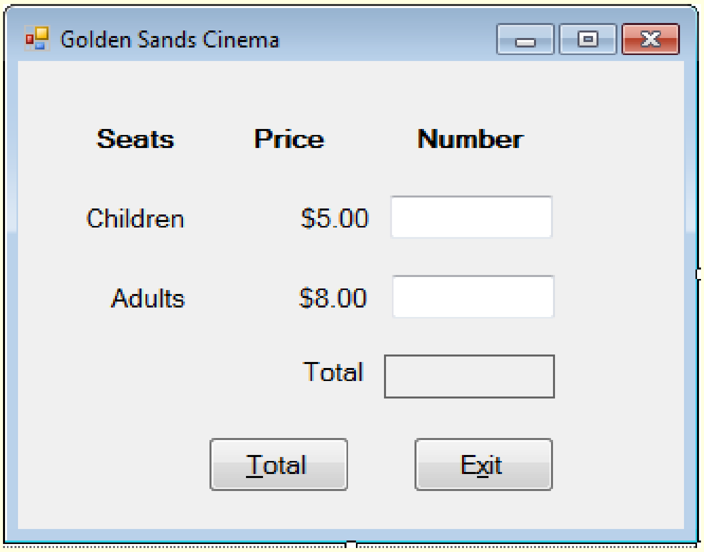
- **Use Case Definitions**
  - 1. User enters Number of Children
  - 2.
  - 3. <and more>

- Write the **Pseudo code**

  (ie. algorithm)

- **GUI Form**

  Design :



Golden Sands Cinema

Seats	**Price**	**Number**
Children	$5.00	
Adults	$8.00	
	Total	

Homework 3%

- ❑ You have completed the GUI in week for all questions in Practical 1B
- ❑ Now add in your C# code to make the application fully "ALIVE"
- ❑ Submit your applications to your Tutor in Class
- ❑ Assignment mark is 3%

E-Assignment – 2%

Storing Data And Performing Math Operations Part I

- If you are confuse and need extra help, you may
  - Download the e-learning material.
    - Powerpoint slide – step by step guide
    - Tutorial guide video
- Tasks for students:
  - Review ALL e-exercises objects
  - Follow the guide and video clip closely
  - Complete the 4 window applications in the exercises

End of Topic 2B

Storing Data
And
Performing Math Operations
Part I