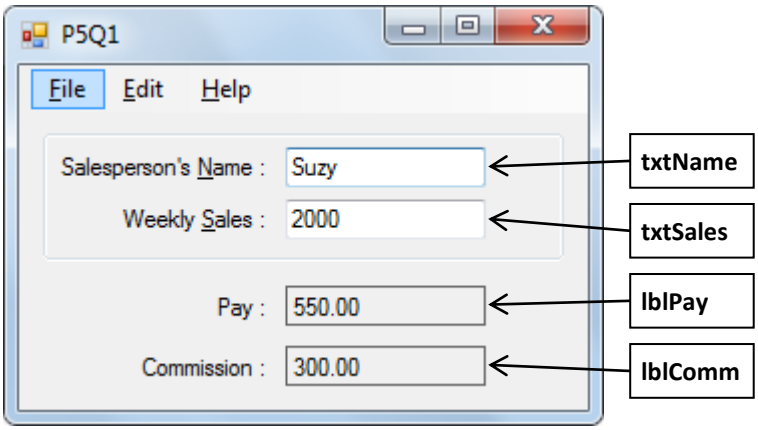
Practical 5: Menus, Common Dialogs and Function Procedures

- **Prepare** your solution at home, and **present** your solution during practical hours.
- Turn on **Option Explicit** and **Option Strict** for the project.
- Create **only one project** (with multiple forms) for the entire practical exercise.
- Program a **switcher** (main form) for launching different forms for different questions.
- Use appropriate and meaningful names for **GUI controls**.
- Improve the **usability** of your program, even if you are not explicitly asked to do so.
- Run the given **executable demo** to have a feeling of how the program should work.

Q1.



- Create the form as shown above. The form accepts **salesperson's name** and **weekly sales** as inputs. After that, it calculates and displays the **pay** and **commission** as outputs.

- The form has a menu (**MenuStrip** control) with the following menu items:

File	Edit	Help
Pay Summary ——— Exit	Clear Reset ——— Font Color	About

Name each menu item by following the recommended VB naming conventions. For example: **mnuFile**, **mnuFilePay**, **mnuFileSummary**, **mnuFileExit**, etc.

Make sure you also specify the access keys by referring to the underlined letters as shown above. In addition, assign shortcut keys to the following 3 menu items:

Menu Item	Shortcut Key
File > Pay	Ctrl + P
File > Summary	Ctrl + S
File > Exit	Alt + F4

- Declare the following class-level constants for **base pay**, **sales quota** and **commission rate**, which will later be used in the calculation of **pay** and **commission**:

```
' Class-level constants
Private Const BASE_PAY As Decimal = 250D
Private Const QUOTA As Decimal = 1000D
Private Const COMM_RATE As Decimal = 0.15D
```

- Create a function named **GetComm()** that accepts **weekly sales** as input parameter. If **weekly sales** is equal to or greater than the **sales quota** (i.e. RM 1000.00), calculate and return the **commission** using the following formula:

```
commission = weekly sales * commission rate
```

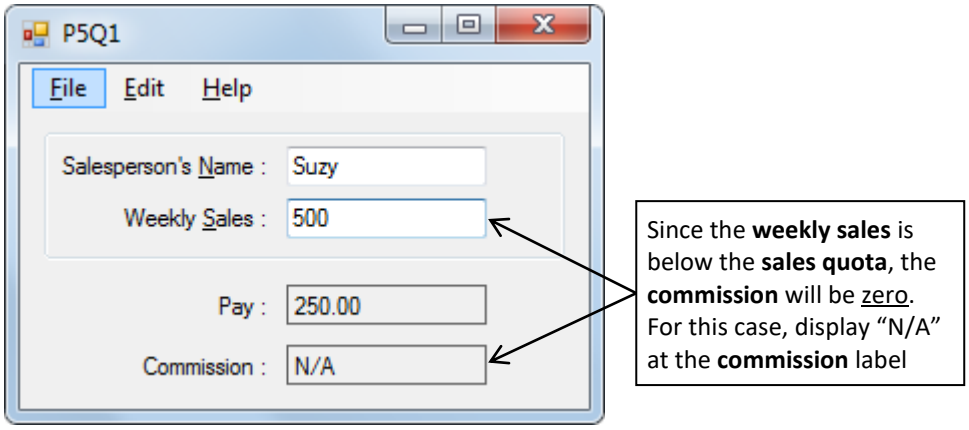
Otherwise (i.e. **weekly sales** is less than the **sales quota**), return zero **commission**. The function signature is shown below:

```
Private Function GetComm(sales As Decimal) As Decimal
    ' Complete the function
End Function
```

