Topic 2C

Storing Data
And
Performing Math Operations
Part II

Topics

- Variable Scope
- More Data Types
- Order of Precedence

Objectives:

- Understand variable scope
- Understand additional data types introduced
- Understand order of precedence used in calculations

Variable scope

- As you know by now, variables are containers assigned to store data in your program.
- Before using variables, we must create them by preceding the variable name with its data type.

(ie. give it a name and define the type of data it can store)

- Example :

```
int totalMarks;
```

Variable scope

- Using Example 1 from last lesson, notice that so far, the variables are created inside a method.

```
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();
}
```

- These are called local variables because they are used locally where they are created within the method.

Variable scope

- Once the method completes its task and exits, the **local variables** within that method is destroyed. The data stored in them will be lost!
- For example, the total marks computed stored in totalMarks will be lost once the method completes its task and exits.

Scope of local variable totalMarks

```
private void btnAdd_Click(obje
{
    // create variables
    int totalMarks, intEnglish

    // store results entered b
    intEnglish = int.Parse(txt
    intMath = int.Parse(txtMat

    // add up values stored in
    // store result in totalMa
    totalMarks = intEnglish +

    lblTotal.Text =  totalMark
}
```

totalMarks Cannot be used after the "**}**"

Variable scope

- If you want to keep the value of totalMarks computed, you should declare it outside the method as shown. This variable is known as a class variable, since it is created within a class. (e.g class frmAdd)

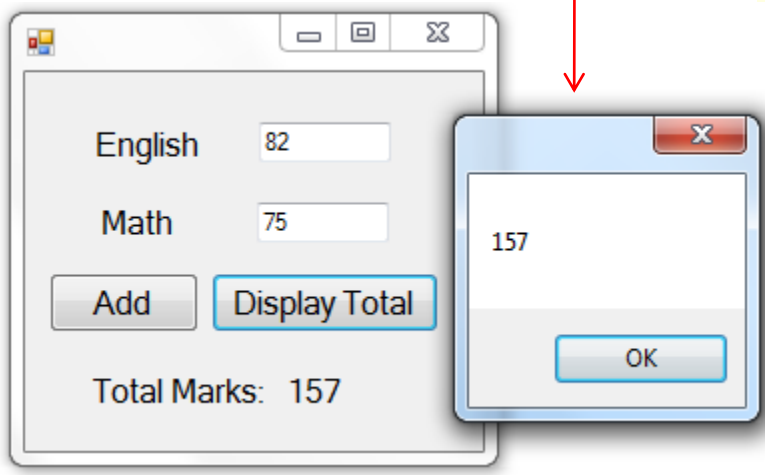
```
namespace Topic2CDemo
{
    public partial class frmAdd : Form
    {
        int totalMarks; // create class variable

        public frmAdd()...

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

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

Scope of **Class** variable totalMarks

Variable scope

- This simple example demonstrates that a class variable retains it value so long as the program is running.

```
namespace Topic2CDemo
{
    public partial class frmAdd : Form
    {
        int totalMarks; // create class variable

        public frmAdd()...

        private void btnAdd_Click(object sender, EventArgs e)...

        private void btnDisplayTotal_Click(object sender, EventArgs e)
        {
            MessageBox.Show(totalMarks.ToString());
        }
    }
}
```



When btnDisplayTotal button is clicked repeatedly, the value of totalMarks is displayed in the Message Box

More Data Types

- The following data types were covered last lesson:

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

- We will introduce 2 additional data types now.

Data Type	Range	Remarks
long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Represents whole numbers
double	between $5.0*10^{-324}$ and $1.7*10^{308}$	Represents numbers with decimal points

More Data Types

- As you can see, long and double data types basically can store numbers of much larger ranges.
- Examples:
  - long population = 3100000L;
  - double sensitivity = 0.00456;

More Data Types

- The last data type to be introduced is **char** which is the short form for character.
- char represents a single character value like letter, digit or special symbol.
- Each character must be enclosed by single quotes.
- Examples :

```
char letter = 'A';
char digit = '2';
char asterisk = '*';
```

More Data Types

- Character values are represented internally in a computer using the ASCII character coding system..

	0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
1	NL	VT	FF	CR	SO	SI	DLE	DC1	DC2	DC3
2	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
3	RS	US		!	"	#	$	%	&	'
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	'	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	DEL		

Check this out on ASCII
http://computer.howstuffworks.com/bytes2.htm

Strings

- Recall that a character variable can only store a single character. Example:

```
char letter = 'A';
```

- If we want to represent a sequence of characters, e.g. "John", we need to use a string.
- A string is a sequence of characters enclosed by double quotes. Example :

