

```
D:\work\server\server\server\bin\server.exe
Foreground : green, background : darkblue
Foreground : yellow, background : darkgreen
Choose one of 16 colors for your console text !
Aqua blue grey purple pink ... and more ! _
```

VB.Net Console Student Booklet

by Doug Semple

Tasks and sample programs to teach the AQA GCE Computing COMP1 syllabus.

Contents

Download and Setup.....	3
Your first program.....	3
Example Program 1 – Sequence	4
Example Program 2 - Assignment.....	5
Example Program 3 - Arithmetic.....	6
Example Program 4 – Selection	7
Example Program 5 - Relational operators.....	8
Example Program 6 - Boolean operators.....	9
Moderation Exercise 1 – Central Heating	10
Example Program 7 - Logical bitwise operators.....	11
Example Program 8 - Built-in functions	12
Example Program 9	13
Example Program 10.....	14
Example Program 11 - Repetition.....	15
Section 12 - Flowcharts.....	16
Example Program 13 – Procedures & Functions.....	18
Section 14 – By Ref vs. By Val	19
Moderation Exercise 2 – Car Hire	20
Example Program 15 - Constants.....	21
Example Program 16 - Data Structures.....	22
Other Built-in Data Types.....	23
Byte	23
Enumerated	23
Records.....	24
Example Program 17 – 1 dimensional arrays.....	26
Example Program 18a – Read from a text file	27
Example Program 18b – Write to a text file	28
Example Program 19.....	29
Example Program 20 - Validation	30
Example Program 21 – 2 dimensional arrays.....	31
Example Program 22 – Enumerated	32
Section 23 - Set operators.....	33
Union.....	33

Range	33
Intersection	33
Difference (Distinct)	34
Membership (Contains)	36
Example Program 24 - Linear search	37
Example Program 25 - Bubble sort	38
Extension Set A: Form Applications	39
Extension Set B: COMP1 Revision	40

Download and Setup

Download VB Express and install it on your computer at home.

<http://www.microsoft.com/visualstudio/eng/products/visual-studio-2010-express>

If you don't run Windows as your operating system you can try using Mono

<http://www.go-mono.com/mono-downloads/download.html>

Your first program

Start a new Project and select Console Application. Type the following code:

```
module module1
    sub main()
        console.writeline("Hello World!")
        console.ReadKey()
    end sub
end module
```

(Press F5 to run the code)

Tasks

0.1 – Write a program that displays the message “Hello Dave”

0.2 – Write a program that displays the message “My name is Dave and I live in Brussels” (replace Dave and Brussels with your own information)

0.3 – Write a program that displays the lyrics to your national anthem. Each line of the song should be on a fresh line.

Example Program 1 – Sequence

```
Module Module1
```

```
    Sub Main()  
        Console.WriteLine("This is the first line of the program.  It will be executed  
first.")  
        Console.ReadKey()  
        Console.ForegroundColor = ConsoleColor.Red  
        Console.WriteLine("But then the colour changes to Red.")  
        Console.ReadKey()  
        Console.BackgroundColor = ConsoleColor.White  
        Console.WriteLine("And the background goes white")  
        Console.ReadKey()  
        Console.ResetColor()  
        Console.WriteLine("But it's okay.  We can reset it")  
        Console.ReadKey()  
        Console.ForegroundColor = ConsoleColor.Yellow  
        Console.WriteLine("The order of lines of code is important")  
        Console.WriteLine("We start at the top and work down")  
        Console.ReadKey()  
    End Sub
```

```
End Module
```

Tasks

- 1.1) Display the words red, amber, green on a fresh line each in order on the screen. Colour the words appropriately.
- 1.2) Display the names of the colours of the rainbow. Each on a separate line, coloured and in order.

Example Program 2 - Assignment - Integer, byte, real, boolean, character, string, date/time.

Module Module1

```
Sub Main()

    Dim theNumber As Integer
    Dim theWord As String

    theWord = "Bird"
    theNumber = 9

    Console.WriteLine(theWord & " is the word")
    Console.WriteLine("And the number is " & theNumber)

    Console.ReadKey()

    theWord = "Cat"
    Console.WriteLine("Enter a number>")
    theNumber = Int(Console.ReadLine())

    Console.WriteLine("Now " & theWord & " is the word and the number is " &
theNumber)
    Console.ReadKey()

End Sub

End Module
```

Example Program 3 - Arithmetic - +, -, /, x, DIV, MOD

Module Module1

Sub Main()

Dim number1, number2, total As Integer

Console.WriteLine("Enter first number")
number1 = Int(Console.ReadLine())

