Chapter 13

Formatting Functions

- Learning how to use the Formatting Functions

13.1 Format Function

The **Format** function is a very powerful formatting function that can display the numeric values in various forms. There are two types of Format function, one of them is the built-in or predefined format, and the user can define another one.

(i) The format of the predefined Format function is

Format (n, "style argument")

Where n is a number and the list of style arguments is given in Table 13.1

Style argument	Explanation	Example
General Number	Displays the number without having separators between thousands	Format(8972.234, "General Number")=8972.234
Fixed	Displays the number without having separators between thousands and rounds it up to two decimal places.	Format(8972.2, "Fixed")=8972.23
Standard	Displays the number with separators or separators between thousands and rounds it up to two decimal places.	Format(6648972.265, "Standard")= 6,648,972.27
Currency	To display the number with the dollar sign in front has separators between thousands as well as rounding it up to two decimal places.	Format(6648972.265, "Currency")= $6,648,972.27
Percent	Converts the number to the percentage form, displays a % sign, and rounds it up to two decimal places.	Format(0.56324, "Percent")=56.32 %

Table 13.1: The Format Function

Example 13.1

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click, Button5.Click, Button4.Click,
Button3.Click
Label1.Text = Format(8972.234, "General Number")
Label2.Text = Format(8972.2, "Fixed")
Label3.Text = Format(6648972.265, "Standard")
Label4.Text = Format(6648972.265, "Currency")
Label5.Text = Format(0.56324, "Percent")
End Sub
```

The Output window is shown below:

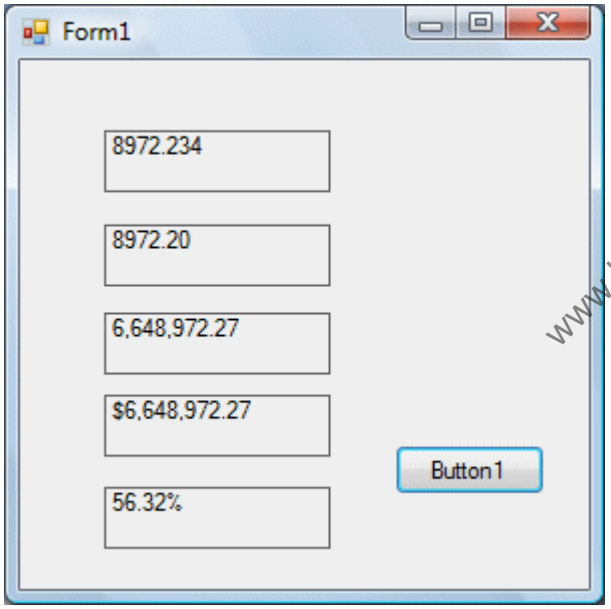

Figure 13.1

(ii) The format of the user-defined Format function is

Format (n, "user's format")

Although it is known as user-defined format, we still need to follow certain formatting styles. Examples of user-defined formatting style are listed in Table 13.2

Example	Explanation	Output
Format(781234.57,”0”)	Rounds to whole number without separators between thousands	781235
Format(781234.57,”0.0”)	Rounds to one decimal place without separators between thousands	781234.6
Format(781234.576,”0.00”)	Rounds to two decimal places without separators between thousands	781234.58
Format(781234.576,”#,##0.00”)	Rounds to two decimal places with separators between thousands	781,234.58
Format(781234.576,”$#,##0.00”)	Shows dollar sign and rounds to 2 decimal places with separators between thousands	$781,234.58
Format(0.576,”0%”)	Converts to percentage form without decimal places.	58%
Format(0.5768,”0.00%”)	Converts to percentage form with 2 decimal places	57.68%

Table 13.2: User’s Defined Functions

Example 13.2

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click, Button5.Click, Button4.Click,
Button3.Click
Label1.Text = Format(8972.234, "0.0")
Label2.Text = Format(8972.2345, "0.00")
Label3.Text = Format(6648972.265, "#,##0.00")
Label4.Text = Format(6648972.265, "$#,##0.00")
Label5.Text = Format(0.56324, "0%")
End Sub
```