```
string club;
club = "Manchester United";
```

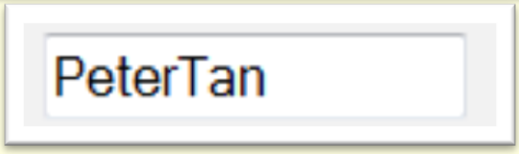
- We can join or concatenate strings too. Example :

```
string club1, club2, display;
club1 = "ManU";
club2 = "Arsenal";
display = club1 + "  and "  + club2;
// display stores "ManU  and Arsenal"
```

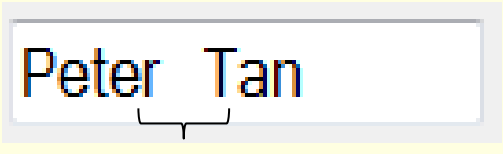
String Methods

- String has several methods that allow you to control its output. Let us look at some of them.
- PadRight
  - PadRight adds spaces to the right of a string.
  - Example:

```
string name = "Peter";
string surname = "Tan";
txtDisplay.Text = name + surname;
```




```
string name = "Peter";
string surname = "Tan";
txtDisplay.Text = name.PadRight(8) + surname;
```



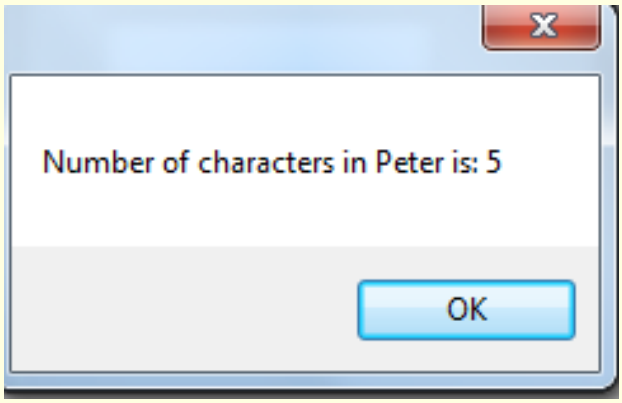
3 spaces

String Methods

- PadLeft
  - This is similar to PadRight except it adds spaces to the left of a string.

- Length
  - To determine the number of characters in a string
  - Example:

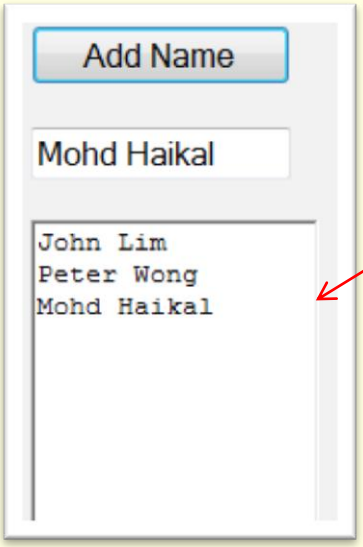
```
string name = "Peter";
int numberChar = name.Length;

MessageBox.Show("Number of characters in Peter is: " + numberChar.ToString());
```



String Methods

- To display strings in a Window Form, we can use a Label, Text Box or Rich Text Box.
- If you need to display multiple lines of Text on a form, Rich Text Box will be a useful control. Since it has an AppendText method to append text to itself. Let us see how it works:

```
private void btnAddName_Click(object sender,
{
    rtbNameList.AppendText(txtName.Text
        + Environment.NewLine);
}
```



Use Courier New font for Rich Text Box so that spaces are aligned properly

Operator Precedence

- The basic operators are covered in last lesson are:

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

- So far, we have seen basic operations like

