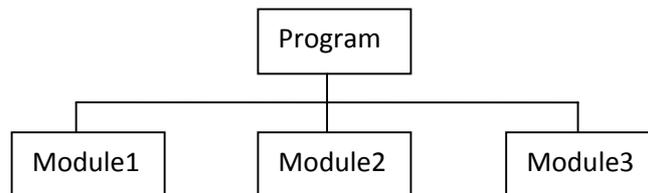


Language Features : Modularity, Procedures & Functions

Structured Programs (Modularity)

To simplify writing complex programs, most Programmers (Designers/Developers) choose to split the problem into simpler smaller tasks (modules). They then solve each small task before linking the 'modules' together as a complete program.



The modules, being smaller and simpler, are easier to design, write and test. The final program is made by combining individually tested modules, so that final checking consists of making sure that the modules work together as intended.

Advantages to structuring a program in this way are that it aids readability when designing or amending. It also reduces the size of the program (a module is written once and can be used in several parts of a program and even included in other programs).

Program Libraries and predefined Functions

Program Libraries are a collection of prewritten programs and subroutines that can be called for use in your programs. Some you have used without thinking such as:

```
Console.ReadKey()  
Console.SetCursorPosition(12,20)
```

There are other more specialised functions:

-Maths functions

Math.Round	Returns a Double value containing the number nearest the specified value.
Math.Sqrt	Returns a Double value specifying the square root of a number.
Math.Fix()	Returns the integer portion of a number. Fix(5.3333) returns 5.
Math.Pow()	Returns a number raised to a power. Math.Pow(12,2) returns 144.

-String functions

Len()	Returns the length (number of characters) of the specified string
Mid\$ ()	Returns a substring containing a specified number of characters from a string.
Left\$ ()	Returns a substring containing a specified number of characters from the beginning (left side) of a string.
UCase\$ ()	Converts all lowercase letters in a string to uppercase.
Val()	Converts a numeric expression to a number.

-Other specialised function libraries

Graphics functions

DrawLine(Pen, Point, Point)	Draws a line connecting two Points
DrawEllipse(Pen, Int32, Int32, Int32, Int32)	Draws an ellipse defined by a bounding rectangle specified by coordinates for the upper-left corner of the rectangle, a height, and a width.

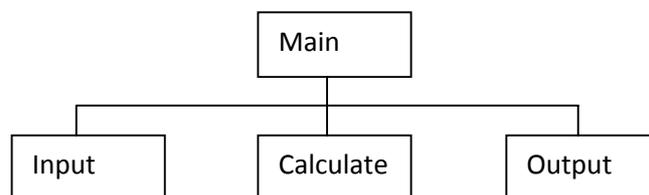
Procedures and Functions

A procedure (or Sub in Visual Basic) is used to break a program down into smaller tasks in the same way as Modularity above.

Example prog

The following program is broken down into three procedures: Input, Process and Output. The same advantages apply –ease of writing & test, aids readability, etc.

This program calculates the cost of a Travel Pass based on the Name Age and Type of Pass required (Week or Month). Those younger than 16 pay £10 for a week pass or £20 for a Month pass. Adults pay £12 for a month pass or £25 for a month pass.



'These are **Global Variables** so can be used in all lower level routines.
 Dim Name as String
 Dim Age as Integer
 Dim PassType as String
 Dim Cost as Decimal

```

Sub Main( )
    Input()      'calls the Input subroutine
    Calculate()  'calls the Calculate subroutine
    Output()     'calls the Output subroutine
End Sub

Sub Input( )
    Console.Clear()
    Console.WriteLine("Enter Name :")
    Name= Console.ReadLine()

    Console.WriteLine("Enter Age :")
    Age= Console.ReadLine()

    Console.WriteLine("Pass Type (Week or Month) :")
    TypePass= Console.ReadLine()
EndSub

Sub Calculate ( )
    If PassType = "Week" then
        If Age < 16 then
            Cost = 10
        Else
            Cost = 12
        Endif
    Endif
    If PassType = "Month" then
        If Age < 16 Then
            Cost = 20
        Else
            Cost = 25
        Endif
    Endif
End Sub

Sub Output( )
    Console.Write("Name :")
    Console.WriteLine("{0}",Name)
    Console.Write("Pass Type :")
    Console.Write("Age :")
    Console.WriteLine("{0}",Age)

    Console.WriteLine("{0}",PassType)
    Console.Write("Cost :")
    Console.WriteLine("{0}",Cost)
End Sub

```

The diagram illustrates the execution flow of the code. It shows three subroutines: Input(), Calculate(), and Output(). Arrows indicate the following sequence: Main() calls Input(), Input() calls Calculate(), and Calculate() calls Output().

Exercise

-
- 1) Run the program and test that it works correctly.