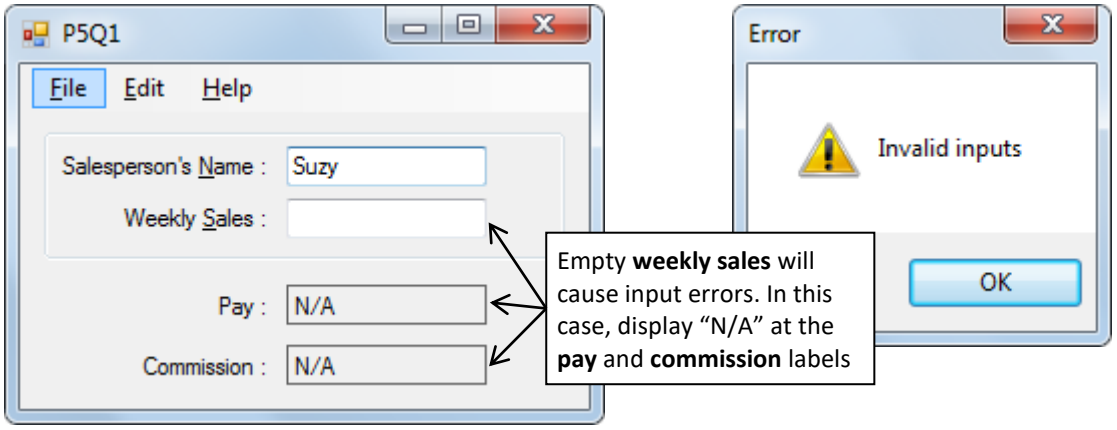
- When the **File > Pay** menu item is clicked, read the **weekly sales**. After that, calculate and display the **pay** and **commission**. Use the **GetComm()** function to obtain the **commission**, whereas **pay** can be calculated using the following formula:

```
pay = base pay + commission
```

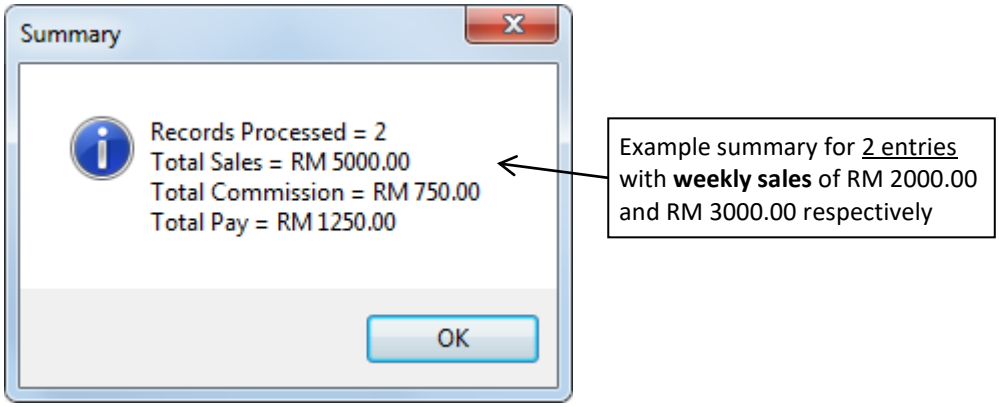
Display and format both **pay** and **commission** to 2 decimal places. If it is a zero **commission**, display "N/A" at the **commission** label (rather than leave it as 0.00). For example:



In addition, use a **try-catch** block to handle possible input errors. In the case of input errors, display "N/A" at the **pay** and **commission** labels, as well as show a message box to alert the user. For example:



- When the **File > Summary** menu item is clicked, show a message box that displays the count of **records processed**, **total sales**, **total commission** and **total pay** for ALL salespersons that have been entered so far.