```
totalMarks = intEnglish + intMaths;
```

The result of the addition is stored in totalMarks

- For more complex operations, we need to understand its order of precedence

Operator Precedence

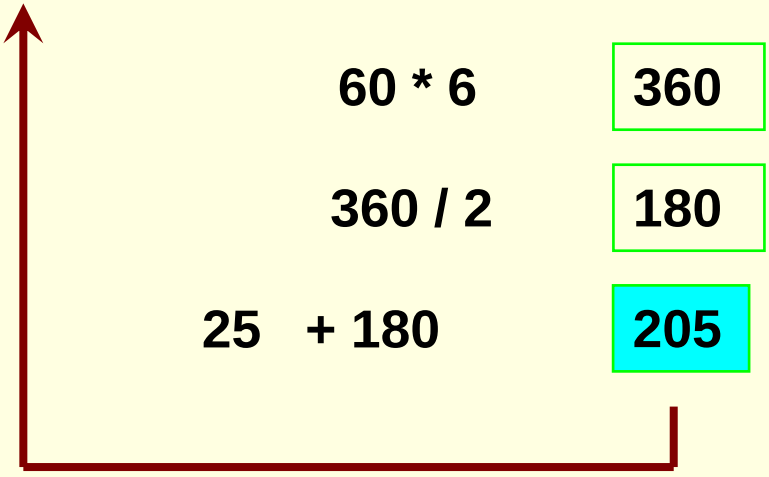
- In the table below, 1 has highest order of precedence.

		Order of precedence
( )	Left to Right	**1**
* / %	Left to Right	**2**
+ -	Left to Right	**3**
=	Right to Left	**4**

Example:

scale = 2, n = 6

butter = 25 + 60*n/scale



A Quick Review

Let's summarize what we have learnt so far:

- Variables
  - Variables created within methods are called local variables. They will be discarded once the methods exits
  - Variables created outside methods within its own class are called class variables. Class variables retain their values so long as a program is running

- char and strings
  - char data type represents a single character value like letter, digit or special symbol. Example char alpha = 'A';

A Quick Review

- char and strings
  - A string is a sequence of characters enclosed by double quotes. Example:

    string name = “John”;

- String methods
  - ✓ PadRight
