**Introduction to the VB.NET Programming Language**

Contents

[Subroutines, Inputs, Outputs, and Variables 4](#_Toc421707950)

[Breaking code down into subroutines: 4](#_Toc421707951)

[Outputting to the screen (Console.WriteLine) 4](#_Toc421707952)

[Changing text colour 4](#_Toc421707953)

[Storing data using variables (Dim) 4](#_Toc421707954)

[Getting Data from the user and storing it (Console.ReadLine) 4](#_Toc421707955)

[Strings 5](#_Toc421707956)

[Counting the number of letters in a word 5](#_Toc421707957)

[Getting specific letter from the middle of a word (Mid) 5](#_Toc421707958)

[Maths 6](#_Toc421707959)

[Doing some maths 6](#_Toc421707960)

[Doing maths directly with variables 6](#_Toc421707961)

[Using random numbers (Rnd) 6](#_Toc421707962)

[Formatting Numbers to Currency, Percentages, decimals etc. 6](#_Toc421707963)

[Using maths functions e.g. Rounding (Math) 6](#_Toc421707964)

[Conditions 7](#_Toc421707965)

[Using conditions ('if' statements) 7](#_Toc421707966)

[Using multiple conditions ('if', 'elseif' and 'else' statements) 7](#_Toc421707967)

[Using conditions to check lots of possibilities (Select Case) 7](#_Toc421707968)

[Loops 8](#_Toc421707969)

[Repeating something as set number of times ('for' loops) 8](#_Toc421707970)

[Repeating and counting where you're up to ('for' loops) 8](#_Toc421707971)

[Repeating something while counting backwards ('for' loops) 8](#_Toc421707972)

[Repeating something until you want to stop ('Do' loops) 8](#_Toc421707973)

[Repeating something until you want to stop ('While' loops) 8](#_Toc421707974)

[Files 9](#_Toc421707975)

[Saving data in files (StreamWriter) 9](#_Toc421707976)

[Loading data from a file and displaying it on screen (StreamReader) 9](#_Toc421707977)

[Loading data from a file and storing it (StreamReader) 9](#_Toc421707978)

[Checking if a file exists 9](#_Toc421707979)

[Data Structures: Lists 10](#_Toc421707980)

[Storing multiple variables together (Lists) 10](#_Toc421707981)

[Adding and removing Data from a list (List.Add, List.Remove, List.Clear) 10](#_Toc421707982)

[Sorting a List into Alphabetical or Numeric order (Sort) 11](#_Toc421707983)

[Reversing the order of a List (Reverse) 11](#_Toc421707984)

[Randomising a List (OrderBy) 11](#_Toc421707985)

[Data Structures: Arrays 12](#_Toc421707986)

[Storing a known quantity of multiple variables (Arrays) 12](#_Toc421707987)

[Sorting an array into alphabetical or numerical order (Arrays) 12](#_Toc421707988)

[Randomizing the items in an Array (OrderBy) 12](#_Toc421707989)

[Data Structures: Classes 13](#_Toc421707990)

[Grouping lots of related variables together (Classes) 13](#_Toc421707991)

[Using a contstructor to make a new object from a class (public sub new) 13](#_Toc421707992)

[Time and Date 14](#_Toc421707993)

[Pausing the program (Sleep) 14](#_Toc421707994)

[Using Dates and Times 14](#_Toc421707995)

[Timing Events (Now) 14](#_Toc421707996)

[Handing Errorrs 15](#_Toc421707997)

[Stopping a program from crashing (Try/Catch) 15](#_Toc421707998)

[Stopping a program from crashing and retrying something (Try/Catch) 15](#_Toc421707999)

[Stopping specific types of errors from occuring (Try/Catch) 15](#_Toc421708000)

[Miscellaneous 16](#_Toc421708001)

[Picking a random task to do (Rnd) 16](#_Toc421708002)

[Changing the console size and title and moving the console cursor 16](#_Toc421708003)

[Doing something every 'n'th time in a loop (Mod) 16](#_Toc421708004)

[Using different ASCII characters (Chr) 17](#_Toc421708005)

[Finding the ASCII value of a character 17](#_Toc421708006)

[Checking for a specific Key Being Pressed (consoleKeyInfo) 17](#_Toc421708007)

[Performing a specific action when a key is pressed 17](#_Toc421708008)

[Subroutines and Functions 18](#_Toc421708009)

[Moving mariables between subroutines (byVal) 18](#_Toc421708010)

[Writing a function and using return values 18](#_Toc421708011)

[Using functions as conditions 19](#_Toc421708012)

# Subroutines, Inputs, Outputs, and Variables

## Breaking code down into subroutines:

Module Module1

Sub Main () 'a ‘sub’ (sub-routine) is a block of code

theFirstThing() 'this goes to another sub and comes back here when finished

theSecondThing() 'this goes to a different sub

Console.ReadKey() 'wait for key press

End Sub

Sub theFirstThing()

Console.WriteLine("This is the first thing")

End Sub

Sub theSecondThing()

Console.WriteLine("This is the second thing")

End Sub

End Module

## Outputting to the screen (Console.WriteLine)

'Anything after an apostrophe is a comment (not code)

Console.WriteLine("Hello")

'use an ampersand (&) to join strings and other variables

Console.WriteLine("1 + 2 = " & 1 + 2)

Console.ReadKey()

## Changing text colour

'text colour

Console.ForegroundColor = ConsoleColor.White

'behind the text colour

Console.BackgroundColor = ConsoleColor.Red

Console.WriteLine("Hello!")

Console.ReadKey()

## Storing data using variables (Dim)

'strings, integers, decimals, and Booleans

'can only be used in the sub where they’re created

Dim name As String = "Juan"

Dim age As Integer = 17

Dim height As Decimal = 1.82

Dim britishCitizen As Boolean = True

Console.ReadKey()

## Getting Data from the user and storing it (Console.ReadLine)

Dim name As String

Console.WriteLine("What is your name?")

'read in what the user types and store it in the 'name' variable

name = Console.ReadLine

Console.WriteLine("Hello " & name & "!")

Console.ReadKey()

# Strings

## Counting the number of letters in a word

Dim word As String = "Potato"

Console.WriteLine("The word {0} has {1} letters.", word, word.Length)

Console.ReadKey()

**Converting to upper- or lower-case (toupper, tolower)**

Dim name As String

Console.WriteLine("What is your name?")

name = Console.ReadLine

'convert the string to lower case

Console.WriteLine("{0}?", name.ToLower)

'convert the string to upper case

Console.WriteLine("{0}!", name.ToUpper)

Console.ReadKey()

**Getting specific letters from the start or end of a word (Left, Right)**

Dim name As String

Console.WriteLine("What is your name?")

name = Console.ReadLine

'the left commands get a number of characters from the left of a string

Console.WriteLine("Your name begins with {0}.", Left(name, 2))

'the right commands get a number of characters from the right of a string

Console.WriteLine("Your name ends with a {0}.", Right(name, 1))

Console.ReadKey()

## Getting specific letter from the middle of a word (Mid)

Dim word As String = "Potato"

'starting at letter #1, get 1 letter from the word

Console.WriteLine(Mid(word, 1, 1))

'starting at letter #2, get 2 letters from the word

Console.WriteLine(Mid(word, 2, 2))

'starting at letter #4, get 3 letter from the word

Console.WriteLine(Mid(word, 4, 3))

Console.ReadKey()

# Maths

## Doing some maths

'make two variables

Dim a As Integer = 7

Dim b As Integer = 4

'calculate the answer

Dim answer As Integer = a + b

'put variables and strings together

Console.WriteLine(a & " + " & b & " = " & answer)

Console.ReadKey()

## Doing maths directly with variables

'make two variables

Dim a As Integer = 7

Dim b As Integer = 4

Dim total As Integer = 0

'add the value of a to the total (total will be: 0 + 7 = 7)

total += a

'subtract the value of b from the total (total will be: 7 - 4 = 3)

total -= b

'multiply the total by itself (total will be 3 \* 3 = 9)

total \*= total

Console.WriteLine("{0} - {1} times itself is {2}", a, b, total)

Console.ReadKey()

## Using random numbers (Rnd)

Dim x As Integer

Randomize()

'Rnd() gives a random number from 0.000 to 0.999

'multiply by something to make it bigger (e.g. by 100 to get 0 to 99)

'add one to make it one bigger e.g. (1 to 100)

x = Rnd() \* 100 + 1

Console.ReadKey()

## Formatting Numbers to Currency, Percentages, decimals etc.

Dim x As Decimal = 1.25

Console.WriteLine("Number is {0}", x)

Console.WriteLine("To one decimal place: {0:f1}", x)

Console.WriteLine("As currency: {0:c2}", x)

Console.WriteLine("As a percentage: {0:p0}", x)

Console.ReadKey()

## Using maths functions e.g. Rounding (Math)

Dim a, b As Integer

a = 3

b = 6

'find an average (mean) and round it

Console.WriteLine(Math.Round((a + b) / 2))

'find the highest (also works with Math.Min)

Console.WriteLine(Math.Max(a, b))

Console.ReadKey()

# Conditions

## Using conditions ('if' statements)

Dim answer As String

Console.WriteLine("What is the capital of Sweden?")

answer = Console.ReadLine

If answer = "Stockholm" Then

Console.WriteLine("Correct!")

Else

Console.WriteLine("Incorrect")

End If

Console.ReadKey()

## Using multiple conditions ('if', 'elseif' and 'else' statements)

Dim answer As String

Console.WriteLine("What year was the battle of Hastings?")

answer = Console.ReadLine

'check if the condition is true and do something as a result

If answer = "1066" Then

Console.WriteLine("Correct!")

'If the condition above was not true, check this one instead

ElseIf answer = "1065" Or answer = "1067" Then

Console.WriteLine("Your were one year out!")

'if none of the conditions above were true, do this

Else

Console.WriteLine("Incorrect")

End If

Console.ReadKey()

## Using conditions to check lots of possibilities (Select Case)

Dim age As Integer

Console.WriteLine("How old are you in years?")

age = Console.ReadLine

'give the name of the variable you're checking

Select Case age

'List all the possible options and what to do in each case

Case Is < 0

Console.WriteLine("That's not possible!")

Case 1 To 4

Console.WriteLine("You're just a baby!")

Case 5 To 10

Console.WriteLine("You're probably in primary school.")

Case 11 To 15

Console.WriteLine("Are you doing your GCSEs yet?")

Case 16, 17, 18

Console.WriteLine("Have you gone to college?")

Case 19 To 21

Console.WriteLine("Are you at University?")

Case Is >= 65

Console.WriteLine("You're retired!")

Case Else

Console.WriteLine("You probably have a job and a family by now!")

End Select

Console.ReadKey()

# Loops

## Repeating something as set number of times ('for' loops)

'loop ten times

For counter = 1 To 10

Console.WriteLine("Hello!")

Next

## Repeating and counting where you're up to ('for' loops)

'using the 'counter' variable

For counter = 1 To 4

Console.WriteLine(counter & " potato")

Next

## Repeating something while counting backwards ('for' loops)

'counting backwards

For counter = 5 To 0 Step -1

Console.WriteLine(counter & "...")

Next

Console.ReadKey()

## Repeating something until you want to stop ('Do' loops)

Dim answer As String

This loop will always happen at least once

Do

Console.WriteLine("Type 'exit' to exit.")

answer = Console.ReadLine

'check if we should do this again

Loop Until answer = "exit"

Console.WriteLine("Exiting...")

Console.ReadKey()

## Repeating something until you want to stop ('While' loops)

Dim money As Decimal

Dim spend As Decimal

Console.WriteLine("How much money do you have?")

money = Console.ReadLine

'this loop might not happen at all (if we start with 0 money or less)

While money > 0

Console.WriteLine("How much do you want to spend?")

spend = Console.ReadLine

money -= spend

Console.WriteLine("You now have {0:c2}.", money)

End While

Console.WriteLine("You ran out of money!")

Console.ReadKey()

# Files

## Saving data in files (StreamWriter)

'this will make a file called 'myTextFile'

'stored in your home documents folder in school (U:)

'the 'true' setting adds an extra line to the file each time

'change 'true' to 'false' to completely overwrite the file each time

Dim file As New IO.StreamWriter("U:\myTextFile.txt", True)

'write this text to the file

file.WriteLine("Hello!")

'always close a file once you're finished with it

file.Close()

Console.ReadKey()

## Loading data from a file and displaying it on screen (StreamReader)

'declare a file to use

Dim file As New IO.StreamReader("U:\myTextFile.txt")

'keep going until the end of the file

While Not file.EndOfStream

'read line from file, write it to screen

Console.WriteLine(file.ReadLine)

End While

Console.ReadKey()

## Loading data from a file and storing it (StreamReader)

'make a list of strings to dtore data from the file

Dim myList As New List(Of String)

Dim file As New IO.StreamReader("U:\myTextFile.txt")

'keep going until the end of the file

While file.EndOfStream = False

'read line from file, add it to the list

myList.Add(file.ReadLine)

End While

File.Close

Console.ReadKey()

## Checking if a file exists

Dim fileName As String = "U:\myTextFile.txt"

'check if a file exists before you load it

'otherwise your program might crash

If IO.File.Exists(fileName) Then

Console.WriteLine("Found the file {0}", fileName)

Else

Console.WriteLine("{0} does not exist.", fileName)

End If

Console.ReadKey()

# Data Structures: Lists

## Storing multiple variables together (Lists)

'this makes a list. note the 'new' keyword here

Dim myList As New List(Of String)

'things in lists are numbered, starting at zero

myList.Add("Larry") 'item 0

myList.Add("Moe") 'item 1

myList.Add("Curly") 'item 2

'we can use a loop to look at them all

For counter = 0 To 2

'this outputs item 'i' from the list

Console.WriteLine(myList(counter))

Next

Console.ReadKey()

## Adding and removing Data from a list (List.Add, List.Remove, List.Clear)

'Create a list

Dim dinosaurs As New List(Of String)

'Add some items to it

dinosaurs.Add("Tyrannosaurus")

dinosaurs.Add("Amargasaurus")

dinosaurs.Add("Mamenchisaurus")

dinosaurs.Add("Deinonychus")

dinosaurs.Add("Compsognathus")

'Output all items in the list using a loop

For counter = 0 To dinosaurs.Count - 1

Console.WriteLine(dinosaurs(counter))

Next

'Output the number of items in the list

Console.WriteLine("Count: {0}", dinosaurs.Count)

'Output an item at a specific position

Console.WriteLine("dinosaurs(3) is: {0}", dinosaurs(3))

'Remove the specified item from the list

dinosaurs.Remove("Compsognathus")

Console.WriteLine("Removed 'Compsognathus'")

'Remove the item at position 0 in the list

dinosaurs.RemoveAt(0)

Console.WriteLine("Removed dinosaurs(0)")

'Clear all items from the list

dinosaurs.Clear()

Console.WriteLine("Count is now: {0}", dinosaurs.Count)

Console.ReadKey()

## Sorting a List into Alphabetical or Numeric order (Sort)

'Create a list

Dim dinosaurs As New List(Of String)

'Add items

dinosaurs.Add("Pachycephalosaurus")

dinosaurs.Add("Amargasaurus")

dinosaurs.Add("Mamenchisaurus")

dinosaurs.Add("Deinonychus")

'Sort the list

dinosaurs.Sort()

Console.WriteLine(" Sorted List:")

'Output each item using a loop to show the sorted list

For counter = 0 To dinosaurs.Count - 1

Console.WriteLine(dinosaurs(counter))

Next

Console.ReadKey()

## Reversing the order of a List (Reverse)

'Create a list

Dim dinosaurs As New List(Of String)

'Add items

dinosaurs.Add("Pachycephalosaurus")

dinosaurs.Add("Amargasaurus")

dinosaurs.Add("Mamenchisaurus")

dinosaurs.Add("Deinonychus")

'reverse the order of the list

dinosaurs.Reverse()

Console.WriteLine(" Reversed list:")

'Output each item using a loop to show the reverse-ordered list

For counter = 0 To dinosaurs.Count - 1

Console.WriteLine(dinosaurs(counter))

Next

Console.ReadKey()

## Randomising a List (OrderBy)

Randomize()

'Create a list

Dim dinosaurs As New List(Of String)

'Add items

dinosaurs.Add("Pachycephalosaurus")

dinosaurs.Add("Amargasaurus")

dinosaurs.Add("Mamenchisaurus")

dinosaurs.Add("Deinonychus")

'randomize the list

dinosaurs = dinosaurs.OrderBy(Function() Rnd()).ToList

Console.WriteLine(" Randomized list:")

'Output each item using a loop to show the randomized list

For counter = 0 To dinosaurs.Count - 1

Console.WriteLine(dinosaurs(counter))

Next

Console.ReadKey()

# Data Structures: Arrays

## Storing a known quantity of multiple variables (Arrays)

Dim myArray(5) As String

'things in arrays are numbered, starting at zero

myArray(0) = "Larry" 'item 0

myArray(1) = "Moe" 'item 1

myArray(2) = "Curly" 'item 2

'we can use a loop to look at them all

For counter = 0 To 2

'this outputs item 'i' from the list

Console.WriteLine(myArray(counter))

Next

Console.ReadKey()

## Sorting an array into alphabetical or numerical order (Arrays)

'Create a list

Dim dinosaurs(3) As String

'Add items

dinosaurs(0) = "Pachycephalosaurus"

dinosaurs(1) = "Amargasaurus"

dinosaurs(2) = "Mamenchisaurus"

dinosaurs(3) = "Deinonychus"

'Sort the list into order, then reverse it

Array.Sort(dinosaurs)

Array.Reverse(dinosaurs)

Console.WriteLine("Sorted and reversed list:")

'Output each item using a loop to show the list

For counter = 0 To dinosaurs.Count - 1

Console.WriteLine(dinosaurs(counter))

Next

Console.ReadKey()

## Randomizing the items in an Array (OrderBy)

Randomize()

'Create a list

Dim dinosaurs(3) As String

'Add items

dinosaurs(0) = "Pachycephalosaurus"

dinosaurs(1) = "Amargasaurus"

dinosaurs(2) = "Mamenchisaurus"

dinosaurs(3) = "Deinonychus"

'Sort the list into order, then reverse it

dinosaurs = dinosaurs.OrderBy(Function() Rnd()).ToArray

Console.WriteLine("Randomized list:")

'Output each item using a loop to show the list

For counter = 0 To dinosaurs.Count - 1

Console.WriteLine(dinosaurs(counter))

Next

Console.ReadKey()

# Data Structures: Classes

## Grouping lots of related variables together (Classes)

Module Module1

'a class is like a variable with multiple parts

'it is declared OUTSIDE OF Sub Main()

Public Class person

Public forename As String

Public surname As String

Public age As Integer

End Class

Sub Main ()

'create a person

Dim person1 As New person

'set their details

person1.forename = "Mohammad"

person1.surname = "Ali"

person1.age = 73

'even make a list

Dim boxers As New List(Of person)

'add our existing person

boxers.Add(person1)

'output details from the list

For Each p As person In boxers

Console.WriteLine(p.forename & " " & p.surname & " is " & p.age)

Next

Console.ReadKey()

End Sub

End Module

## Using a contstructor to make a new object from a class (public sub new)

Module Module1

Public Class person

Public forename As String

Public surname As String

Public DOB As DateTime

'this sets the variables above to the data sent when creating a new person

Public Sub New(ByVal f As String, ByVal s As String, ByVal d As DateTime)

forename = f

surname = s

DOB = d

End Sub

End Class

Sub Main()

'create a person AND set their details, then add them to the list

Dim person1 As New person("John", "Cleese", "27-10-1939")

Dim montyPython As New List(Of person)

montyPython.Add(person1)

'add directly to a list

montyPython.Add(New person("Eric", "Idle", "29-03-1943"))

montyPython.Add(New person("Michael", "Palin", "05-05-1943"))

For Each p As person In montyPython

Console.WriteLine("{0} {1}: {2:yyyy}", p.forename, p.surname, p.DOB)

Next

Console.ReadKey()

End Sub

End Module

# Time and Date

## Pausing the program (Sleep)

Console.WriteLine("Wait for it...")

'the sleep command pauses the program for a number of milliseconds

'1000 milliseconds = 1 second

Threading.Thread.Sleep(3000)

Console.WriteLine("Finished now!")

Console.ReadKey()

## Using Dates and Times

Sub Main()

'use the "DateTime" data type for storing dates and/or times

Dim currentTime As DateTime

'use the "Now()" function to get the current date/time

currentTime = Now()

'use some properties of the variable to output parts of the date or time

Console.WriteLine("It's {0}", currentTime.Year)

Console.WriteLine("This is month number {0}", currentTime.Month)

Console.WriteLine("The time is {0}", currentTime.ToShortTimeString)

'use formatting to do similar things. Often easier to do

'the codes are: y for year, M for month, d for day

'h for hour (12hr), H for hour (24hr), m for minute, s for second

'f for fractions of a second, t for AM/PM, K for time zone, g for AD/BC

'use between one and four of those letters to change the format

Console.WriteLine("It's {0:yyyy gg}", currentTime)

Console.WriteLine("Another {0:dddd} in {0:MMMM}", currentTime)

Console.WriteLine("The time is {0:HH:mm tt}", currentTime)

Console.ReadKey()

## Timing Events (Now)

Dim startTime, endTime As DateTime

'record the start time. The now() function gets the current date/time

startTime = Now

Console.WriteLine("Program started at {0}", startTime.TimeOfDay)

Console.WriteLine("Press any key to stop timing.")

Console.ReadKey()

'record the end time

endTime = Now

Console.WriteLine("That took {0} seconds", (endTime - startTime).TotalSeconds)

Console.ReadKey()

# Handing Errorrs

## Stopping a program from crashing (Try/Catch)

Dim age As Integer

Console.WriteLine("Enter your age:")

Try

'here, someone might type in a string

'a string can't be stored in an integer variable

'if the program tries to do that, it will begin to crash

age = Console.ReadLine

Catch ex As Exception

'this part will only happen if the program is crashing

'an "exception" is an error the program encounters

Console.WriteLine("That's not a number!")

End Try

Console.ReadKey()

## Stopping a program from crashing and retrying something (Try/Catch)

Dim age As Integer

Dim validInput As Boolean = False

Do

Console.WriteLine("Enter your age:")

Try

age = Console.ReadLine

'if the previous line caused the program to crash…

'the program will jump to the "catch" and the line below won't happen

validInput = True

Catch ex As Exception

Console.WriteLine("That's not a number!")

End Try

'we can't continue unless a valid value was entered

Loop Until validInput = True

Console.ReadKey()

## Stopping specific types of errors from occuring (Try/Catch)

'the 'byte' data type if for whole nubmers from 0 to 255

Dim age As Byte

Dim validInput As Boolean = False

Console.WriteLine("Enter your age:")

Try

age = Console.ReadLine

Catch ex As OverflowException

'overflow is when a number is out of range for its data type

'e.g. any number < 0 or > 255 in a byte

Console.WriteLine("Age must be between 0 and 255 years!")

Catch ex As InvalidCastException

'invalid cast is where we trying to store the wrong type of data

'e.g. trying to store a string in an integer

Console.WriteLine("That's not a number!")

Catch ex As Exception

'used for any other types of error

Console.WriteLine("The number you entered was not valid.")

End Try

Console.ReadKey()

# Miscellaneous

## Picking a random task to do (Rnd)

Dim x As Integer

Randomize()

'pick a number

x = Rnd() \* 4

'check the random number;

'do different things depending on its value

If x = 0 Then

Console.WriteLine("Alpha")

ElseIf x = 1 Then

Console.WriteLine("Bravo")

ElseIf x = 2 Then

Console.WriteLine("Delta")

Else

Console.WriteLine("Charlie")

End If

Console.ReadKey()

## Changing the console size and title and moving the console cursor

Do

Console.Clear()

Console.Title = "This is the window title"

Console.CursorVisible = False

Console.WindowWidth = 29

Console.WindowHeight = 9

Console.CursorLeft = 7

Console.CursorTop = 4

If Console.CapsLock = True Then

Console.Write("CAPS LOCK IS ON")

Else

Console.Write("caps lock is off")

End If

Threading.Thread.Sleep(100)

Loop

## Doing something every 'n'th time in a loop (Mod)

'a simple loop that happens thirty times

For counter = 1 To 30

'write the value of the counter (formatted as two digits, e.g. 01, 02)

'note that this is a console.write, not a console.writeline

Console.Write("{0:00} ", counter)

'check if the counter is divisible for 5

If counter Mod 5 = 0 Then

'if it is, start a new line

Console.WriteLine()

End If

Next

Console.ReadKey()

## Using different ASCII characters (Chr)

Dim x As Integer

Console.WriteLine("Enter a number from 0 to 255")

x = Console.ReadLine

Console.WriteLine("The ASCII character for {0} is '{1}'.", x, Chr(x))

Console.ReadKey()

## Finding the ASCII value of a character

Dim c As String

Console.WriteLine("Enter a single character from the keyboard")

c = Console.ReadLine

Console.WriteLine("The ASCII value for '{0}' is {1}.", c, Asc(c))

Console.ReadKey()

## Checking for a specific Key Being Pressed (consoleKeyInfo)

'this variable type records information about which button was pressed

Dim pressed As ConsoleKeyInfo

Do

Console.WriteLine("Press any key! (Escape to exit)")

'the 'true' here means the letter won't be shown on the screen

pressed = Console.ReadKey(True)

Console.WriteLine("You pressed {0}", pressed.Key.ToString)

Loop Until pressed.Key = ConsoleKey.Escape

Console.ReadKey()

## Performing a specific action when a key is pressed

Dim pressed As ConsoleKeyInfo

Console.CursorVisible = False

Console.WriteLine("Use the arrows!")

Console.Write(Chr(1))

pressed = Console.ReadKey(True)

Select Case pressed.Key

Case ConsoleKey.RightArrow

Console.CursorLeft += 1

Case ConsoleKey.DownArrow

Console.CursorTop += 1

End Select

Console.Write(Chr(1))

Console.ReadKey()

# Subroutines and Functions

## Moving mariables between subroutines (byVal)

Module Module1

Sub Main()

Dim forename, surname As String

Console.WriteLine("Enter your forename:")

forename = Console.ReadLine

Console.WriteLine("Enter your surname:")

surname = Console.ReadLine

'to go to a subroutines we type its name

'in the brackets, we give any varaibales to be passed

output(forename, surname)

Console.ReadKey()

End Sub

'in the brackts here, we give details of the incomine values/variables

Sub output(ByVal f As String, ByVal s As String)

Console.WriteLine("Hello {0} {1}!", f, s)

End Sub

End Module

## Writing a function and using return values

Module Module1

Sub Main()

Dim r As Decimal

Console.WriteLine("Enter a radius:")

r = Console.ReadLine

'to go to a subroutines we type its name

'in the brackets, we give any varaibales to be passed

Console.WriteLine(area(r))

Console.WriteLine(circumference(r))

Console.ReadKey()

End Sub

'in the brackts here, we give details of the incomine values/variables

'after the brackets, we state what type of data will be returned

Function area(ByVal r As Decimal) As Decimal

'return sends an answer back to where the function was called from

Return Math.PI \* r ^ 2

End Function

Function circumference(ByVal r As Decimal) As Decimal

Return 2 \* Math.PI \* r

End Function

End Module

## Using functions as conditions

Module Module1

Sub Main()

Dim myList As New List(Of String)

myList.Clear()

myList.AddRange({"Alpha", "Beta", "Gamma", "Delta"})

Do

Console.WriteLine("Enter a Greek letter:")

'in this condition we check if "validchoice" returns as true

'we give the function our list and an input from the user

Loop Until validchoice(myList, Console.ReadLine) = True

Console.WriteLine("Exiting...")

Console.ReadKey()

End Sub

'in the brackts here, we give details of the incomine values/variables

'after the brackets, we state what type of data will be returned

Function validChoice(ByVal l As List(Of String), ByVal input As String) As Boolean

If l.Contains(input) Then

'when a return happens, the function ends and sends its answer back

Console.WriteLine("Valid choice!")

Return True

Else

Console.WriteLine("Invalid choice.")

Return False

End If

End Function

End Module