Console.WriteLine("Enter second number")
number2 = Int(Console.ReadLine())

total = number1 + number2

Console.WriteLine("The total is " & total)
Console.ReadKey()

End Sub

End Module

Tasks

- 3.1) Write a program that divides a number entered by the user by 2
- 3.2) Write a program that displays the 7 times table
- 3.3) Write a program that displays any times table the user requests

Example Program 4 – Selection

```
Dim intInput As Integer

System.Console.WriteLine("Enter an integer...")

intInput = Val(System.Console.ReadLine())

If intInput = 1 Then
    System.Console.WriteLine("Thank you.")
ElseIf intInput = 2 Then
    System.Console.WriteLine("That's fine.")
ElseIf intInput = 3 Then
    System.Console.WriteLine("Too big.")
Else
    System.Console.WriteLine("Not a number I know.")
End If

Console.ReadKey()
```

1. Write a program to ask the user what 24+9 is. Say “Excellent” if they get it right.
2. Write a program to ask the user “how many in a bakers dozen?” and say “most excellent” if they get it right.
3. Write a program to ask the user to enter their age. If their age is under 18 then say “Sorry, this film is not for you”.
4. Write a program to ask the user for two numbers. Compare the first with the second and then print out one of three messages. Either the numbers are equal, the first is bigger, or the second is bigger. You will need more than one IF to solve this one.
5. Write a program which asks the user to enter their password. If they enter the word “PASSWORD” then display the message “Welcome to the treasure”, otherwise display a message which says “go away, it’s all mine”.
6. Write a program which asks the user to enter a number between 1 and 10. If the number entered is out with this range then display a message “Sorry...out of range”.

Example Program 5 - Relational operators - =, <, >, <>, <=, >=

```
Module Module1
```

```
    Sub Main()
```

```
        Dim age As Integer
```

```
        Console.WriteLine("What is your age?")
```

```
        age = Int(Console.ReadLine())
```

```
        If age > 16 Then
```

```
            Console.WriteLine("You can drive!")
```

```
        Else
```

```
            Console.WriteLine("You are too young to drive")
```

```
        End If
```

```
        Console.ReadKey()
```

```
    End Sub
```

```
End Module
```

The symbols we can use to test for conditions are as follows:

< Less than

<= Less Than or Equal To

> Greater than

>= Greater Than or Equal To

== IS Equal To

!= Not Equal To

Example Program 6 - Boolean operators - NOT, AND, OR

Module Module1

Sub Main()

Dim age, points As Integer

Console.WriteLine("What is your age?")
age = Int(Console.ReadLine())

Console.WriteLine("How many points do you have on your licence?")
points = Int(Console.ReadLine())

If age > 16 And points < 9 Then
 Console.WriteLine("You can drive!")
Else
 Console.WriteLine("You are not eligible for a driving licence")
End If

Console.ReadKey()

End Sub

End Module

Moderation Exercise 1 – Central Heating

The heating system in a school should be switched on if the average temperature is less than 17 degrees Celsius (°C). The average temperature is found from the temperatures in the Art, English and Music Departments. You are required to write a program that allows the user to input the three temperatures. The program calculates and displays the average temperature then displays 'Heating should be on.' or 'Heating should be off.' as appropriate.

You may practice this task at home, but won't be able to refer to any notes or files when assessed.

Submit the following for marking in 1 word document:-

1. Your code listing
2. Screenshots for suitable test data

Example Program 7 - Logical bitwise operators - NOT, AND, OR, XOR

```
Dim a, b, c, d, e, f, g As Boolean

a = 23 > 14 And 11 > 8
b = 14 > 23 And 11 > 8
' The preceding statements set a to True and b to False.

c = 23 > 14 Or 8 > 11
d = 23 > 67 Or 8 > 11
' The preceding statements set c to True and d to False.

e = 23 > 67 Xor 11 > 8
f = 23 > 14 Xor 11 > 8
g = 14 > 23 Xor 8 > 11
' The preceding statements set e to True, f to False, and g to False.

Dim x, y As Boolean
x = Not 23 > 14
y = Not 23 > 67
' The preceding statements set x to False and y to True.
```

Tasks

- 1) Adapt the example program so that there is output of the result of each comparison to the screen

Example Program 8 - Built-in functions - Arithmetic functions: round, truncation.

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

Console.Write("Enter a real number")
num = Console.ReadLine()

rounded = Math.Round(num)

squareert = Math.Sqrt(num)

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

trunc = Math.Truncate(num)

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

Tasks

- 1) Write a program that asks for 5 numbers, calculates the mean average and then rounds it down. Display the result on screen.