  - ✓ PadLeft
  - ✓ Length

- Text Box, Rich Text Box

A Quick Review

- Mathematical operations follow an order of precedence:

		Order of precedence
( )	Left to Right	**1**
* / %	Left to Right	**2**
+ -	Left to Right	**3**
=	Right to Left	**4**

- Let us now examine a practical example.

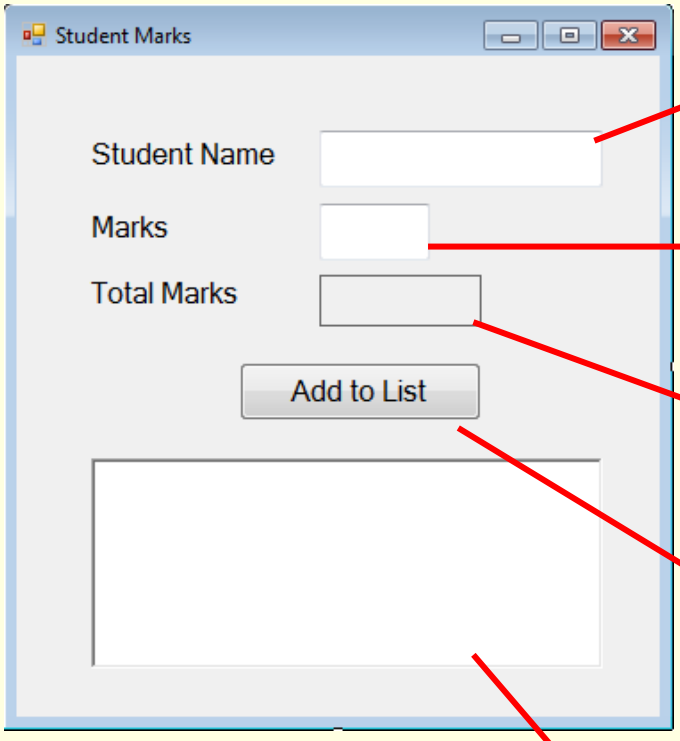
Example 1: A Student Marks Entry Form

- **Problem Statement**: Create a Form to accept one or more entry of the following:
  - Name of customer
  - Marks

  Each time the button is clicked, the marks are total up and displayed. Each student and marks are also displayed as a record per line.
- **Use Case Definitions**
  1. User enters Name
  2. User enter Marks
  3. User clicks Add to List, total mark will be displayed
  4. Record entered will be displayed
  5. User repeats step 1-3, new record will be added in the display

Example 1:
A Student Marks Entry Form

GUI Form Design



Text Box; txtName

_____

_____

_____

_____

_____

Fill in the type of control and control name

Example 1:
A Student Marks Entry Form

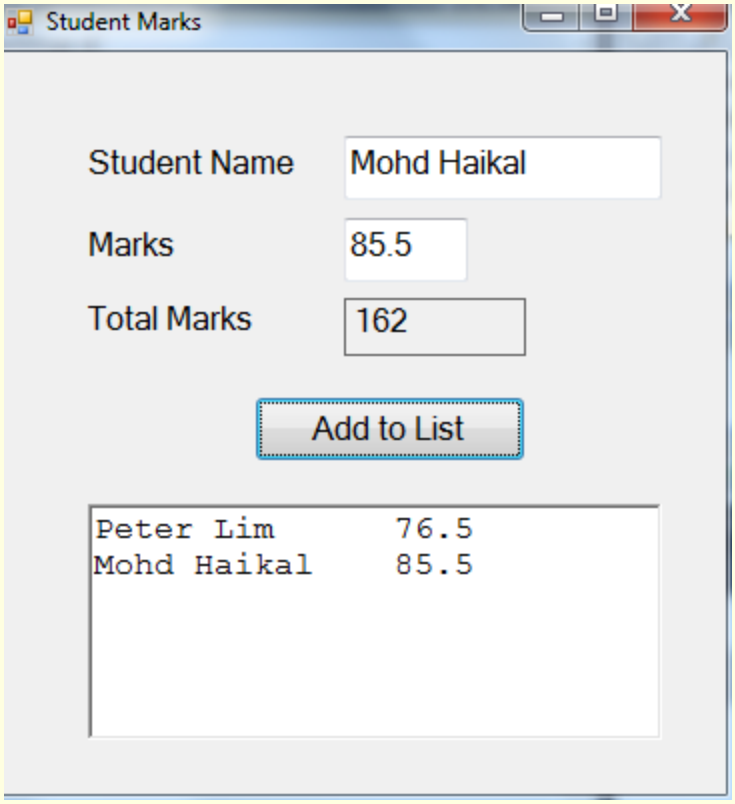
- Code:

```
// class variable
// to store total marks
1 float totalMarks = 0;

private void btnAdd_Click(object sender, EventArgs e)
{
    // add marks of each student to total marks
2   totalMarks = totalMarks + float.Parse(txtMarks.Text);
    // display in label
    lblTotal.Text = totalMarks.ToString();
    string studentName = txtName.Text;
    /// format display
3   string display = studentName.PadRight(15) + txtMarks.Text
   // append to rich text box
4   rtbDisplay.AppendText(display + Environment.NewLine);
}
```

1. Create class variable to store total marks
2. Add marks entered into totalMarks
3. Format display with PadRight
   "**+**" will join the studentName and txtMarks for display
4. Append to Rich Text Box

Example 1:
A Student Marks Entry Form

- Sample Output:



Example 1: Debug - A Student Marks Entry Form

- Tutor distributes the demo file for example 1
- Compile and Build the application
- Set a break point at

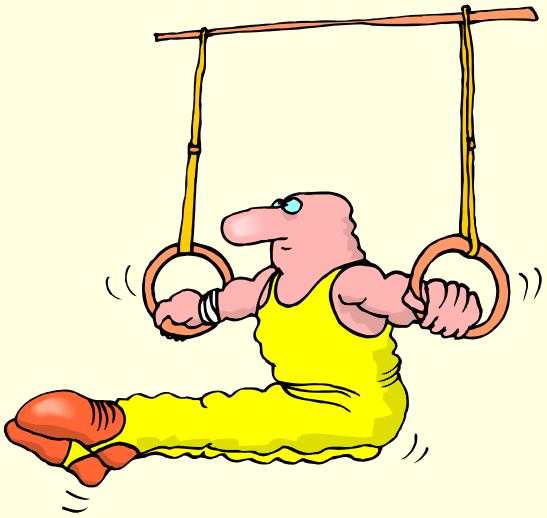
2 totalMark = totalMark +
float.Parse (txtMark.text);

- Watch the totalMark variable
- Explain the line.

totalMark = totalMark +
float.Parse (txtMark.text)

Summary

- Variables have scope – local and class
- Additional data types include long, double and char
- Mathematical operations have an order of precedence
- char data type represents a single character value
- A string is a sequence of characters enclosed by double quotes.



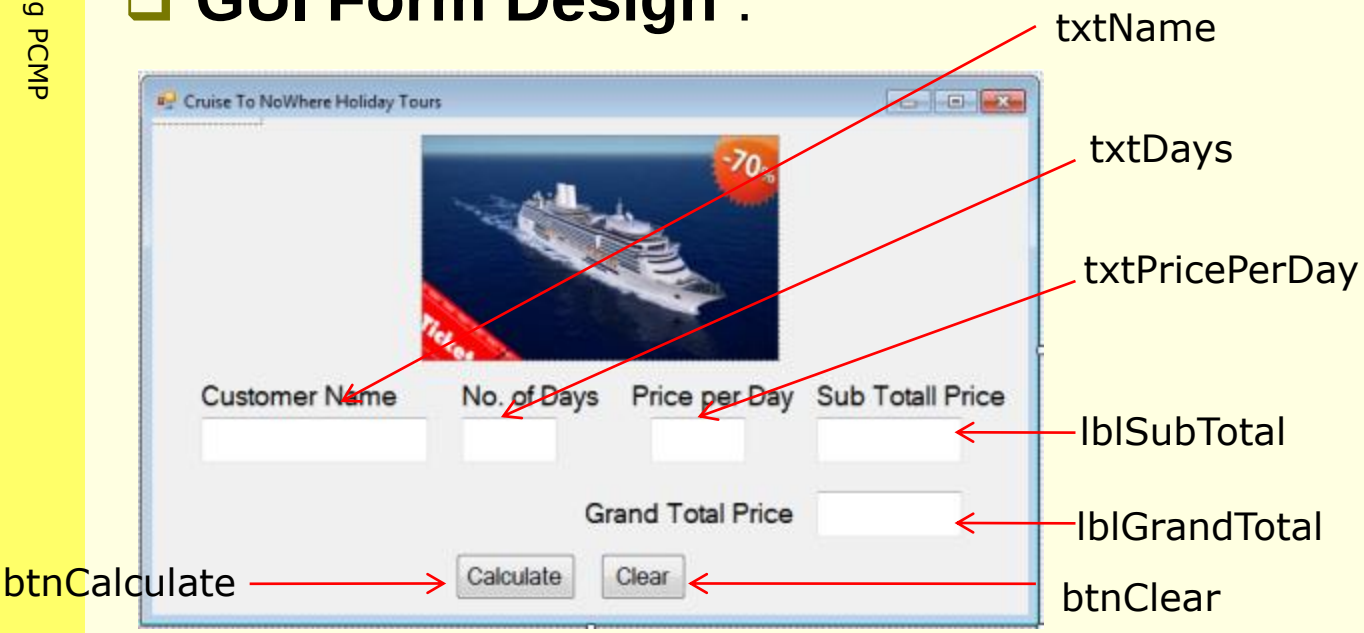
Example 2: A Holiday Cruise Booking Form

- **Problem Statement**: Create a Form to accept the following:
  - Name of customer
  - Sex
  - No of days selected for cruise
  - Price per day

  The application should have the following features:
  - Allow user to enter 1 or more customer records
  - Calculate sub total price paid by each customer
  - Display grand total price for all customers
- **Use Case Definition**
  1. User enters Name
  2. User enters Sex
  3. User enters No of Days
  4. User enters Price per day
  5. User clicks on Calculate Button, sub total and grand total prices will be displayed
  6. User clicks on Clear Button, both sub total and grand total will be cleared

Example 2:
A Holiday Cruise Booking Form

❑ GUI Form Design :



Think about the Pseudo code or program structure:

1) Read inputs into variables
2) Do calculations
3) Display results (outputs)

Example 2:
A Holiday Cruise Booking Form

❑ Codes:

