

College of computer technology
Information network department
Programming with Visual BasicII



Sub Procedure and Function Procedure

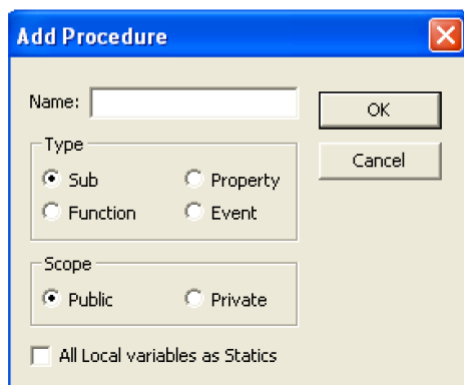
Most computer programs that solve real-world problems are much larger than those presented in the first few chapters. Experience has shown that the best way to develop and maintain a large program is to construct it from smaller pieces each of which is more manageable than the original program. This technique is called divide and conquer. This chapter describes many key features that facilitate the design, implementation, operation and maintenance of large programs.

Functions and **Subroutines** are programs designed for specific task, and could be called from the main program or from sub-procedures without pre definition or declaration. Users are allowed to call in any number of times which save the main program space, since it avoids repetition of code these subroutines could be designed by user or could be previously built. The concepts and descriptions are summarized in the following table.

Item	Subroutine	Function
Code	Sub Name (arguments) Statements End Sub	Function Name (arguments) Statements End Function
Remark	<ul style="list-style-type: none"> • Need call statement • Return values by arguments • Return many values (arguments) • Used for Input/output, condition treatment • Could be used with out arguments. 	<ul style="list-style-type: none"> • Used in arithmetic statement • Return value by its name • Return one value • Used for arithmetic's or conversion of variable type.
Call Statement	Call Name(value1,value2,,,,)	Z=name(value1)
Exit statement	Exit Sub	Exit Function

Sub Procedures

Sub procedure are created with the add procedure dialog (displayed when add procedure is selected from the tools menu). The add procedure menu item is grayed unless the code window is visible. Figure (9-1) displays the add procedure dialog. The procedure name is entered in TextBox Name and can be any valid identifier. Frame Type contains option buttons for selecting the procedure type (Sub or Function). Frame scope contains option buttons for selecting keyword public or keyword private that will procedure, we will use keyword private, which also preceded our event procedures.



add procedure dialog

College of computer technology
Information network department
Programming with Visual BasicII

=====

Once a valid name has been type into textbox name (add) has been passed, the procedure appears in the code window. Figure shows procedure (add) which we created with the add procedure dialog. The code representing (add) in figure is called the sub procedure definition.

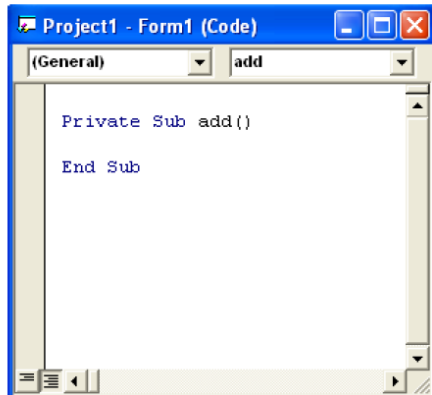


Figure : A sub procedure created with the add Procedure dialog.

Sub Procedures can also be created by typing the sub procedure directly into the code window. Once a line such as

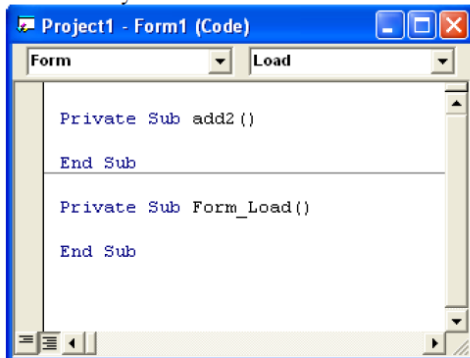
Private Sub add2 ()

Is typed and the enter key pressed, visual basic automatically creates the end sub line. Figure shows the results when (add2) is typed directly into the code window.

The line

Private Sub add2 ()

is the sub procedure header. The header contains keyword private, keyword sub, the procedure name, and parentheses. Any declarations and statements the programmer places between the header and end sub form the sub procedure body. Every time the sub procedure is called (or invoked) the body is immediately executed.



: A Sub procedure created by typing directly into the code window.

Execution of the sub procedure terminates when end sub is reached. Program execution then continues with the statement immediately following the call to (add2).

All **Sub** procedure definitions contain parentheses which may be empty (e.g., add2). Consider the following sub procedure:

College of computer technology
Information network department
Programming with Visual BasicII

=====

Private Sub Calculate (a as single, b as double)

Picture1.print a*b

End Sub

Which declares two parameter variables, (a, and b), in the parameter list. Parameter variables are declared using the **As** keyword. Parameter variables are not explicitly given a type default to **Variant**. Parameter variables receive their values from the procedure call and are used in the procedure body
The call to **Calculate** could also have been written as

Call Calculate (30,10.0)

Which uses keyword **Call** and encloses the arguments passed in a set of parentheses. The arguments passed can be variable names as well, for example, the call

Call Calculate (a, b)

Would pass a, and b to Calculate.

Example Write a code program to read three integer numbers. Using a define sub procedure (Minimum) to determine the smallest of three integers. Display the smallest value in textbox.

Solution:

```
Private Sub Command1_Click()
```

```
Dim Num1 As Single, Num2 As Single, Num3 As Single
```

```
Num1 = Fix(Text1.Text)
```

```
Num2 = Fix(Text2.Text)
```

```
Num3 = Fix(Text3.Text)
```

```
Call Minimum(Num1, Num2, Num3, min)
```

```
Text4.Text = Str(min)
```

```
End Sub
```

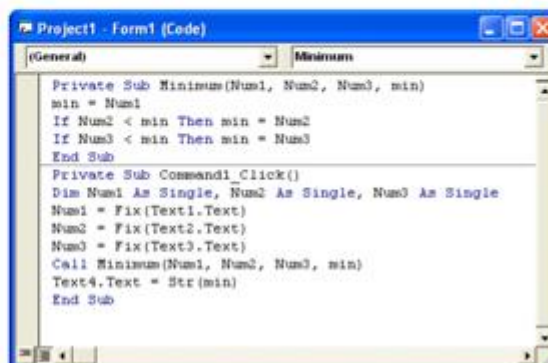
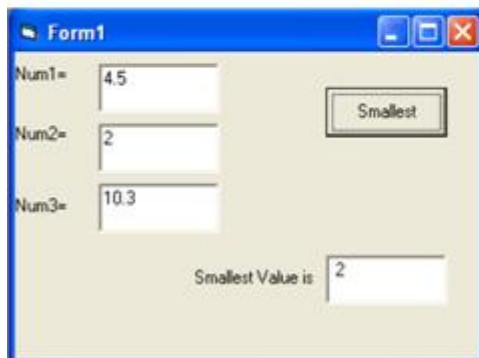
```
Private Sub Minimum(Num1, Num2, Num3, min)
```

```
min = Num1
```

```
If Num2 < min Then min = Num2
```

```
If Num3 < min Then min = Num3
```

```
End Sub
```

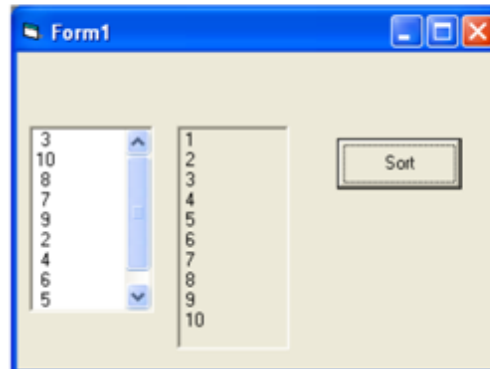


**College of computer technology
Information network department
Programming with Visual BasicII**

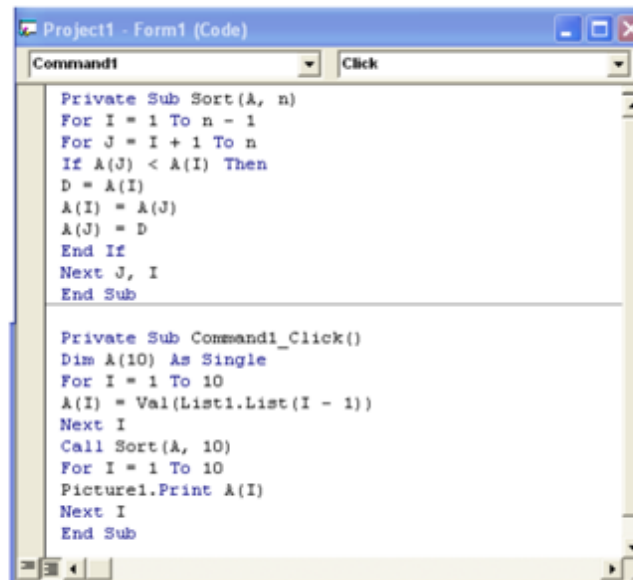
Example Write a code program to read a one dimension array A (10). Using a define sub procedure (Sort) to Sort (increasing) the array A. Display the new array into picturebox.

Solution:

```
Private Sub Command1_Click()
Dim A(10) As Single
For I = 1 To 10
A(I) = Val(List1.List(I - 1))
Next I
Call Sort(A, 10)
For I = 1 To 10
Picture1.Print A(I)
Next I
End Sub
```



```
Private Sub Sort(A, n)
For I = 1 To n - 1
For J = I + 1 To n
If A(J) < A(I) Then
D = A(I)
A(I) = A(J)
A(J) = D
End If
Next J, I
End Sub
```

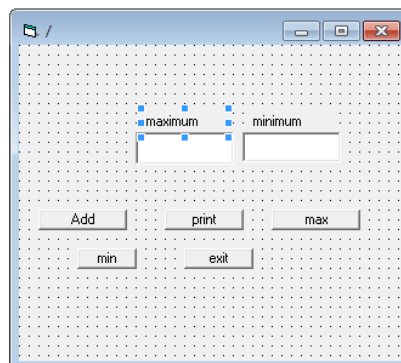


Example: write a program to read and write array of 3x3 and then find a maximum number and minimum number by a procedure

```
Dim i As Integer
Dim j As Integer
Dim arr(3, 3) As Integer
Dim arr1(3, 3) As Integer

Sub add(arr)
For i = 1 To 3
For j = 1 To 3
arr(i, j) = Val(InputBox("enter no"))
Next
Next
End Sub

Sub prn(arr)
For i = 1 To 3
```



College of computer technology
Information network department
Programming with Visual BasicII

```
For j = 1 To 3
    Print arr(i, j);
Next
Print
Next
End Sub
Sub maximum(arr)
    Max = arr(1, 1)
    For i = 1 To 3
        For j = 1 To 3
            If arr(i, j) > Max Then
                Max = arr(i, j)
            End If
        Next
    Next
    Text1.Text = Max
End Sub
Sub minimum(arr)
    Max = arr(1, 1)
    For i = 1 To 3
        For j = 1 To 3
            If arr(i, j) < Max Then
                Max = arr(i, j)
            End If
        Next
    Next
    Text2.Text = Max
End Sub

Private Sub Command1_Click()
    Call add(arr1)
End Sub

Private Sub Command2_Click()
    Call prn(arr1)
End Sub

Private Sub Command3_Click()
    Call maximum(arr1)
End Sub

Private Sub Command4_Click()
    Call minimum(arr1)
End Sub

Private Sub Command5_Click()
    End
End Sub
```

College of computer technology
Information network department
Programming with Visual BasicII

=====

Example: write a procedure to sum two numbers and write another procedure to find average of two numbers

```
Sub total(a As Integer, b As Integer)
Dim sum As Integer
sum = a + b
Form1.Text3.Text = sum
End Sub
Sub average(i As Integer, j As Integer)
Dim ave As Integer
ave = (i + j) / 2
Form1.Text4.Text = ave
End Sub
Private Sub Command1_Click()
Dim no1 As Integer
Dim no2 As Integer
no1 = Val(Text1.Text)
no2 = Val(Text2.Text)
total no1, no2
End Sub
Private Sub Command2_Click()
Dim no11 As Integer
Dim no22 As Integer
no11 = Val(Text1.Text)
no22 = Val(Text2.Text)
average no11, no22
End Sub
Private Sub Command3_Click()
End
End Sub
```