



Syllabus Content:

11.1 Built-in functions

Use a subset of the built-in functions and program library. This should include those used for:

-  string/character manipulation
-  formatting of numbers

Programming environments provide many built-in functions. Some of them are always available to use; some need to be imported from specialist module libraries.

Built-in functions

STRING Functions

LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING returns
leftmost x characters from ThisString
Example: LEFT("ABCDEFGH", 3) returns "ABC"

RIGHT(ThisString : STRING, x : INTEGER) RETURNS STRING returns
rightmost x characters from ThisString
Example: RIGHT("ABCDEFGH", 3) returns "FGH"

MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING returns a string
of length y starting at position x from ThisString
Example: MID("ABCDEFGH", 2, 3) returns "BCD"

LENGTH(ThisString : STRING) RETURNS INTEGER returns the integer
value representing the length of ThisString
Example: LENGTH("Happy Days") returns 10

LCASE(ThisChar : CHAR) RETURNS CHAR
returns the character value representing the lower case equivalent of ThisChar
Characters that are not upper case alphabetic are returned unchanged
Example: LCASE('W') returns 'w'

UCASE(ThisChar : CHAR) RETURNS CHAR
returns the character value representing the upper case equivalent of ThisChar
Characters that are not lower case alphabetic are returned unchanged
Example: UCASE('a') returns 'A'

1

TO_UPPER(ThisString : STRING) RETURNS STRING
returns a string formed by converting all characters of ThisString to upper case
Example: TO_UPPER("Error 803") returns "ERROR 803"

TO_LOWER(ThisString : STRING) RETURNS STRING
returns a string formed by converting all characters of ThisString to lower case

Example: TO_LOWER("JIM 803") returns "jim 803"

NUM_TO_STR(x : <data type>) RETURNS STRING returns
a string representation of a numeric value Note: <data
type> may be REAL or INTEGER

Example: NUM_TO_STR(87.5) returns "87.5"

STR_TO_NUM(x : <data type1>) RETURNS <data type2> returns
a numeric representation of a string Note: <data type1> may be
CHAR or STRING

Note: <data type2> may be REAL or INTEGER

Example: STR_TO_NUM("23.45") returns 23.45

IS_NUM(ThisString : STRING) RETURNS BOOLEAN
returns the value TRUE if ThisString represents a valid numeric value

Note: <data type> may be CHAR or STRING

Example: IS_NUM("12.36") returns TRUE

Example: IS_NUM("-12.36") returns TRUE

Example: IS_NUM("12.3a") returns FALSE

ASC(ThisChar : CHAR) RETURNS INTEGER returns an
integer value (the ASCII value) of ThisChar

Example: ASC('A') returns 65

CHR(x : INTEGER) RETURNS CHAR
returns the character whose integer value (the ASCII value) is

x Example: CHR(87) returns 'W'

NUMERIC Functions

INT(x : REAL) RETURNS INTEGER
returns the integer part of x

Example: INT(27.5415) returns 27

RAND(x : INTEGER) RETURNS REAL
returns a real number in the range 0 to x (**not** inclusive of x)

Example: RAND(87) could return 35.43





DATE Functions

Note: Date format is assumed to be DDMMYYYY unless otherwise stated.

DAY(ThisDate : DATE) RETURNS INTEGER
returns the current day number from ThisDate

Example: DAY(4/10/2003) returns 4

MONTH(ThisDate : DATE) RETURNS INTEGER
returns the current month number from ThisDate

Example: MONTH(4/10/2003) returns 10

YEAR(ThisDate : DATE) RETURNS INTEGER
returns the current year number from ThisDate

Example: YEAR(4/10/2003) returns 2003

DAYINDEX(ThisDate : DATE) RETURNS INTEGER
returns the current day index number from ThisDate where Sunday = 1, Monday = 2, Tuesday = 3 etc.

Example: DAYINDEX(12/05/2020) returns 3

SETDATE(Day, Month, Year : INTEGER) RETURNS DATE returns
a variable of type DATE

NOW() RETURNS DATE
returns the current date

OTHER Functions

EOF(FileName : STRING) RETURNS BOOLEAN
returns TRUE if there are no more lines to be read from file FileName

Note: This function will generate an ERROR if the file is not already open in READ mode



OPERATORS

&	Concatenates (joins) two strings Example: "Summer" & " " & "Pudding" evaluates to "Summer Pudding" Note: This operator may also be used to concatenate a character with a string
AND	Performs a logical AND on two Boolean values Example: TRUE AND FALSE evaluates to FALSE
OR	Performs a logical OR on two Boolean values Example: TRUE OR FALSE evaluates to TRUE
NOT	Performs a logical NOT on a Boolean value Example: NOT TRUE evaluates to FALSE
MOD	Finds the remainder when one number is divided by another Example: 10 MOD 3 evaluates to 1
DIV	Finds the quotient when one number is divided by another Example 10 DIV 3 evaluates to 3

Programming environments provide many built-in functions. Some of them are always available to use; some need to be imported from specialist module libraries.