```
public partial class frmCost : Form
{
    // class variable
1   float grandTotalPrice = 0.0f;
    private void btnCalculate_Click(object sender, EventArgs e)
    {
2       // create local  variables
        int numberOfDays;
        float pricePerDay, subTotalPrice;

        // convert info entered by user and store in variables
3       numberOfDays = int.Parse(txtDays.Text);
        pricePerDay = float.Parse(txtPricePerDay.Text);
```

TO DO : Explain what code does

1

2

3

Example 2: A Holiday Cruise Booking Form

Codes (cont'd):

```
    // Display in subtotalprice text box
    // method ToString("C") will display value in currency format
4   subTotalPrice = numberOfDays * pricePerDay;
    lblSubTotal.Text = subTotalPrice.ToString("C");

    // calculate Grand total
5   grandTotalPrice = grandTotalPrice + numberOfDays * pricePerDay;
6   lblGrandTotal.Text = grandTotalPrice.ToString("C");


}
```

4

5 TO DO : Explain what code does

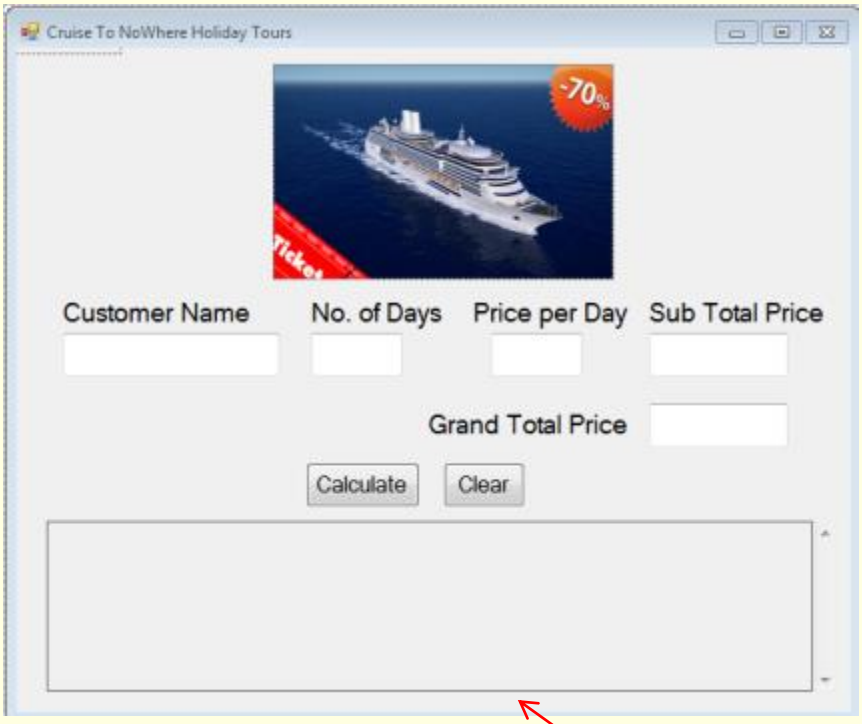
6

Example 2:
A Holiday Cruise Booking Form

- Sample Output

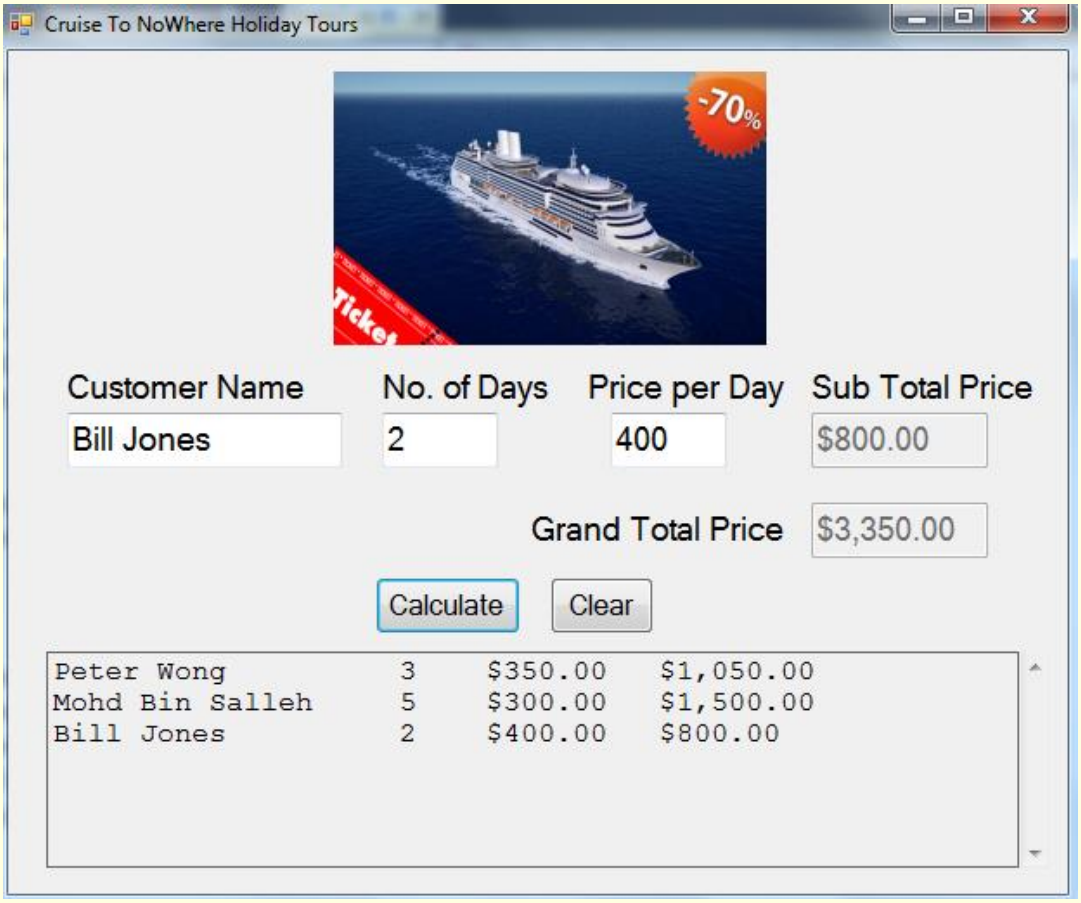
Example 2:
A Holiday Cruise Booking Form

- You will notice that so far in this program, the records entered are not displayed on the form cumulatively.
- Let us add a TextBox to display cumulatively every record entered by the user.



Example 2:
A Holiday Cruise Booking Form

- Sample Output



Example 2:
A Holiday Cruise Booking Form

```
    // calculate Grand total
    grandTotalPrice = grandTotalPrice + numberOfDays * pricePerDay;
    lblGrandTotal.Text = grandTotalPrice.ToString("C");

    string displayRecord;