Example Program 9 - String handling functions *length, position, substring, concatenation.*

```
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.
```

Tasks

1. Write a program that checks a username against a stored value. How the user enters the username should NOT be case sensitive.
2. Adapt program 1 so that it also takes in a password. If the user enters spaces after the password the computer will trim them out automatically.
3. Write a program that will check a phone number is of the correct length.
4. Write a program that asks for a user's full name in one inputbox/textbox but then stores the first and second names in different variables.

Example Program 10 - String conversion functions

to/from integer, real, date/time.

```
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.
```

Tasks

- 1) Using the toString function, recreate the example program in reverse

Example Program 11 - Repetition

```
Module Module1
```

```
    Sub Main()
```

```
        Dim theNumber As Integer
```

```
        theNumber = 7
```

```
        'a loop
```

```
        For x = 1 To 10
```

```
            Console.WriteLine("7 x " & x & " = " & (7 * x))
```

```
        Next
```

```
        'the end of the loop
```

```
        Console.ReadKey() 'pause so user can see
```

```
    End Sub
```

```
End Module
```

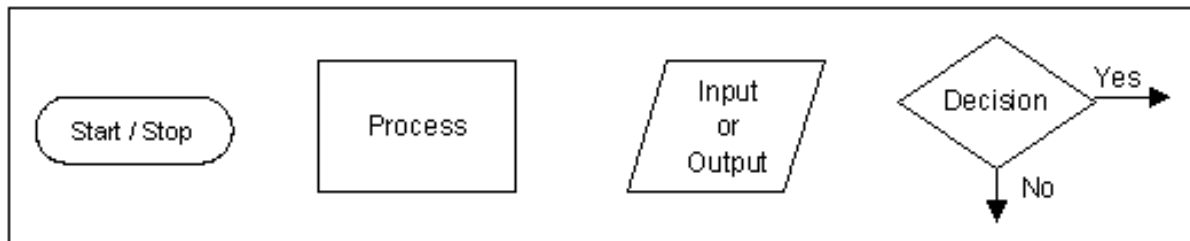
Tasks

1. Write a program which asks for your name and then displays it 5 times on the screen.
2. Write a program to display the name of the town you live in 10 times.
3. Write a program to ask for a person's favourite CD and the artist. Both should be displayed on the same line 5 times.
4. Write a program to ask for a number and display its multiplication table 1 to 100
5. Write a program that asks the user for a number 5 times and adds them all up to give a total.

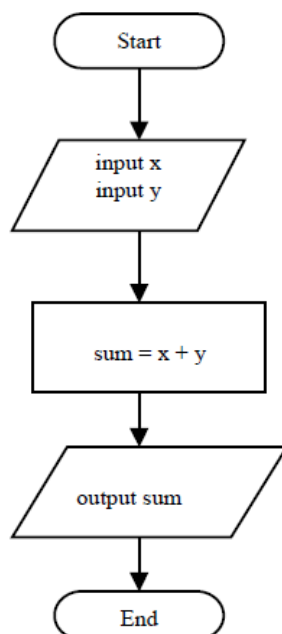
Section 12 - Flowcharts

A flowchart is another way of breaking down a program in the form of a diagram.

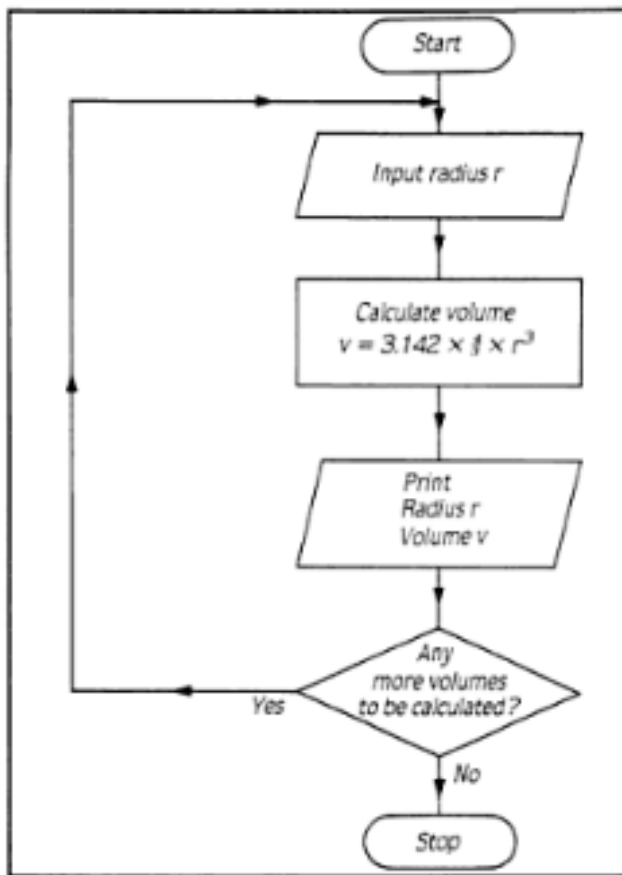
The following are recognised flowchart symbols:



