DEPARTMENT OF COMMERCE [CA]

Software Development with Visual Basic
(18BCA52C)

Semester : V

III B.Com(CA)

Unit - IV

Built in functions: String functions - Numeric functions - Date and Time functions - financial functions. Functions and Procedures Procedures and Sub Procedures.

Reference Books:

1. "Visual Basic 6.0 From the Ground Up" by Gary Cornell.
2. Visual Basic 6.0 Programming by Content Development Group.

Prepared By
Dr. S. VASANTHA
Assistant Professor.

Visual Basic Built-in Functions

Many built-in functions are offered by Visual Basic that fall under various categories. These functions are Produces that return a value.

* Date and Time functions
* Formal functions.
* String functions.

String functions:

(1) Case Conversion functions:-

This type of functions help us to Convert lower case letters to upper case letters and viceversa.

(a) LCase (b) UCase

LCase: Syntax

LCase(<string>)

example:

LCase("HELLO")

output:

hello

(b) UCase: syntax

UCase(<string>)

ex UCase("hello")

output: HELLO

(ii) Justifying and Trimming functions:

These functions are used to cut the unwanted spaces before and after the string. This type of functions should be used when in-depth String Manipulations occur.

(a) LTrim()

LTrim(String) - used to trim the spaces at the left-side of the text.

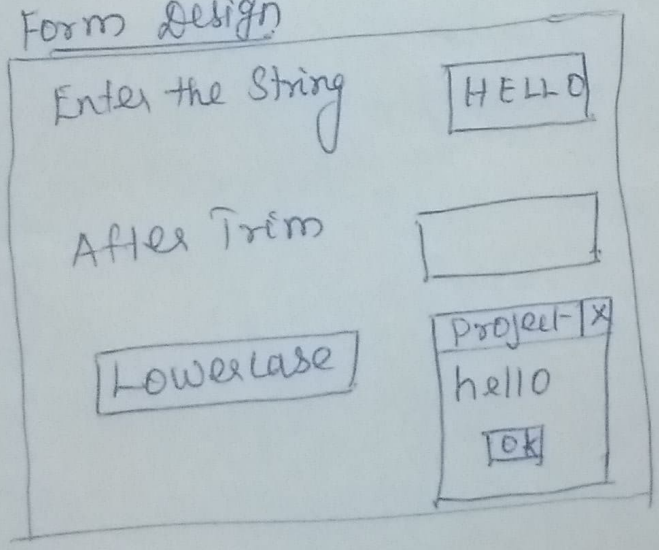
(b) RTrim()

RTrim(String) - Used to trim the spaces at the right-side of the text.

(c) Trim()

Trim(String) - Used to trim the spaces at both the sides of the Text.

Form Design



Coding

```
Private Sub Command1_Click()
Select Case Command1.Caption
Case "Uppercase": MsgBox Ucase(Text1.Text)
                  Command1.Caption = "Lowercase"
                  Text1.Text = ""
                  Text1.SetFocus
```

```
Case "Lowercase" : MsgBox LCase(Text1.Text)
                   Command1.Caption = "Lowercase"
                   Text1.Text = ""
                   Text1.SetFocus
Case "Left Trim" : Text2.Text = LTrim(Text1.Text)
                   Command1.Caption = "Right Trim"
                   Text1.Text = ""
                   Text1.SetFocus
Case "Right Trim" : Text2.Text = RTrim(Text1.Text)
                    Command1.Caption = "Trim"
                    Text1.Text = ""
                    Text1.SetFocus
Case "else" : Msgbox "Wrong Commitment"
End Select
End Sub.
```

Chr():

This function converts ASCII value to their respective characters.

Syntax:

```
Chr(charcode as long)
```

(ex)

```
Chr(val(Text1.Text))
// Assume Text1.Text = 67
```

The result will be

"C"

The SubString functions :-

(1) This means extracting specific values from the given String either to left (or) Right (or) Mid.

(1) Left
(2) Right
(3) Mid

1. Left()

This function is used to extract the specific number of characters from left of the given String.

Syntax

Left("String", Number of characters to be extracted)

example:

Msgbox left("Hello",2)

output

He

2. Right()

This function will extract the number of characters from right of the given

Syntax:

Right("string", no of characters to be extracted)

eg:

Msgbox right("Hello",2)

output

lo

(3) Mid()

This function is used to extract the String from Specified left value to Specified lengths value.

Syntax Mid("String", Start value, [lengths])

(ex) Msgbox(Mid("hello", 2, 2))

output el

Formal Functions

The formal function accepts a numeric value and Converts it to a String in the format specified by the format argument.

Syntax:

Format(expression[, format[, first day of week[, first week of year]]])

Numeric functions:-

The numeric functions are used to solve Some Mathematical Problems.

Int(Numeric value) – used to convert any type of numeric value to integer

(ex) Int(6.8) = 6

(2) Sqr(Numeric value) – Used to find the Square root of the given numeric value

(ex)
Sqr(16)
output = 4

other Numeric functions:

Log(Numeric value) – Returns neutral logarithmic value

Cos(Numeric value) – To find the Cos value of the argument

Sin(Numeric value) – To find the Sin value of the argument.

Tan(Numeric value) – To find the Tan value of the argument.

Date and Time functions

In VB we can use Date and Time functions to display System date and System time and it can perform some functions with those datas.

1. Date : which returns the current date in the system in the form of mm/dd/yyyy
2. Time : This function is used to display the current system time in the form of (HH:MM:SS)

Now : This is used to display Date and Time simultanesly.

Day : This functions is used to display only current day.

Year(Now) : This function is used to display only current year.

DateAdd() : This function returns a new date by adding two dates.

PROCEDURES AND SUB PROCEDURES

Procedures

Visual Basic programs can be broken into smaller logical components called procedures. procedures are useful for condensing repeated operations such as the frequently used calculations, text- and control manipulations, etc.

The following are the benefits of using Procedures in programming

* It is easier to debug a program with procedures, which breaks a program into discrete logical limits.
* Procedures used in one program can act as building blocks for other programs with slight modifications.

A procedure can be a Sub, Function (or) property ~~prody~~ Procedure.

Sub Procedures

A Sub procedure can be placed in standard, class and form Modules. Each Time the procedure is called, ~~the~~

the statements between Sub and End Sub are executed. The syntax for a Sub procedure is as follows.

[Private/Public] [Static] Sub Procedure name [(arglist)]

[Statements]

End Sub.

arglist is a list of argment names separated by commas. Each argument acts like a variable in the procedure.

There are two types of Sub Procedures

(a) general procedures

(b) event procedures.

Event procedures

An event procedure is a procedure block that contains the control's actual name, an underscore(_), and the event name. The following syntax represents the event procedure for a Form_Load event.

```
private Sub Form_Load()
  Statement block
End Sub.
```

General Procedures

A general Procedure is declared when Several event procedures perform the same actions. It is a good Programming practice to write common Statements in a Separate procedure and then call them in the event procedure. In order to add a general procedure following steps are followed

* The code window is opened for the Module to which the procedure is to be added.
* The Add Procedure option is chosen from the tools menu, which opens an Add Procedure dialog box
* The name of the procedure is typed in the Name Textbox.
* under Type, Sub is selected to create a Sub procedure, Function to create a function procedure (or) Property to create property procedure.
* under Scope, Public is selected to create a procedure that can be invoked outside the Module (or) Private to create a procedure only from within the Module.