    displayRecord =txtName.Text.PadRight(20)+ numberOfDays.ToString().PadRight(5);
    displayRecord = displayRecord + pricePerDay.ToString("C").PadRight(10);
    displayRecord = displayRecord + subTotalPrice.ToString("C");
    // display record in txtDisplay
    txtDisplay.Text = txtDisplay.Text + displayRecord + Environment.NewLine;
}
```

TO DO : Explain what code does

Review Questions on Example 1

Reference to Example1's code:

1. Name the local variables.

2. Name the class variable.

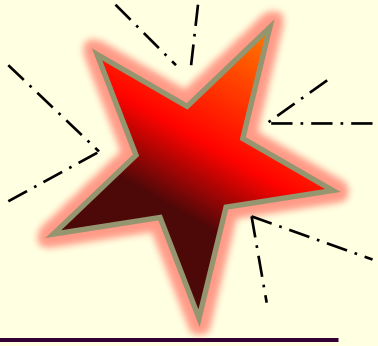
3. State the reason for using the class variable in Example1.

4. Is it possible not to use Class variable in the Example? Yes/ No Why?

Practical 2C

Q1. Suggests what data type you would use for the following. Write the C# code to define and assign the data.

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



Practical 2C

Q2. Create a window application for the following

❑ Problem Statement:

Create a form to compute the total cost and grand total cost of downloads. The form accepts Number of Song Downloads.

It computes the total and grand cost and display it.

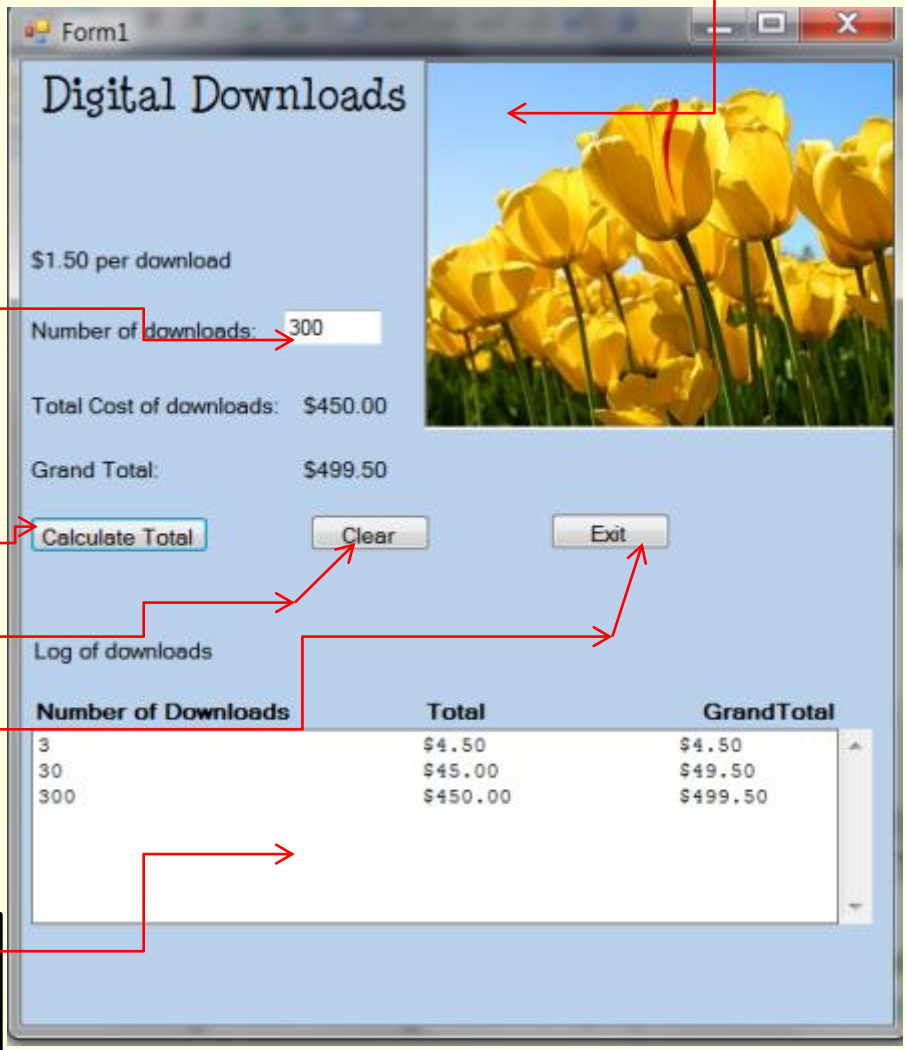
❑ Use Case Definition

1. User enters Number of Song Downloads
2. User clicks on Calculate Button, it displays the total cost, grand total cost and the log details.
3. User clicks on Clear Button, to clear data entry, computed total, grand total cost and the update the log of downloads.
4. User clicks on Exit Button to close the window
5. User is allowed to enter 1 or more downloads
6. Display the log of all downloads

Practical 2C