  You will need to declare the relevant class-level variables for accumulating the required count and totals. You will also need to modify the event handler for **File > Pay** menu item for increasing the count and totals accordingly.

- When the **File > Exit** menu item is clicked, close the form.

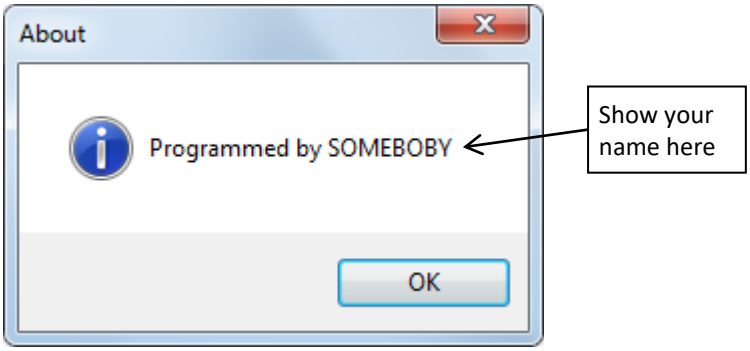
- When the **Edit > Clear** menu item is clicked, empty the textboxes and labels. Set the focus at **salesperson's name**.

- When the **Edit > Reset** menu item is clicked, reset the accumulated count and totals back to zero (i.e. the summary message box should show zeros for the count and totals).

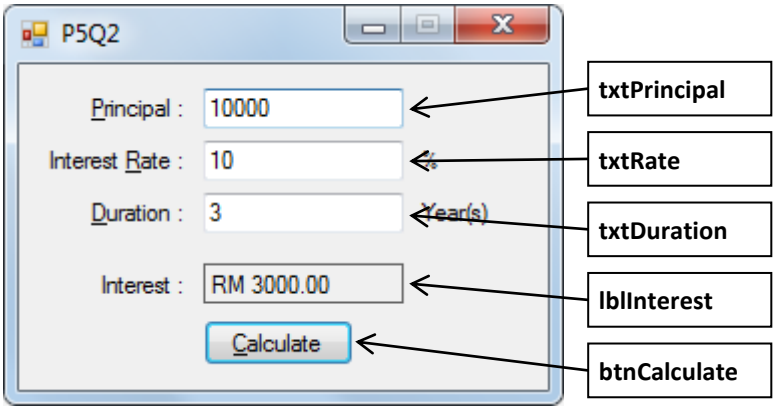
- When the **Edit > Font** menu item is clicked, show a font dialog (**FontDialog** control) to allow the user to change the font for the **pay** and **commission** labels. Before showing the dialog, ensuring the current label font is used as the dialog's default font.

- When the **Edit > Color** menu item is clicked, show a color dialog (**ColorDialog** control) to allow the user to change the text color for the **pay** and **commission** labels. Before showing the dialog, ensuring the current label text color is used as the dialog's default color.

- When the **Help > About** menu item is clicked, show a message box that displays the programmer's name (i.e. your name). For example:



Q2.