Task 12.1 Create a program from the following flowchart:



Task 12.2 Create a program for the following flowchart:



Task 12.3 - Write a flowchart for example program 6

Task 12.4 - Write a flowchart for example program 11

Example Program 13 – Procedures & Functions

Module Module1

```
'this is a procedure
Sub timestable(ByRef number As Integer)
    For x = 1 To 10
        Console.WriteLine(number & " x " & x & " = " & (number * x))
    Next
End Sub

'this is a function (functions return a value)
Function adder(ByRef a As Integer, ByVal b As Integer)
    adder = a + b
    Return adder
End Function

Sub Main()

    timestable(7) 'this is a call (executes a procedure or function)
    timestable(3) 'this is a second call to the same procedure but now with
different data
    timestable(9)

    Console.ReadKey()

    Console.Clear()

    Dim x As Integer
    x = adder(2, 3) 'call to function adder which returns a value
    Console.WriteLine("2 + 3 = " & x)

    Console.WriteLine("4 + 6 = " & adder(4, 6)) 'you can simply the code by
putting the call directly into the print statement

    Console.ReadKey()

End Sub

End Module
```

Tasks

- 1) Write a function for calculating the volume of a cuboid
- 2) Write a function for calculating the volume of a sphere
- 3) Write a procedure for checking a user's password

Section 14 – By Ref vs. By Val

Parameters can be passed by reference (byref) or by value (byval).

If you want to pass the value of the variable, use the ByVal syntax. By passing the value of the variable instead of a reference to the variable, any changes to the variable made by code in the subroutine or function will not be passed back to the main code. This is the default passing mechanism when you don't decorate the parameters by using ByVal or ByRef.

If you want to change the value of the variable in the subroutine or function and pass the revised value back to the main code, use the ByRef syntax. This passes the reference to the variable and allows its value to be changed and passed back to the main code.

Moderation Exercise 2 – Car Hire

Tervuren Car Hire Company has a range of cars for rent. Charges start at £20 a day for the cheapest to £70 a day for the most expensive. The company requires a program that gives customers printed details of charges. The program asks the user for a car make, model and daily rate¹ then displays a table of daily charges for between one and fourteen days. If the make is VW, a message indicating a gift of a free road atlas is displayed.

An example of output is shown below. The output from your program may look different but must meet the specification.

```
TERVUREN CAR HIRE COMPANY

Enter car make.  VW
Enter car model. Golf
Enter daily rate. 22

      Days      Charge (£)
      1         22
      2         44
      3         66
      4         88
      5        110
      6        132
      7        154
      8        176
      9        198
     10        220
     11        242
     12        264
     13        286
     14        308

YOU GET A FREE ROAD ATLAS WITH THIS HIRE.
```

You may practice this task at home, but won't be able to refer to any notes or files when assessed.

Submit the following for marking in 1 word document:-

1. Your code listing
2. Screenshots for suitable test data

¹ Daily rates are in whole numbers of pounds.

Example Program 15 - Constants

```
'a constant is a value that doesn't change
'using them greatly improves the readability of your code

'number constants
Const conPi = 3.14159265358979
Const conMaxPlanets As Integer = 9

'string constants
Const conVersion = "07.10.A"
Const conCodeName = "Enigma"

'try assigning a new value to one of your constants

'to make a constant available to the whole program "Public" should precede it
'and it should be placed inside the module but outside the procedure
'try it

'there are string constants created by VB
Console.WriteLine("Pi is" & conPi)
Console.WriteLine("Pi is" & vbCr & conPi)
Console.WriteLine("Pi is" & vbTab & conPi)

Console.ReadKey()
```

Example Program 16 - Data Structures

Visual Basic has many data type such as Integer, String, Double, Single, etc. Aside from those intrinsic data type that has been provided by VB, you can create your own data type by combining several different types. This UDT is useful to create single variable contain several useful information. It is usually useful for database record. You define the UDT using syntax:

```
Type VariableType
Variable1 as varType1
Variable2 as varType2
....
VariableN as varTypeN
End Type
```

After that you can declare any name as this new variable type and use it as ordinary variable.

Note: Special for this project, please do not copy and paste from this tutorial. Type the code manually so that you may understand the different between UDT and normal variable.

Create new console project

In the general declaration section, type:

```
Private Type City
```

```
Name As String
Population As Long
Diameter As Double ' in km
Year As Integer
```

```
End Type
```

```
Sub Main()
Dim myRecord As City
myRecord.Name = Console.ReadLine()
myRecord.Population = Console.ReadLine()
myRecord.Year = Console.ReadLine()
myRecord.Diameter = Console.ReadLine()
Console.WriteLine(myRecord.Name & " city has population of " & _ myRecord.Population & "
million people " & vbCr & " and diameter of " & myRecord.Diameter & " km in year " & _
myRecord.Year
End Sub
```

Tasks

- 1) Run the program and try the data for some real city. You see that UDT combine several data type into single unit.
- 2) Adapt the program so each input has a suitable prompt

Other Built-in Data Types

We are familiar with int, double, boolean, string and char already.

There are also bytes and dates/times.

Byte

Bytes are 8-bit (1 byte) unsigned integers ranging in value from 0 to 255 (0 to 2^8-1). You can declare Byte variables using the BYTE keyword with the DIM statement. For example:

DIM I AS BYTE

Byte variables are particularly useful for storing small, unsigned integral quantities like the number of days in a month. You should not use Byte variables in FOR/NEXT loops, as they are highly inefficient.

Enumerated: is a data type consisting of a set of named values called elements

```
Enum CardSuit
    Clubs
    Diamonds
    Hearts
    Spades
End Enum
```

```
Enum DayOfWeek
    Monday = 1
    Tuesday = 2
    Wednesday = 3
    Thursday = 4
    Friday = 5
    Saturday = 6
    Sunday = 7
End Enum
```

```
Enum Result
    Win = 3
    Lose = 1
    Draw = 0
End Enum
```


Records

In VB, Records are known as structures. They allow you to combine several data items (or fields) into one variable. For example in college instead of having a separate variable to store information about each fact about a student, you would create a student record with fields to store each fact (or field!):

Structure student 'record declaration

```
dim id as integer 'field
```

```
dim name as string 'field
```

```
dim DoB as date 'field
```

End Structure

Sub Main()

```
dim newStudent as student
```

```
console.writeline("insert the id")
```

```
newStudent.id = console.readline()
```

```
console.writeline("insert the name")
```

```
newStudent.name = console.readline()
```

```
console.writeline("insert the Date of Birth")
```

```
newStudent.DoB = console.readline()
```

```
console.writeline("new record created: " & newStudent.id & " " & newStudent.name & " " &  
newStudent.DoB)
```

End Sub

But why should we use Records? Well it makes for much neater code, without having to declare dozens of variables, we could declare a couple of students. You'll also see how neat your code can become when using arrays of records.

Tasks

- 1) Test the above code
- 2) Add an additional field to the record
- 3) Create a record structure for cars and test it with a main method
- 4) Create a record structure for pets and test it with a main method

Example Program 17 – 1 dimensional arrays

```
Dim countries(5) As String
Dim randomNum As Integer

countries(1) = "Scotland"
countries(2) = "Belgium"
countries(3) = "Netherlands"
countries(4) = "Germany"
countries(5) = "France"

Randomize()
randomNum = Int(Int((5 * Rnd()) + 1))

Console.WriteLine("You should go to " & countries(randomNum) & " on holiday.")
Console.ReadKey()
```

Tasks

- 1) Write a program which will set up an array to hold 50 numbers. Call the array numbers. Display the array's contents across the screen. They should all be 0.
- 2) Create a program that stores an array of car records. At least 5 cars and 4 fields per record.
- 3) Create a program that stores an array of 5 people records. The information should be entered by the user.
- 4) Adapt program 2 to now do a linear search for a certain car and display it's details.

Example Program 18a – Read from a text file

```
Dim objStreamReader As IO.StreamReader
Dim strLine As String

'Pass the file path and the file name to the StreamReader constructor.
objStreamReader = New IO.StreamReader("H:\Diary.txt")

'Read the first line of text.
strLine = objStreamReader.ReadLine

'Continue to read until you reach the end of the file.
Do While Not strLine Is Nothing

    'Write the line to the Console window.
    Console.WriteLine(strLine)

    'Read the next line.
    strLine = objStreamReader.ReadLine
Loop

'Close the file.
objStreamReader.Close()

Console.ReadLine()
```

Tasks

- 1) Write a program that reads the students' names from a txt file and displays them on the screen
- 2) Write a program that reads 10 team names from a txt file and stores them in an array
- 3) Write a program that reads 5 song titles from a csv file and displays them on the screen
- 4) Write a program that reads 20 team names from a csv file into an array, then displays the array on screen

Example Program 18b – Write to a text file

```
Dim objStreamWriter As IO.StreamWriter

'Pass the file path and the file name to the StreamWriter constructor.
objStreamWriter = New IO.StreamWriter("H:\Testfile.txt")

'Write a line of text.
objStreamWriter.WriteLine("Hello World")

'Write a second line of text.
objStreamWriter.WriteLine("From the StreamWriter class")

'Close the file.
objStreamWriter.Close()
```

Tasks

- 1) Write a program that asks for 5 names and then writes them to a file
- 2) Write a program that writes the colours of the rainbow to a csv file
- 3) Write a program that reads a hiscore table from a file, asks the user to add a new hiscore, then writes this new hiscore table to the file.

Example Program 19 - Read/write records from/to a file of records.

```
Imports System.IO
Imports System.Runtime.Serialization.Formatters.Binary

Module Module1

    Structure person
        Dim name As String
        Dim age As Integer
        Dim alive As Boolean
    End Structure

    Dim File As String

    Sub Main()

        Dim student1 As person
        student1.name = "John"
        student1.age = 17
        student1.alive = True

        Dim student2 As person
        student2.name = "Emily"
        student2.age = 16
        student2.alive = True

        File = "H:\test.txt" 'location of the file

        ' write file
        FileOpen(1, File, OpenMode.Binary, OpenAccess.Write)
        FilePut(1, student1)
        FilePut(1, student2)
        FileClose(1)

        ' read file
        FileOpen(1, File, OpenMode.Binary, OpenAccess.Read)
        FileGet(1, student1)
        FileGet(1, student2)
        FileClose(1)

        'display data
        Console.WriteLine("Student1: " & student1.name & vbTab & student1.age & vbTab
& student1.alive)
        Console.WriteLine("Student2: " & student2.name & vbTab & student2.age & vbTab
& student2.alive)
        Console.ReadKey()

    End Sub

End Module
```

Tasks

- 1) Amend the program so that there is a write **procedure** and a read **procedure** which are called
- 2) Amend the program so that the 2 student are stored in an array of people, then write and read the array to the file

Example Program 20 - Validation

```
Dim mark As Integer

Do
    Console.WriteLine("Enter a mark between 0 and 10")
    mark = Val(Console.ReadLine())
    If (mark > 10) Or (mark < 0) Then Console.WriteLine("That was not a valid mark")
Loop Until (mark >= 0) And (mark <= 10) ' keeps going until a valid mark is entered

Console.WriteLine("Well done!")
Console.ReadKey()
```

Tasks

- 1) Write a program that validates a user is old enough to drive (older than 17, younger than 80)
- 2) Write a program that checks that a telephone number entered is long enough (string length)
- 3) Write a program that checks that both a username and password is correct before allowing you to proceed.

Example Program 21 – 2 dimensional arrays

```
' Declare two-dimensional array of strings.
Dim values(,) As String = New String(,) {{"ant", "aunt"},
    {"Sam", "Samantha"},
    {"clozapine", "quetiapine"},
    {"flomax", "volmax"},
    {"toradol", "tramadol"}}

' Get bounds of the array.
Dim bound0 As Integer = values.GetUpperBound(0)
Dim bound1 As Integer = values.GetUpperBound(1)

' Loop over all elements.
For i As Integer = 0 To bound0
    For x As Integer = 0 To bound1
        ' Get element.
        Dim s1 As String = values(i, x)
        Console.Write(s1)
        Console.Write(" ")
    Next
    Console.WriteLine()
Next

Console.ReadKey()
```

Tasks

- 1) Create a 2d array which stores and displays a grid for noughts and crosses. Allow users to pick a location and set it to an X or a O.
- 2) 2d array which stores a grid for Battleships. Allows the user to place their 5 ships and isplay on screen.

Example Program 22 – Enumerated

An Enumeration is basically a new data type (like String or Integer) that you design yourself, which has an associated list of possible values (known as elements).

You might think it would be better to use a String to hold the current value instead, but there are several reasons why an Enum is better, such as:

- The list of valid options will be shown in a pop-up list when you are writing code (see the picture below, after the first Code snippet), so you don't need to remember them or even type them - instead you can just select them with your mouse or keyboard.
- It is harder to make a typo that causes bugs, as not only are the values shown in the list for you to pick from, but also valid entries you type will be changed to the case you declared them - so you can see when the case is not automatically corrected for you.
- If you use Option Explicit (which is highly recommended anyway), any typo's will cause a message warning you about them.
- They take less memory than Strings, and code that uses them runs faster than the String equivalent.

```
Enum Importance
    Critical = 4
    Important = 3
    None = 0
    Regular = 2
    Trivial = 1
End Enum

Sub Main()
    Dim value As Importance = Importance.Critical
    ' Select the enum and print a value.
    Select Case value
        Case Importance.Trivial
            Console.WriteLine("Not true")
            Return
        Case Importance.Critical
            Console.WriteLine("True")
            Exit Select
    End Select
End Sub
```

Section 23 - Set operators

Union

```
' Create two arrays of integer values.
Dim ints1() As Integer = {5, 3, 9, 7, 5, 9, 3, 7}
Dim ints2() As Integer = {8, 3, 6, 4, 4, 9, 1, 0}

' Get the set union of the two arrays.
Dim union As IEnumerable(Of Integer) = ints1.Union(ints2)

' Display the resulting set's values.
Dim output As New System.Text.StringBuilder
For Each num As Integer In union
    output.AppendLine(num & " ")
Next
Console.WriteLine(output.ToString())
Console.ReadKey()
```

Range

```
' Generate a sequence of integers from 1 to 10
' and project their squares.
Dim squares As IEnumerable(Of Integer) = Enumerable.Range(1,
10).Select(Function(x) x * x)

Dim output As New System.Text.StringBuilder
For Each num As Integer In squares
    output.AppendLine(num)
Next

' Display the output.
Console.WriteLine (output.ToString())
Console.ReadKey()
```

Intersection

```
' Create two integer arrays.
Dim id1() As Integer = {44, 26, 92, 30, 71, 38}
Dim id2() As Integer = {39, 59, 83, 47, 26, 4, 30}

' Find the set intersection of the two arrays.
Dim intersection As IEnumerable(Of Integer) = id1.Intersect(id2)

Dim output As New System.Text.StringBuilder
For Each id As Integer In intersection
    output.AppendLine(id)
Next

' Display the output.
Console.WriteLine (output.ToString)
Console.ReadKey()
```

Difference (Distinct)

```

Module Module1
    Sub Main()
        Dim products() As Product =
        {New Product With {.Name = "apple", .Code = 9},
        New Product With {.Name = "orange", .Code = 4},
        New Product With {.Name = "apple", .Code = 9},
        New Product With {.Name = "lemon", .Code = 12}}

        ' Exclude duplicates.

        Dim noduplicates = products.Distinct(New ProductComparer())

        For Each product In noduplicates
            Console.WriteLine(product.Name & " " & product.Code)
        Next

        Console.ReadKey()

    End Sub
End Module

Public Class Product
    Public Property Name As String
    Public Property Code As Integer
End Class

' Custom comparer for the Product class
Public Class ProductComparer
    Implements IEqualityComparer(Of Product)

    Public Function Equals1(
        ByVal x As Product,
        ByVal y As Product
    ) As Boolean Implements IEqualityComparer(Of Product).Equals

        ' Check whether the compared objects reference the same data.
        If x Is y Then Return True

        ' Check whether any of the compared objects is null.
        If x Is Nothing OrElse y Is Nothing Then Return False

        ' Check whether the products' properties are equal.
        Return (x.Code = y.Code) AndAlso (x.Name = y.Name)
    End Function

    Public Function GetHashCode1(
        ByVal product As Product
    ) As Integer Implements IEqualityComparer(Of Product).GetHashCode

        ' Check whether the object is null.
        If product Is Nothing Then Return 0

        ' Get hash code for the Name field if it is not null.
        Dim hashProductName =
            If(product.Name Is Nothing, 0, product.Name.GetHashCode())

        ' Get hash code for the Code field.
        Dim hashProductCode = product.Code.GetHashCode()
    
```

```
        ' Calculate the hash code for the product.  
        Return hashProductName Xor hashProductCode  
    End Function  
End Class
```

Membership (Contains)

```
' Create an array of strings.
Dim fruits() As String = {"apple", "banana", "mango", "orange",
"passionfruit", "grape"}

' This is the string to search the array for.
Dim fruit As String = "mango"

' Determine if the array contains the specified string.
Dim hasMango As Boolean = fruits.Contains(fruit)

Dim text As String = IIf(hasMango, "does", "does not")

' Display the output.
Console.WriteLine ("The array " & text & " contain " & fruit)
Console.ReadKey()
```

Example Program 24 - Linear search

```

Sub Main()

    Dim array1 As Integer() = New Integer(19) {}

    Dim randomNumber As Random = New Random()
    Dim index As Integer

    ' creates string containing 11 random numbers
    For index = 0 To array1.GetUpperBound(0)
        array1(index) = randomNumber.Next(1000)
        Console.Write(array1(index) & " ")
    Next
    Console.WriteLine("")

    Console.WriteLine("What number do you want to search for?")
    Dim searchKey As Integer = Console.ReadLine

    Dim element As Integer = LinearSearch(searchKey, array1)

    If element <> -1 Then
        Console.WriteLine("Found Value in index " & element)
    Else
        Console.WriteLine("Value Not Found")
    End If

    Console.WriteLine("Press any key to close")
    Console.ReadKey()

End Sub

Function LinearSearch(ByVal key As Integer, ByVal numbers As Integer()) As Integer

    Dim n As Integer

    ' structure iterates linearly through array
    For n = 0 To numbers.GetUpperBound(0)
        If numbers(n) = key Then
            Return n
        End If
    Next

    Return -1
End Function ' LinearSearch

```

Tasks

- 1) Search for the position of a player in a teamsheet (array) by surname
- 2) Search for the position of a song in a top 10 chart by title and return it's position
- 3) Using 2 arrays, create an address book of friends. 1 array for names, 1 for addresses.
Prompt the user for a name and then return that persons address.
- 4) Add mobile telephone numbers to the previous program (Hint: 3rd array needed)

Example Program 25 - Bubble sort

```

Sub Main()

    System.Console.WriteLine("Enter any sequence of numbers followed by a space.")

    'read the array to a string
    Dim OurArrayString As String = System.Console.ReadLine()

    Dim OurArray As String()
    'split the string to an array
    OurArray = OurArrayString.Split(" ")

    'Start Bubble sort algorithm
    Dim i As Integer
    Dim j As Integer

    For i = 0 To UBound(OurArray) Step 1
        'Ubound of an array will be the maximum index of that array

        'start another for loop
        For j = 0 To UBound(OurArray) - 1

            If CInt(OurArray(j + 1)) < CInt(OurArray(j)) Then

                'Swapping the variables in the array
                Dim temp As Integer = CInt(OurArray(j + 1))
                OurArray(j + 1) = OurArray(j)
                OurArray(j) = temp

            End If

        Next
    Next

    'display the out in the console window
    Console.WriteLine("the sorted array will be")
    For Each x In OurArray
        Console.Write(x & " ")
    Next

    Console.ReadLine()

End Sub

```

Tasks

- 1) Write a program that sorts an array of ages.
- 2) Write a program that reads in scores from a file, sorts them and displays them
- 3) Add to this program so that there is a corresponding list of names which is also sorted to go with the score.
- 4) Add to this program so that it is possible for a user to add an additional hiscore and it appears in the correct position when displayed.

Extension Set A: Form Applications

Hello World

<http://howtostartprogramming.com/vb-net/vb-net-tutorial-2-hello-world/>

Form Properties

<http://howtostartprogramming.com/vb-net/vb-net-tutorial-6-form-properties/>

Progress Bar

<http://howtostartprogramming.com/vb-net/vb-net-tutorial-7-progress-bar/>

Menu Strip

<http://howtostartprogramming.com/vb-net/vb-net-tutorial-10-menu-strip/>

Advanced Message Box

<http://howtostartprogramming.com/vb-net/vb-net-tutorial-20-advanced-messagebox/>

Web Browser

<http://howtostartprogramming.com/vb-net/vb-net-tutorial-31-web-browser/>

Try Catch

<http://howtostartprogramming.com/vb-net/vb-net-tutorial-38-try-catch/>

Extension Set B: COMP1 Revision

- 1) Create a program that converts decimal to binary and back
- 2) Create a program that converts decimal to hex and back
- 3) Create a program that codes a string into its ASCII codes and back
- 4) Create a program that adds binary numbers
- 5) Create a program that multiplies binary numbers
- 6) Create a program that subtracts binary numbers
- 7) Create a program that converts a decimal number to floating point and back.
- 8) Create a program that displays a number of bits as bytes, kilobytes, megabytes.....terabytes
- 9) Create a program that calculates a pictures file size
- 10) Create a program that calculates a sound files size