Built-in functions in Visual Basics

Built-in functions - Arithmetic functions: round, truncation.

Example Program - Built-in functions - Arithmetic functions: round, truncation.

```
Module module1
Sub main()
```

```
Dim num As Double
Dim rounded As Integer
Dim squareroot As Double
Dim trunc As Integer
```

```
Console.WriteLine("Enter a real number")
```

```
num = Console.ReadLine()
rounded = Math.Round(num)
squareroot = Math.Sqrt(num)
```

```
Console.WriteLine("round: " & rounded & vbNewLine & "Square Root: " & squareroot)
```

```
trunc = Math.Truncate(num)
```

```
Console.WriteLine("The number truncated is " & trunc)
Console.WriteLine("This is not always the same as rounded")
Console.ReadKey()
```

```
End Sub
End Module
```



Example Program - String handling functions *length, position, substring, concatenation.*

Module module1

Sub main()

Dim theString As String

theString = "Hello Dave, you're my friend now!"

Console.WriteLine(theString)

Console.WriteLine(theString.Length) 'display the string's length

Console.WriteLine(theString.ToUpper) 'display the string in upper case

Console.WriteLine(theString.ToLower) 'display the string in lower case

Console.WriteLine(theString.Contains("Dave")) 'is Dave there?

Console.WriteLine(theString.IndexOf("D")) 'position of D

Console.WriteLine(theString.Substring(12)) 'displays the substring starting at position 12

Dim newString As String

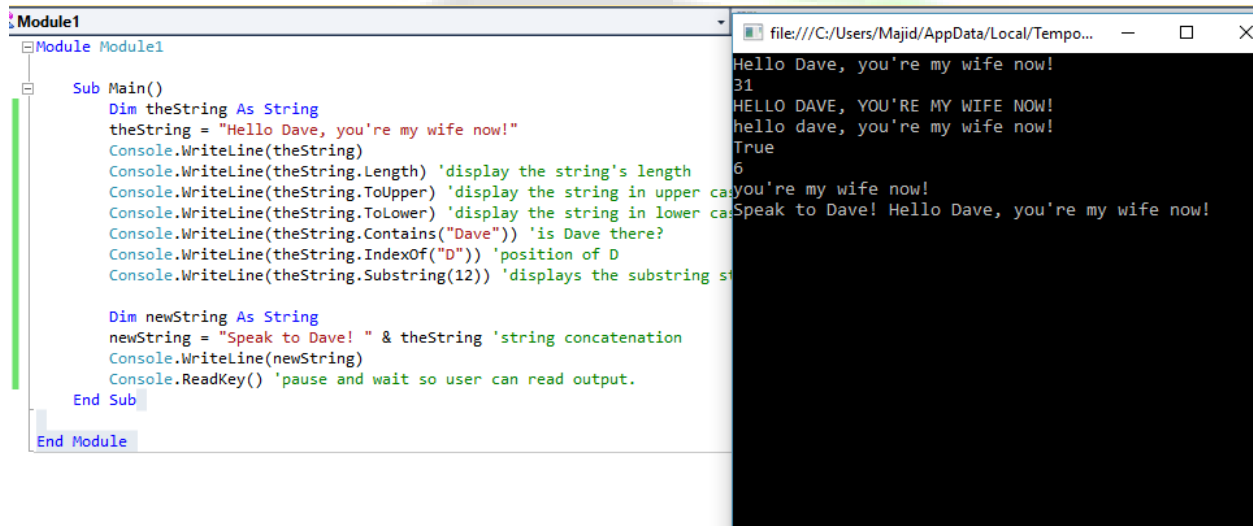
newString = "Speak to Dave! " & theString 'string concatenation

Console.WriteLine(newString)

Console.ReadKey() 'pause and wait so user can read output.

End Sub

End Module



```

Module1
  Module Module1
    Sub Main()
      Dim theString As String
      theString = "Hello Dave, you're my wife now!"
      Console.WriteLine(theString)
      Console.WriteLine(theString.Length) 'display the string's length
      Console.WriteLine(theString.ToUpper) 'display the string in upper case
      Console.WriteLine(theString.ToLower) 'display the string in lower case
      Console.WriteLine(theString.Contains("Dave")) 'is Dave there?
      Console.WriteLine(theString.IndexOf("D")) 'position of D
      Console.WriteLine(theString.Substring(12)) 'displays the substring starting at position 12

      Dim newString As String
      newString = "Speak to Dave! " & theString 'string concatenation
      Console.WriteLine(newString)
      Console.ReadKey() 'pause and wait so user can read output.
    End Sub
  End Module

```

```

file:///C:/Users/Majid/AppData/Local/Tempo...
Hello Dave, you're my wife now!
31
HELLO DAVE, YOU'RE MY WIFE NOW!
hello dave, you're my wife now!
True
6
you're my wife now!
Speak to Dave! Hello Dave, you're my wife now!