The Output window is shown below:

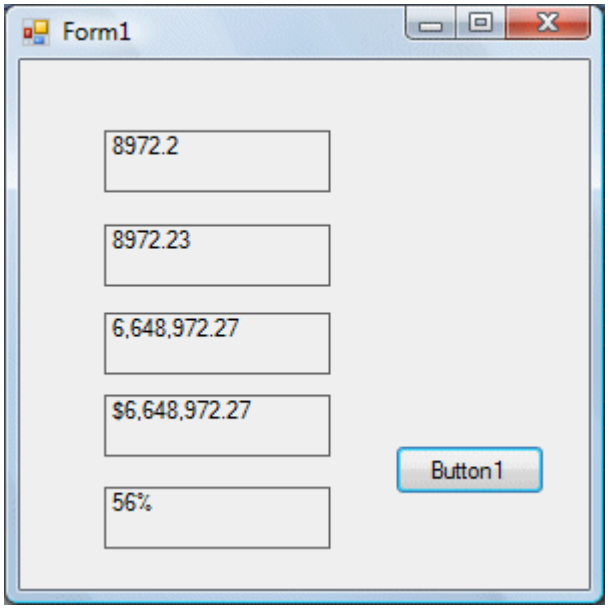

Figure 13.2: User's Defined Functions

13.2 Formatting Using ToString Method

Other than using the Format function, VB.Net has introduced the ToString method to format output. It is used together with the standard numeric format specifiers such as "c" which stand for currency. Some of the most common numeric specifiers are listed in the table below:

Format specifier	Explanation	Examples
"C"	i) Displays a currency value. The default is the US currency $ and in two decimal places. ii) To display other currency, add a culture code that specifies a country. For example, for Great Britain, you add en-GB using the keyword "CultureInfo. CreateSpecificCulture"	**Dim myNum as Single =2011.123456** myNum.ToString("C")= $2011.12 myNum.ToString("C4")= $2011.1234 myNum.ToString("C3", CultureInfo. CreateSpecificCulture("en-GB"))= £2011.123

	iii) Displays number of decimal digits by placing the digit after C, for example, C4 for decimal places.	
"D" or "d"	Express a Number with in integer form with specified number of digits. For example, D4 means four-digit integer.	Dim myNumber As Integer = 2012.2344 myNumber.ToString("D4")=2012
"E" or "e"	Express a number in exponential form with specified number of decimal places	Dim myNumber As Double = 2012.2344 myNumber.ToString("e3")= 2.012e+003
"P" or "p"	Mutiply a number by 100 and displayed with a percentage symbol % .	Dim myNumber As Double = 0.23456 myNumber.ToString("P2")= 23.46%
"F" or "f"	Specifies number of decimal points	Dim myNumber As Double=0.23456 myNumber.ToString(“F”)=0.23 myNumber.ToString(“F3”)=0.235

Table 13.3: Standard numeric format specifiers

* More on ToString Method

The ToString method together with the currency specifier “C” displays the output with the currency sign $ and in two decimal places. The default currency is the currency used by your computer system; in this case, it is the US currency. If you are not sure of what default currency your computer uses, you can add the keyword “CultureInfo.CurrentCulture” to the ToString method as shown in the example below:

```
FutureValue = FV.ToString("C", CultureInfo.CurrentCulture)
```

If you wish to display the output in different currencies, you can use the keyword" "CultureInfo.CreateSpecificCulture together with the culture identifiers. For example, if you want to display the output in Japanese currency, you can use the ja-JP culture identifier, as shown in the example below:

```
FutureValue = FV.ToString("C", CultureInfo. CreateSpecificCulture("ja-JP")
```

The output is in Japanese currency sign ¥ instead of the $ sign.

Summary

In this chapter, you learned how to format your output using the Format and the ToString functions.

- In section 13.1, you learned how to use various formatting styles
- In section 13.2, you learned how to format output using ToString output.