❑ GUI Form Design

pictureBox1
Select image as Tulips.jpg



txtNoDownload

btnTotal

btnClear

btnExit

txtLog
Set Font properties as Courier New, 7.8pt **

** Need to set font type to Courier New. This font is proportionate so that the result displayed will be aligned after pad right with spaces.

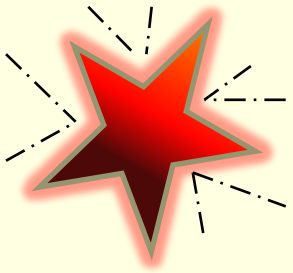
Practical 2C

- Tasks:
  1. Open a new window application
  2. Create the form and controls objects as shown in the GUI form design
  3. Define the properties
  4. Create a class variable for the grandtotal
  5. Code the details in **btnTotal_Click**
  6. Code the details in **btnClear_Click**

Hints:

- You need to apply the concepts of:
  - Local and class variables
  - Accumulative adding
  - Concatenate strings
  - Format string using PadRight()
  - Adding picture to form

Practical 2C



Q3. Problem statement:

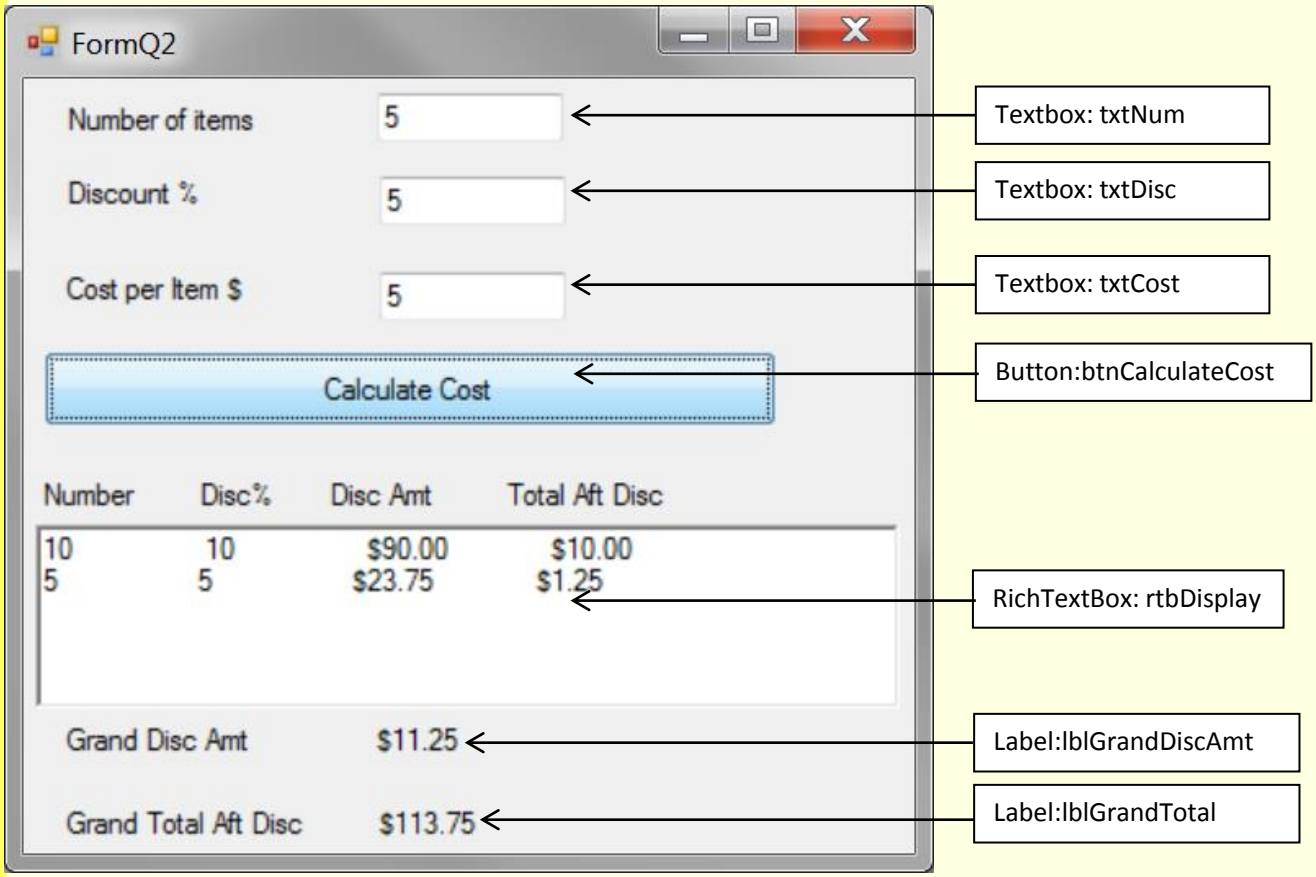
Create a window application to read in Number of items to be purchased (e.g 1), discount percentage (e.g 10.0%) and Cost per item (.e.g 10.0).

When user clicks on Calculate Cost button, program:

- Calculates **Discount Amount** and the **Total after discount**
- Accumulates the total discount amount (save amount) of each purchase in **GrandDisc**
- Accumulates the discounted total of each purchase in **GrandTotal**
- Displays result in a rich text box.

- Each line in rich text box depicts one purchase. Program allows entries of multiple purchases.

Practical 2C



The above figure shows example of purchases.

End of Topic 2C

Storing Data And Performing Math Operations Part II