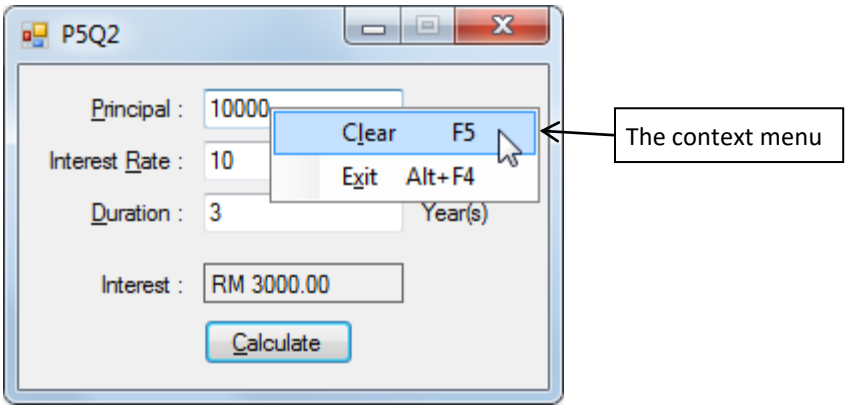


- Create the form as shown above. The form accepts a **principal**, **interest rate** and **duration** as inputs. After that, it calculates and displays the **interest** as output.

- The form has a context menu (**ContextMenuStrip** control) with the following menu items:

Menu Item	Name	Shortcut Key
Clear	mnuCtxClear	F5
Exit	mnuCtxExit	Alt + F4

Make sure you also specify the access keys by referring to the underlined letters as shown above. In addition, assign shortcut keys to the menu items. In run-time, the context menu is as follows:



- Attach the context menu to the form, **txtPrincipal**, **txtRate** and **txtDuration**, so that when the user right-clicks on these items, the context menu will be shown.

- Create a function named **GetInterest()** that accepts 3 input parameters: **principal**, **interest rate** and **duration**, as shown below:

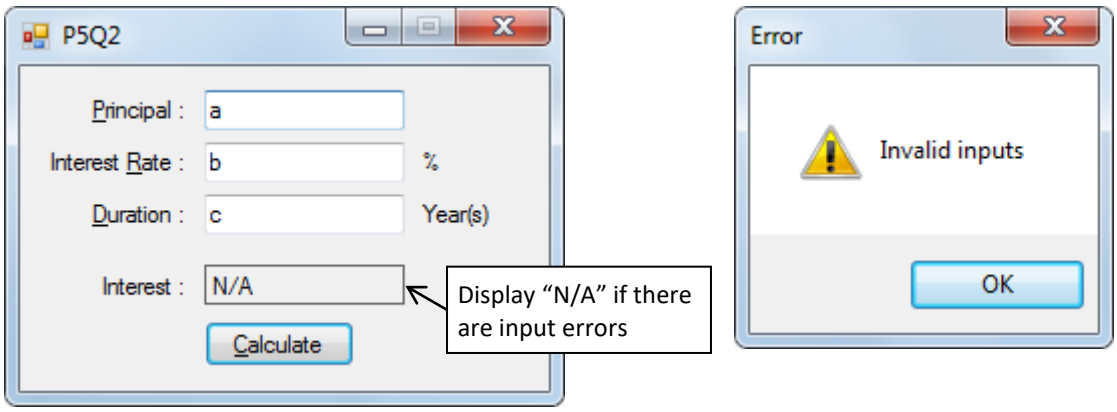
```
Private Function GetInterest(principal As Decimal,
                             rate As Decimal,
                             duration As Integer) As Decimal
    ' Complete the function
End Function
```

The function should calculate and return the **interest** using the following formula:

```
interest = principal * interest rate * duration
```

- When the **Calculate** button is clicked, read the 3 **inputs**. After that, calculate and display the **interest** by using the **GetInterest()** function. Beware that you will need to convert the **interest rate** from integer (e.g. 10%) to decimal (e.g. 0.10) before you pass it to the function.

  In addition, use a **try-catch** block to handle possible input errors. In the case of input errors, display "N/A" at the **interest** label, as well as show a message box to alert the user. For example:



- When the **Clear** context menu item is clicked, empty the textboxes and label. Set the focus at **principal**.

- When the **Exit** context menu item is clicked, close the form.