```

Example Program - String conversion functions *to/from integer, real, date/time.*

Module module1

Sub main()

Dim theInt, theReal, theDate As String

theInt = "23021980"

theReal = "230.21980"

theDate = "23-02-1980"

'whole numbers

Console.WriteLine(theInt)

Console.WriteLine(theInt + "1")

Console.WriteLine(Convert.ToInt32(theInt))

Console.WriteLine((Convert.ToInt32(theInt) + 1))

Console.WriteLine()

'real numbers

Console.WriteLine(theReal)

Console.WriteLine(theReal + "1")

Console.WriteLine(Convert.ToDouble(theReal))

Console.WriteLine(Convert.ToDouble(theReal) + 1)

Console.WriteLine()

'dates

Console.WriteLine(theDate)

Console.WriteLine(theDate + "1")

Console.WriteLine(DateTime.Parse(theDate))

Console.WriteLine(DateTime.Parse(theDate).AddDays(1))

Console.ReadKey() 'pause and wait so user can read output.

End Sub

End Module



Summary of VB Functions

Date/Time Functions

Function	Description
CDate	Converts a valid date and time expression to the variant of subtype Date
Date	Returns the current system date
DateAdd	Returns a date to which a specified time interval has been added
DateDiff	Returns the number of intervals between two dates
DatePart	Returns the specified part of a given date
DateSerial	Returns the date for a specified year, month, and day
DateValue	Returns a date
Day	Returns a number that represents the day of the month (between 1 and 31, inclusive)
FormatDateTime	Returns an expression formatted as a date or time
Hour	Returns a number that represents the hour of the day (between 0 and 23, inclusive)
IsDate	Returns a Boolean value that indicates if the evaluated expression can be converted to a date
Minute	Returns a number that represents the minute of the hour (between 0 and 59, inclusive)
Month	Returns a number that represents the month of the year (between 1 and 12, inclusive)
MonthName	Returns the name of a specified month
Now	Returns the current system date and time
Second	Returns a number that represents the second of the minute (between 0 and 59, inclusive)
Time	Returns the current system time
Timer	Returns the number of seconds since 12:00 AM
TimeSerial	Returns the time for a specific hour, minute, and second
TimeValue	Returns a time
Weekday	Returns a number that represents the day of the week (between 1 and 7, inclusive)
WeekdayName	Returns the weekday name of a specified day of the week
Year	Returns a number that represents the year



Conversion Functions

Function	Description
Asc	Converts the first letter in a string to ANSI code
CBool	Converts an expression to a variant of subtype Boolean
CByte	Converts an expression to a variant of subtype Byte
CCur	Converts an expression to a variant of subtype Currency
CDate	Converts a valid date and time expression to the variant of subtype Date
CDbl	Converts an expression to a variant of subtype Double
Chr	Converts the specified ANSI code to a character
CInt	Converts an expression to a variant of subtype Integer
CLng	Converts an expression to a variant of subtype Long
CSng	Converts an expression to a variant of subtype Single
CStr	Converts an expression to a variant of subtype String
Hex	Returns the hexadecimal value of a specified number
Oct	Returns the octal value of a specified number

Format Functions

Function	Description
FormatCurrency	Returns an expression formatted as a currency value
FormatDateTime	Returns an expression formatted as a date or time
FormatNumber	Returns an expression formatted as a number
FormatPercent	Returns an expression formatted as a percentage

Math Functions

Function	Description
Abs	Returns the absolute value of a specified number
Atn	Returns the arctangent of a specified number
Cos	Returns the cosine of a specified number (angle)
Exp	Returns e raised to a power
Hex	Returns the hexadecimal value of a specified number
Int	Returns the integer part of a specified number
Fix	Returns the integer part of a specified number
Log	Returns the natural logarithm of a specified number
Oct	Returns the octal value of a specified number
Rnd	Returns a random number less than 1 but greater or equal to 0
Sgn	Returns an integer that indicates the sign of a specified number
Sin	Returns the sine of a specified number (angle)
Sqr	Returns the square root of a specified number
Tan	Returns the tangent of a specified number (angle)

References:

-  Cambridge International AS & A level Computer Science Course book by Sylvia Langfield and Dave Duddell
-  Visual Basics Console Cook Book
-  VB.NET **AQA** Console Book by *Dough Simple*