Q3.

- Add a new VB.NET module file to the project (right-click the project in Solution Explorer, select **Add > Module**). Name the module file as "**Helper.vb**".

```
Module Helper

End Module
```

- Add the following 5 public functions to the **Helper** module:

(a) Write a function named **GetTriangleArea()**. The function accepts 2 parameters: **a** (Double) and **b** (Double), which represent the 2 perpendicular sides of a right-angled triangle. The function then calculates and returns the **area** (Double) of the triangle.

```
area = 0.5 * a * b
```

```
Public Function GetTriangleArea(a As Double, b As Double) As Double
    ' Complete the function
End Function
```

(b) Write a function named **GetCircleArea()**. The function accepts 1 parameter: **radius** (Double), which represents the radius of a circle. The function then calculates and returns the **area** (Double) of the circle.

```
area = Math.PI * radius ^ 2
```

```
Public Function GetCircleArea(radius As Double) As Double
    ' Complete the function
End Function
```

(c) Write a function named **InchToCM()**. The function accepts 1 parameter: **inch** (Double). The function then converts and returns the unit in **cm** (Double).

```
cm = inch * 2.54
```

```
Public Function InchToCM(inch As Double) As Double
    ' Complete the function
End Function
```

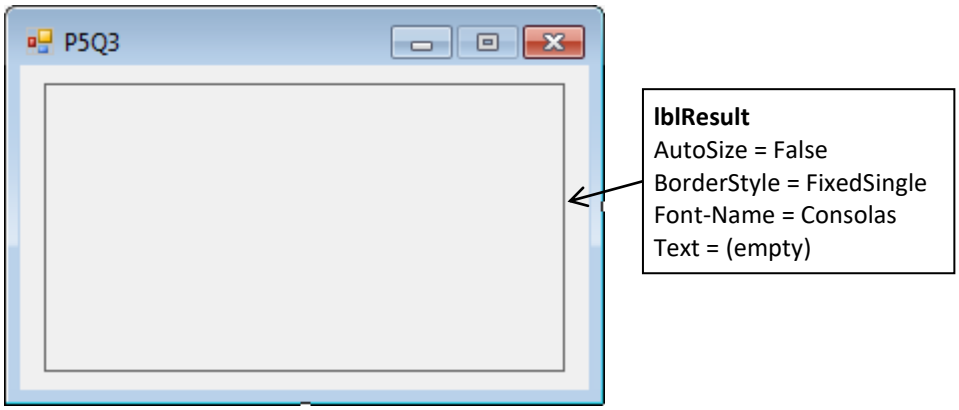
(d) Write a function named **DayOfWeek()**. The function accepts 1 parameter: **day** (Integer), and returns the correspondent **name** (String) of the day (e.g. 0 returns "Sunday", 1 returns "Monday", and so on). If the input is out of range, return "N/A".

```
Public Function DayOfWeek(day As Integer) As String
    ' Complete the function
End Function
```

(e) Write a function named **GetGrade()**. The function accepts 1 parameter: **mark** (integer), and return the correspondent **grade** (String) based on the following table:

Mark	Grade
90 – 100	A
80 – 89	B
60 – 79	C
50 – 59	D
0 – 49	F
Out of range	X

- Create a form as shown below, which has only 1 big label:



- Program the **Load** event handler of the form, so that the 5 functions in the **Helper** module will be called with the respective inputs, and the results returned are displayed on **lblResult**:

```
Private Sub FrmP5Q3_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    lblResult.Text = "(a) " & Helper.GetTriangleArea(12.3, 45.6) & vbNewLine &
                     "(b) " & Helper.GetCircleArea(10.0) & vbNewLine &
                     "(c) " & Helper.InchToCM(10.0) & vbNewLine &
                     "(d) " & Helper.DayOfWeek(3) & vbNewLine &
                     "(c) " & Helper.GetGrade(67) & vbNewLine
End Sub
```

- Run the form and check the results: