**VB.Net Console Programming Support**

**v1.0**

Coding the VB.Net console is a core requirement of the A Level course. The examples and exercises will help you gain an understanding of this. This material is derived from the “Fundamentals of Programming” AQA course on Wikibooks Examples which are followed by (1) are creative commons cod from this course.

A number of code examples followed by (2) are derived from the VB.Net Console Student Booklet. In general they have been modified however in these cases it may be helpful to refer to this guide, which is available online. All the other tasks marked with (\*) are created by author, and those marked with (#) are derived from code which is attribution free. Where there is a # in front of the number it means that additional support is available from the VB.Net Console Booklet

1. Basic Hello World Program (1)

module module1

 Sub main()

 console.writeline("Hello World!")

 console.readline()

 End Sub

End Module

Task 1.1 Change the text to “Hello John Doe”

Task 2.2 Change the text to your first name and family name split over two lines

(#1) 2. Sequence of code execution (2)

Module Module1

 Sub Main()

 Console.WriteLine("This is the first line of the program.")

 Console.ReadKey()

 Console.ForegroundColor = ConsoleColor.Red

 Console.WriteLine("The colour changes to Red.")

 Console.ReadKey()

 Console.BackgroundColor = ConsoleColor.White

 Console.WriteLine("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("The code starts at the beginning and works through")

 Console.ReadKey()

 End Sub

End Module

Task 2.1 Change the colours and text of this program

Task 2.2 Display a range of five colours in the order they would appear in a rainbow

(#2) 3. Assignment of variables

Module Module1

 Sub Main()

 Dim myNumber As Integer

 Dim myText As String

 myText = "Zebra"

 myNumber = 6

 Console.WriteLine(myText & " - the text used")

 Console.WriteLine("The number used is " & myNumber)

 Console.ReadKey()

 myText = "Man"

 Console.WriteLine("Enter your number")

 myNumber = Int(Console.ReadLine())

 Console.WriteLine("After the input " & myText & " is the text, no is " & myNumber)

 Console.ReadKey()

 End Sub

End Module

 The following code is used to place the value of a variable in a piece of text. The {0} refers to the first variable after the comma.

Console.WriteLine("The average monthly salary is {0} ", averageSalary)

Comments can be added by placing an ‘ before the text

'This is a comment

Task 3.1 Change the text and number assigned to the variable

(#3) 4. Arithmetic Operations (2)

An operator is a programming device that performs a function on one or more inputs, for example arithmetic operators (+,-,/,\*)

Visual Basic .NET provides a basic set of operators to calculate simple arithmetic

+ Addition

- Subtraction

\* Multiplication

/ Division

\ Integer division

Mod Remainder Division

^ Exponentiation

& String concatenation

7 + 2 produces 9

7 - 2 produces 5

7 \* 2 produces 14

7 / 2 produces 3.5

7 \ 2 produces 3

7 Mod 2 produces 1

7 ^ 2 produces 49

"7" & "7" produces "77"

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

BODMAS

You have probably learnt about the order of operations in maths. BODMAS also applies to computer calculations. This means that when calculating a sum, the program will calculate:

Brackets

Order (powers n^2 etc)

Division

Multiplication

Addition

Subtraction

Task 4.1 Write code to calculate (10 + 2) \* number1 \* number2 where the input numbers are 7 and 3. Then calculate 10 + 2 \* number1 \* number2 with the same input numbers to see the operation of this.

Task 4.2 Write code to work out a 50% commission on a sale

Task 4.3 Write code to work out a 27% commission on a sale

Task 4.4 Write code to turn miles to kilometres

Task 4.5 Write code to write the 9 times table

Task 4.6 Write code that takes user input to create a specific times table

Task: Create a calculator which takes to numbers and performs a calculation on the numbers

The calculator should state

“Enter your first number”

Then it should state

“Enter your second number”

Then it should state

“Choose between the following + - \* /”

The output should say “The answer is “ and give the answer

Stretch and challenge

Add the ability to do powers using the “^” option

Task 2: Modify the code below to see the output of Mod under different circumstances

Module Module1

 Sub Main()

 Dim firstNum As Integer

 Dim secondNum As Integer

 Dim answer As Integer

 Dim arithmeticProcess As String

 Dim myMod As Integer

 Dim myCount As Integer

 myCount = 1

 firstNum = 1

 secondNum = 7

 Do Until myCount = 30

 Console.WriteLine(myCount)

 answer = firstNum / secondNum

 myMod = firstNum Mod secondNum

 Console.WriteLine("The answer is {0}", answer)

 Console.WriteLine("The modulus is {0}", myMod)

 myCount = myCount + 1

 firstNum = firstNum + 1

 Loop

 Console.ReadLine()

 End Sub

End Module

(#4) 5. Selection (2)

Selection is the principle of choosing what action to take based on certain criteria.

Module Module1

 Sub Main()

 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()

 End Sub

End Module

5a. Selection without the use of Val, this code also works with integers

Module Module1

 Sub Main()

 Dim myName As String

 Console.WriteLine("Who are you ?")

 myName = Console.ReadLine()

 If myName = "Bob" Then

 Console.WriteLine("Hi Bob")

 Else : Console.WriteLine("I don't know you")

 End If

 Console.ReadKey()

 End Sub

End Module

Task

5.1 Get the user to multiply two numbers together and then based on the outcome state correct or incorrect.

5.2 A username should be entered and then the user should be given “open access to the system” or “partial access to the system” depending on their answer.

5.3 A bar tender is asked whether the customers looks under 25. If they do they should be instructed to ask for ID.

The use of Val

In the code, intInput = Val(System.Console.ReadLine()) ,Val is used. This affects the operation of the code. Both of the following will read the line and take the input and assign it to the variable. The difference is that one uses Val to do this and the other does not

intInput = Console.ReadLine()

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

The difference between the two is that the of the second means that if a user enters something after the number it will be ignored by the code. In the first case if 3a is entered the application will crash, however in the second case the use of Val means that the code will run and the a will be ignored. The Val function stops reading the string at the first character it cannot recognise as a number.

The MSDN provides the following examples to explain the detail

Dim valResult As Double

' The following line of code sets valResult to 2457.

valResult = Val("2457")

' The following line of code sets valResult to 2457.

valResult = Val(" 2 45 7")

' The following line of code sets valResult to 24.

valResult = Val("24 and 57")

(#5) 6. Relational Operators (\*)

The following relational operators are available in Visual Basic.

=, <, >, <>, <=, >= and they stand for equals, less than, greater than, not equal to, less than or equal to and greater than or equal to.

An operand is a value within an expression

MSDN defines the relational operators as follows

The = operator tests whether the two operands are equal.

The <> operator tests whether the two operands are not equal.

The < operator tests whether the first operand is less than the second operand.

The > operator tests whether the first operand is greater than the second operand.

The <= operator tests whether the first operand is less than or equal to the second operand.

The >= operator tests whether the first operand is greater than or equal to the second operand.

6.1

Module Module1

 Sub Main()

 Dim i As Integer

 Console.WriteLine("Enter a number")

 i = Console.ReadLine()

 If i < 0 Then

 Console.WriteLine("the number is less than 0")

 ElseIf i <= 100 And i >= 0 Then

 Console.WriteLine("the number is less than or equal to one hundred and greater than or equal to 0")

 ElseIf i > 100 And i < 200 Then

 Console.WriteLine("the number is greater than one hundred less than 200")

 Else

 Console.WriteLine("the number is greater than or equal to 200")

 End If

 Console.ReadLine()

 End Sub

End Module

6.2

Module Module1

 Sub Main()

 Dim myWealth As Integer

 Console.WriteLine("How rich are you ?")

 myWealth = Console.ReadLine()

 If myWealth > 1000000 Then

 Console.WriteLine("You are a millionaire")

 Else : Console.WriteLine("You are not a millionaire")

 End If

 Console.ReadKey()

 End Sub

End Module

Task 6.1 Write a program that asks people for their salary. If it is below a minimum wage of £6 it tells the user that the employer is paying below the minimum wage.

Task 6.2 Use the same program but use >= to deal to additionally tell the user that their salary is below the minimum wage if it is £6.

Rounding (ac)

The following code deals with rounding. It shows how to round to a certain number of decimal points.

Module Module1

 Sub Main()

 Dim before As Double = 123.44896876

 Dim after1 As Double

 Dim after2 As Double

 after1 = Math.Round(before)

 after2 = Math.Round(before, 5)

 Console.WriteLine(after1)

 Console.WriteLine(after2)

 Console.ReadLine()

 End Sub

End Module

**Currency Rounding**

 Dim num1 As Double = 2000

 Dim num2 As Double = 500

 Dim ans As Double = num1 + num2

 Console.WriteLine(FormatCurrency(ans))

 Console.ReadLine()

Module Module1

 Sub Main()

 Dim sex As String

 Dim age As Integer

 Dim pension As Integer

 Dim years As Integer

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

 age = Console.ReadLine()

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

 sex = Console.ReadLine()

 Console.WriteLine("How many years have you worked ?")

 years = Console.ReadLine()

 If age > 60 And sex = "female" Then

 pension = 500

 ElseIf age > 65 And sex = "male" Then

 pension = 500

 Else

 pension = 0

 End If

 If years > 30 Then

 pension = pension \* 1

 ElseIf years < 30 Then

 pension = years \* (500 / 30)

 End If

 Console.WriteLine("Your pension is {0}", pension)

 Console.ReadLine()

 End Sub

End Module

Task 6.3a stretch (ac)

You task is to produce a computer program to work out a pension. Women get a pension at 60 and men at 65. The pension is £500 a month. Take input from users and provide output to tell them whether they are eligible for a pension.

Task 6.3b stretch (ac)

A new rule is implemented. You can only get a pension if you have worked for 30 years. If you have worked for less the amount is reduced on a pro rata basis depending on the number of years worked. Calculate the pension that will be received.

Target time 10 mins

Random numbers (ac)

' Initialize the random-number generator.

Randomize()

' Generate random value between 1 and 6.

Dim value As Integer = CInt(Int((6 \* Rnd()) + 1))

Module Module1

 Sub Main()

 ' Initialize the random-number generator.

 Randomize()

 ' Generate random value between 1 and 6.

 Dim value As Integer

 value = (3 \* Rnd()) + 1

 Console.WriteLine(value)

 Console.ReadLine()

 End Sub

End Module

Exercise RN 1

Create a listing for a set of spins of a roulette wheel. A Roulette wheel has 36 numbers and a 0. The odds are red and the evens black. The zero is green. You should ignore the zero in the first version of your program. Your output should state whether the number is number is red or black. You should preferably use modulus to achieve this.

Output required

The number is 32 black

The number is 17 red

The number is 1 red

The number is 18 black

Module Module1

 Sub Main()

 ' Initialize the random-number generator.

 Randomize()

 Dim value As Integer

 Dim myMod As Integer

 For x = 1 To 100

 value = (35 \* Rnd()) + 1

 myMod = value Mod 2

 If myMod = 0 Then

 Console.WriteLine("Number is " & value & " Red")

 End If

 If myMod = 1 Then

 Console.WriteLine("Number is " & value & " Black")

 End If

 Next

 Console.ReadLine()

 End Sub

End Module

#6 7. Boolean Operators (ac)

**Booleans**

Booleans are values that are true or false. They are like any other variable except that the output is always either true or false. Four types of Booleans are covered And, Or, XOr and Not. The easiest way to understand Booleans is to use them in conjunction with relational operators and to create values that are true or false based on these relational operators.

**And**

And yields a true result if both conditions are met but not if one condition is met

Module Module1

 Sub Main()

 Dim age As Integer

 Dim sex As String

 Dim govtPayment As Boolean

 Console.WriteLine("What is your age")

 age = Console.ReadLine()

 Console.WriteLine("What is your sex")

 sex = Console.ReadLine()

 govtPayment = age > 60 And sex = "female"

 Console.WriteLine(govtPayment)

 Console.ReadLine()

 End Sub

End Module

**Or**

Or yields a true result if either of two conditions are met

Module Module1

 Sub Main()

 Dim age As Integer

 Dim govtPayment As Boolean

 Console.WriteLine("What is your age")

 age = Console.ReadLine()

 govtPayment = age > 65 Or age < 5

 Console.WriteLine(govtPayment)

 Console.ReadLine()

 End Sub

End Module

**Xor**

The Xor is an exclusive or. This means that you that the result is true if either or the results are true but not if they are both true

In the following case you get government payments if you are over 60 or a ‘newmother’. If you are both you don’t get payment because you are deemed to be lying.

Module Module1

 Sub Main()

 Dim age As Integer

 Dim parentalStatus As String

 Dim govtPayment As Boolean

 Console.WriteLine("What is your age")

 age = Console.ReadLine()

 Console.WriteLine("What is your parental status")

 parentalStatus = Console.ReadLine()

 govtPayment = age > 60 Xor parentalStatus = "newmother"

 Console.WriteLine(govtPayment)

 Console.ReadLine()

 End Sub

End Module

**Not**

Not yields the result false is a condition is met and true if it is not met

Module Module1

 Sub Main()

 Dim age As Integer

 Dim jobStatus As String

 Dim govtPayment As Boolean

 Console.WriteLine("What is your job status")

 jobStatus = Console.ReadLine()

 govtPayment = Not jobStatus = "employed"

 Console.WriteLine(govtPayment)

 Console.ReadLine()

 End Sub

End Module

You should create a series of programs using Booleans to determine whether a student should apply to study Computer Science at London University

If the students is doing Physics and Economics they should apply

If the student is studying Computer Science or Maths they should apply

If they are studying Computer Science or Maths but not both they should apply

If the student does not have Philosophy as their best subject they should apply

You should use all four Booleans studied to yield a True Result in each case

The output should be an appropriate True of False for each case

**Assessment 1 : Obtaining a mortgage for a £300,000 house**

Your ability to gain a mortgage is dependent on your average salary over three months. You should create a program that asks for the salary each month and then displays the average salary over the three month period. If the average salary is £5,000 per month the program should say “You can obtain a mortgage.” If the amount is below £5,000 the code should say “You cannot obtain a mortgage”.

You can do the exercise prior to the test but will not be able to use notes during the assessment test.

At the end of the assessment you should put a word document with the code and a screenshot of the output of your program in it.

Answer: Mortgage question

Module Module1

 Sub Main()

 Dim monthOne As Integer

 Dim monthTwo As Integer

 Dim monthThree As Integer

 Dim averageSalary

 Console.WriteLine("What is the salary in month one ?")

 monthOne = Console.ReadLine()

 Console.WriteLine("What is the salary in month two ?")

 monthTwo = Console.ReadLine()

 Console.WriteLine("What is the salary in month three ?")

 monthThree = Console.ReadLine()

 averageSalary = (monthOne + monthTwo + monthThree) / 3

 Console.WriteLine("The average monthly salary is {0} ", averageSalary)

 ' Console.WriteLine(averageSalary)

 If averageSalary > 5000 Then

 Console.WriteLine("You are eligible for a mortgage")

 ElseIf averageSalary < 5001 Then

 Console.WriteLine("You are not eligible for a mortgage")

 End If

 Console.ReadLine()

 End Sub

End Module

To add

1. Explain the meaning and purpose of Val
2. Add exercises

The use of Val

In the code, intInput = Val(System.Console.ReadLine()) ,Val is used. This affects the operation of the code. Both of the following will read the line and take the input and assign it to the variable. The difference is that one uses Val to do this and the other does not

intInput = Console.ReadLine()

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

The difference between the two is that the of the second means that if a user enters something after the number it will be ignored by the code. In the first case if 3a is entered the application will crash, however in the second case the use of Val means that the code will run and the a will be ignored. The Val function stops reading the string at the first character it cannot recognise as a number.

The MSDN provides the following examples to explain the detail

Dim valResult As Double

' The following line of code sets valResult to 2457.

valResult = Val("2457")

' The following line of code sets valResult to 2457.

valResult = Val(" 2 45 7")

' The following line of code sets valResult to 24.

valResult = Val("24 and 57")

#8 8. Built in functions (2)

VB.Net has built in math functions that allow you to change a variable into a form appropriate for the needs of a program. It may be that your output relates to an age in which case you may wish to round the answer to an integer. The following program illustrates a number of examples.

Module Module1

 Sub Main()

 Dim num As Double

 Dim rounded As Integer

 Dim squarert As Double

 Dim trunc As Integer

 Console.Write("Enter a real number")

 num = Console.ReadLine()

 rounded = Math.Round(num)

 squarert = Math.Sqrt(num)

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

 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

String handling, creating a string from within a longer string (ac)

Module Module1

 Sub Main()

 Dim myString As String

 Console.WriteLine("Enter a string")

 myString = Console.ReadLine()

 Console.WriteLine(myString.Substring(5)) 'all characters after position 5

 Console.WriteLine(myString.Substring(5, 3)) 'three characters from position 12

 Console.ReadLine()

 End Sub

End Module

#9 9. String handling functions (2)

It is important within programs to be able to handle strings effectively. IT may be necessary to break out a substring (part of a string) This is useful when seeking a piece of text within a string and has a wide range of applications.

Module Module1

 Sub Main()

 Dim theString As String

 theString = "This is a sting that contains the word Dave"

 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)) 'shows 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

#10. String Conversion Functions (2)

The following code illustrates how VB.net deals with a variable of the type string. It shows how you move a variable which is a date forward one day, and how that is different to the change of an integer

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

String Conversion (ac)

Module Module1

 Sub Main()

 Dim x As Double

 Dim y As String

 x = 3

 x = x + x

 Console.WriteLine(x)

 y = x

 y = y + y

 Console.WriteLine(y)

 Console.ReadLine()

 End Sub

End Module

#11. 11. Repetition

Repetition is a central requirement of programming. This allows a program to complete a task many times and provide output to the screen. The program below produces the seven times table.

10.1 (\*)

Module Module1

 Sub Main()

 Dim lineNum As Integer

 lineNum = 1

 'a loop

 For lineNum = 1 To 15

 Console.WriteLine("Line {0}", lineNum)

 Next 'the end of the loop

 Console.ReadKey() 'pause so user can see

 End Sub

End Module

10.2 (2)

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

10.1a Do While Loop (ac)

Module Module1

 Sub Main()

 Dim myCounter As Integer = 0

 Do While myCounter < 10

 Console.WriteLine("The counter is {0}", myCounter)

 myCounter = myCounter + 1

 Loop

 Console.ReadLine()

 End Sub

End Module

Exercises

1. Write a program that asks for the users name and then displays it seven times on the screen
2. Writ a program that asks for a number 5 times and adds them up displaying the output each time to the screen

Figure 1 contains the pseudo-code for a program to output a sequence according to

the ‘Fizz Buzz’ counting game. Fizz Buzz is a game where if a number can be divided by three you say fizz and if it can be divided by 5 you say Buzz. If it can be divided by both three and five you say fizzbuzz.

 Figure 1

 OUTPUT "How far to count?"

 INPUT HowFar

 WHILE HowFar < 1

 OUTPUT "Not a valid number, please try again."

 INPUT HowFar

 ENDWHILE

FOR MyLoop ← 1 TO HowFar

 IF MyLoop MOD 3 = 0 AND MyLoop MOD 5 = 0

 THEN

 OUTPUT "FizzBuzz"

 ELSE

 IF MyLoop MOD 3 = 0

 THEN

 OUTPUT "Fizz"

 ELSE

 IF MyLoop MOD 5 = 0

 THEN

 OUTPUT "Buzz"

 ELSE

 OUTPUT MyLoop

 ENDIF

 ENDIF

 ENDIF

ENDFOR

What you need to do:

Write a program that implements the pseudo-code as shown in Figure 1.

Test the program by showing the result of entering a value of 18 when prompted by the

program.

Test the program by showing the result of entering a value of -1 when prompted by the

program.

Evidence that you need to provide

Include the following in your Electronic Answer Document.

Your PROGRAM SOURCE CODE for the pseudo-code in Figure 1.

 [8 marks]

SCREEN CAPTURE(S) for the tests conducted when a value of 18 is entered by

the user and when a value of -1 is entered by the user.

 [1 mark]

The main part of the program uses a FOR repetition structure.

Explain why a FOR repetition structure was chosen instead of a WHILE repetition

structure.

[1 mark]

Even though a check has been performed to make sure that the variable HowFar

is greater than 1 there could be inputs that might cause the program to terminate

unexpectedly (crash).

Provide an example of an input that might cause the program to terminate and

describe a method that could be used to prevent this.

[3 marks]

FizzBuzz answer

Module Module1

 Sub Main()

 Dim loopNum As Integer

 Dim x As Integer

 Dim mod5total As Integer

 Dim mod3total As Integer

 x = 0

 Console.WriteLine("How far to count")

 loopNum = Console.ReadLine

 If loopNum < 1 Then

 Console.WriteLine("Unacceptable Number")

 End If

 For x = 1 To loopNum

 mod5total = x Mod 5

 mod3total = x Mod 3

 If mod5total = 0 And mod3total = 0 Then

 Console.WriteLine("FizzBuzz")

 ElseIf mod5total = 0 Then

 Console.WriteLine("Fizz")

 ElseIf mod3total = 0 Then

 Console.WriteLine("Buzz")

 Else

 Console.WriteLine(x)

 End If

 Next

 Console.ReadKey() 'pause so user can see

 End Sub

End Module

Consolidation Exercises 1

For A Level Computer Science, particularly for those elements that are common with 7616 (AS level) it is important that students are comfortable with taking any input from the user and modifying the output in any way required. You should try to get to the stage where you are able to do this quickly and easily without thought. In the exam time will be an issue so you should focus on performing the tasks of the level required by the course quickly rather than building very large programs

**CE1.1 How much tax ?**

You should create a program which calculates the tax due at any level of income

Up to £10000 per annum pays no tax

Up to £45000 per annum pays 20% tax

Up to £100,000 per annum pays 40% tax

Above this level pays 45% tax

**CE1.2 Telephone bill**

Your task is to produce a telephone bill. This bill will take a list of calls and work out a cost for each call and then provide a total. The cost of calls is

Germany 0.05, France 0.03, Indonesia 0.3, USA 0.02, Manchester 0.01, Vodafone 0.07

The calls are

Germany 1.3 mins

France 5.4 mins

Indonesia 6.1 mins

Manchester 10.1 mins

Vodafone 1.8 mins

Indonesia 2.1 mins

Manchester 8.1 mins

Vodafone 1.6 mins

**CE1.3 Commission**

The task is to produce a commission statement for the phone calls in the bill above. The salesman gets a 1% commission on all calls except international calls which have a 3% commission. If over 50% of the calls in any bill are international calls then the commission on international calls rises to 5%.

**CE 1.4 Election result forecast**

You are a candidate looking to stand for parliament. You want to know which is the best seat to stand in under different situations. You are a Lib Dem Candidate and you draw your support in the proportion 2:1 from Conservatives and 1:2 from Labour. You need to work out at which level of Lib Dem support you will win in a particular seat.

The two seats had the following vote at the last General Election

Seat 1: Lib Dem 10000 votes, Conservative 30,000 votes, Labour 5000 votes

Seat 2 Lib Dem 10000 votes, Labour 20,,000 votes, Conservatives 5,000 votes

#7 12.Logical bitwise operators

Logical bitwise operators yield the result ‘True’ or ‘False’. Depending of a true or false result the flow of the program can be altered.

Module Module1

 Sub Main()

 Dim a, b, c, d, e, f, g, x, y 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.

 x = Not 23 > 14

 y = Not 23 > 67 ' The preceding statements set x to False and y to True.

 Console.WriteLine(a)

 Console.WriteLine(b)

 Console.WriteLine(c)

 Console.WriteLine(d)

 Console.WriteLine(e)

 Console.WriteLine(f)

 Console.WriteLine(x)

 Console.WriteLine(y)

 If y = True Then

 Console.WriteLine("Yes")

 Else

 Console.WriteLine("False")

 End If

 Console.ReadLine()

 End Sub

End Module

Answer CT1.1

Module Module1

 Sub Main()

 Dim mySalary As Integer

 Dim myTax As Integer

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

 mySalary = Console.ReadLine

 If mySalary < 10000 Then

 myTax = 0

 End If

 If mySalary > 9999 And mySalary < 44999 Then

 myTax = (mySalary - 10000) \* 0.2

 End If

 If mySalary > 44999 And mySalary < 99999 Then

 myTax = (mySalary - 45000) \* 0.4 + 7000

 End If

 If mySalary > 100000 Then

 myTax = (mySalary - 100000) \* 0.45 + 29000

 End If

 Console.WriteLine("The tax is {0}", myTax)

 Console.ReadLine()

 End Sub

End Module

Answer CT1.2

Module Module1

 Sub Main()

 Dim mySalary As Integer

 Dim myArea As String

 Dim totalCost As Double

 Dim myCalllength As Double

 Dim myCost As Double

 Dim x As Integer

 Dim callTotal As Double

 totalCost = 0

 For x = 1 To 10

 If myArea <> "End" Then

 Console.WriteLine("Where is the call to")

 myArea = Console.ReadLine

 End If

 If myArea <> "End" Then

 Console.WriteLine("Where is the length")

 myCalllength = Console.ReadLine

 End If

 If myArea = "Germany" Then

 myCost = 0.03

 ElseIf myArea = "France" Then

 myCost = 0.05

 ElseIf myArea = "USA" Then

 myCost = 0.02

 ElseIf myArea = "Indonesia" Then

 myCost = 0.3

 ElseIf myArea = "Vodafone" Then

 myCost = 0.07

 ElseIf myArea = "Manchester" Then

 myCost = 0.01

 ElseIf myArea = "End" Then

 myCost = 0

 End If

 callTotal = myCost \* myCalllength

 totalCost = totalCost + callTotal

 If myArea = "End" Then

 'Console.WriteLine(" ", totalCost)

 Else

 Console.WriteLine("The call cost is {0}", callTotal)

 Console.WriteLine("The current total cost is {0}", totalCost)

 End If

 Next 'the end of the loop

 Console.WriteLine("The Summary Cost is {0}", totalCost)

 Console.ReadLine()

 End Sub

End Module

Compiling Code Online

Students who have not yet got VB.net at home will need to use an online compiler to compile their code. There are issues and limitations with these systems, however they will allow homework to be completed.

Online compilers are designed for console programming, not for graphical programming. User input must be entered into the stdin box on the web page. These are minor issues and these applications will allow students to write code wherever they have an internet connection.

The site is at the following address and you must choose VB.net

<http://ideone.com/>

Exercise:

Write a program that lists all possible dice throws from two dice, there are 36 in total.

Possibility 1 is dice 1 on 1 and dice 2 on 1

Possibility 2 is dice 1 on 2 and dice 2 on 1

Possibility 2 is dice 1 on 3 and dice 2 on 1

Stretch task: list the % chance of each number 2 to 12 appearing

Module Module1

 Sub Main()

 Dim myCountdo As Integer

 Dim myCountdt As Integer

 Dim myTotal As Integer

 Dim x As Integer

 Dim sevenTotal As Integer

 x = 1

 myCountdo = 1

 myCountdt = 1

 sevenTotal = 0

 For x = 1 To 36

 myTotal = myCountdo + myCountdt

 If myTotal = 7 Then

 sevenTotal = sevenTotal + 1

 End If

 Console.WriteLine("The dice numbers are {0} - {1} - {2}", myCountdo, myCountdt, myTotal)

 myCountdo = myCountdo + 1

 If myCountdo = 7 Then

 myCountdt = myCountdt + 1

 End If

 If myCountdo = 7 Then

 myCountdo = myCountdo - 6

 End If

 Next

 Console.WriteLine("Seven Total: {0}", sevenTotal)

 Console.ReadLine()

 End Sub

End Module

Procedures and functions (ac)

A procedure is a self-contained process or set of related processes

A function is a subroutine that returns a value.

Module Module1

 Sub Main()

 Dim myNumber As Integer

 myNumber = myAdd(2, 3)

 Console.WriteLine("2 + 3 = " & myNumber)

 Console.WriteLine("4 + 6 = " & myAdd(4, 6))

 Console.ReadLine()

 End Sub

 Function myAdd(ByRef x As Integer, ByVal y As Integer)

 myAdd = x + y

 Return myAdd

 End Function

End Module

The difference the rate (byVal) should not change but the debt should so is byRef

This the Value is a fixed value but a reference only refers to the value and can change. The default is byVal so you should use byRef if the value can change.

A parameter passed **ByVal**—by value—can be changed in the new method. Its value will not be changed elsewhere. **ByRef**, by reference, means the variable location itself is copied.

Module Module1

 Sub Main()

 Dim myNumber As Integer

 Console.WriteLine("2 \* 2 \* 3 = " & myAdd(2, 2, 3))

 Console.ReadLine()

 End Sub

 Function myAdd(ByRef x As Integer, ByRef y As Integer, ByRef z As Integer)

 myAdd = x \* y \* z

 Return myAdd

 End Function

End Module

(ByRef x As Integer, ByRef y As Integer, ByRef z As Integer) are the parameters

2, 2, 3 are the arguments

Module Module1

 Sub Main()

 Dim myNumber As Integer

 'myNumber = myAdd(2, 3)

 'Console.WriteLine("2 + 3 = " & myNumber)

 Console.WriteLine("Volume of a sphere " & myAdd(1.333, 3.142, 2, 2, 2))

 Console.ReadLine()

 End Sub

 Function myAdd(ByRef v As Double, w As Double, x As Integer, ByRef y As Integer, ByRef z As Integer)

 myAdd = v \* w \* x \* y \* z

 Return myAdd

 End Function

End Module

An example of a subroutine which prints two lines

Module Module1

 Sub Main()

 WriteArguments(1, 2)

 WriteArguments(1000, 2000)

 WriteArguments(500, 100)

 Console.ReadLine()

 End Sub

 Sub WriteArguments(ByVal param1 As Integer, ByVal param2 As Integer)

 Console.WriteLine(param1)

 Console.WriteLine(param2)

 'Console.ReadLine()

 End Sub

End Module

1. Write a function for calculating the volume of a cube
2. Write a function for calculating the volume of a sphere

Exercise: Write code to check for a username using the code below

Exercise: Write code to check for a username and password

Module Module1

 Sub Main()

 Dim user As String

 Do Until user = "Ed"

 Console.WriteLine("What is username")

 user = Console.ReadLine()

 Loop

 Console.WriteLine("OK")

 Console.ReadLine()

 End Sub

End Module

Module Module1

 Sub Main()

 Dim myUserone As String

 Dim userOnepass As String

 Console.WriteLine("What is your User Name")

 myUserone = Console.ReadLine()

 Console.WriteLine("What is your Password")

 userOnepass = Console.ReadLine()

 If myUserone = "Tom" Then

 If userOnepass = "123" Then

 Console.WriteLine("OK")

 Else

 Console.WriteLine("Not OK")

 End If

 End If

 Console.ReadLine()

 End Sub

End Module

Write code to produce the number of a street name where the street number is 2 digits i.e. 21 Acacia Avenue

Stretch task: write code to produce output of the street number for any normal UK address

Module Module1

 Sub Main()

 Dim myAddress As String

 Dim checkForblank As String

 Dim myRoadnum As String

 Console.WriteLine("What is your address")

 myAddress = Console.ReadLine()

 checkForblank = myAddress.Substring(1, 1)

 If checkForblank = " " Then

 myRoadnum = myAddress.Substring(0, 1)

 Console.WriteLine(myRoadnum)

 End If

 checkForblank = myAddress.Substring(2, 1)

 If checkForblank = " " Then

 myRoadnum = myAddress.Substring(0, 2)

 Console.WriteLine(myRoadnum)

 End If

 checkForblank = myAddress.Substring(3, 1)

 If checkForblank = " " Then

 myRoadnum = myAddress.Substring(0, 3)

 Console.WriteLine(myRoadnum)

 End If

 Console.ReadLine()

 End Sub

End Module

Module Module1

 Sub Main()

 Test1()

 Test2()

 Console.ReadLine()

 End Sub

 Sub Test1()

 ' Write an integer line.

 Dim value As Integer = 7

 Console.WriteLine(value)

 ' Write a string literal line.

 End Sub

 Sub Test2()

 ' Write integer.

 Dim valueTwo As Integer = 888

 Console.WriteLine(valueTwo)

 End Sub

End Module

Task: Write code to request a user name and password using two subroutines called from the main subroutine

Stretch task: Use a string handling function to select the first three characters from within the password and accept it only if the are a number between 200 and 300

Additional stretch task: Write an additional piece of code that accepts only a password with a ‘h’ in it

Additional stretch task: Write an additional piece of code that accepts only a password with a ‘h’ and an a ‘0’ in it

Module Module1

 Sub Main()

 Test1()

 Test2()

 Console.ReadLine()

 End Sub

 Sub Test1()

 Dim myUser As String

 Console.WriteLine("Enter User Name")

 myUser = Console.ReadLine()

 If myUser = "Bob" Then

 Console.WriteLine("OK")

 Else

 Console.WriteLine("Not OK")

 End If

 End Sub

 Sub Test2()

 Dim myPass As String

 Console.WriteLine("Enter Password")

 myPass = Console.ReadLine()

 If myPass = "123password" Then

 Console.WriteLine("OK")

 Else

 Console.WriteLine("Not OK")

 End If

 End Sub

End Module

Subroutine exercise

Write code for an automated health check

Do three subroutines

The first is for a cold - advise taking cough medicine

The second is for a broken bone - advise a tip to casualty

The third is for a headache - advise headache tablets

Base code

Module Module1

 Sub Main()

 question1()

 question2()

 Console.ReadLine()

 End Sub

 Sub question1()

 Dim sickness As String

 Console.WriteLine("What is your sickness")

 sickness = Console.ReadLine()

 Console.WriteLine("You have a {0}", sickness)

 End Sub

 Sub question2()

 Dim sickness As String

 Console.WriteLine("What is your sickness")

 sickness = Console.ReadLine()

 Console.WriteLine("You have a {0}", sickness)

 End Sub

End Module

Stretch tasks

In the case of a cold advise honey and lemon if the patient is a child under 12

If the patient is Jewish check the day and if it is Saturday advise that they wait 24 hours before taking the medicine. Take the day and date in and provide them with the day and date that they should take the medicine. In the case of a broken bone search for the work broken and if it is there assume that it is a broken bone and give the relevant response

Extended stretch Write a program that allows the user to enter natural language and search for the words headache, cold or broken and give the right advice. You do not need to use subroutines for this.

**Function stretch task**

The task is to write a function that produces that tells you the number of the day of the week of a particular day. Monday is day 1. Tuesday day 2 etc.

Your code should take the day and month for any day in the year and produce an output which is a number from 1 to 7. You should write the code in a way that allows you to add a year to provide the day of the week for a particular date in a decade.

This type of function is important for pricing software when a weekend has a lower price.



Module Module1

 Dim myMonth As Integer

 Sub Main()

 Dim myNumber As Integer

 Dim newVar As Integer

 Dim myDay As Integer

 Dim aNewVar As Integer

 Console.WriteLine("What is your day")

 myDay = Console.ReadLine()

 Console.WriteLine("What is your Month")

 myMonth = Console.ReadLine()

 Console.WriteLine("The day of the week is = " & myAdd(myDay, 7))

 aNewVar = myAdd(myDay, 7)

 If aNewVar = 1 Then

 Console.WriteLine("Monday")

 End If

 Console.ReadLine()

 End Sub

 Function myAdd(ByRef x As Integer, ByVal y As Integer)

 If myMonth = 2 Then

 x = x + 31

 Else If myMonth = 3 Then

 x = x + 59

 ElseIf myMonth = 4 Then

 x = x + 90

 End If

 myAdd = x Mod y

 myAdd = myAdd + 1

 Return myAdd

 End Function

End Module

Flowchart Exercise 1: Create a program from the symbols above

Sum = x + y

End

Output sum

Input x

Input y

Start

Display error message on screen

Is

PIN correct?

Flowchart Exercise 2: Create a program from the symbols

Display Menu on screen

Put card in machine

Read details from chip

Enter

PIN

Flowchart exercise 3a: Create a flowchart to work out a taxi fare. You are not permitted to take unaccompanied minors and should reject their request for a fare. The cost of the taxi is £2.00 per mile with a minimum charge of £5.00. Give this to your neighbour and ask them to produce code based on this.

Flowchart exercise 3b: Create a flowchart to work out a taxi fare. You are not able to take wheelchairs and should reject a request for a fare from a disabled customer. The cost of the taxi is £1.00 per mile with a minimum charge of £4.00. Give this flowchart to your neighbour and ask them to produce code based on this.

**Pseudo code exercise**

1a. Students should write pseudocode for an airport checking system. You should check for three items, passport, weight of baggage and ticket. If the weight is too great you should offer the chance to pay £10 per kilo up to a maximum of £100. Above that level the passenger should not allowed any more luggage

1b. Students should write pseudocode for a passport checking/immigration system. You should check for a valid passport. You should check that the passenger has £1000 unless they are from China when they should have £2000. If the traveller is from Syria they should be not be allowed entry if they have an entry stamp for Macedonia, Turkey or Greece but in all other cases they should be allowed entry.

 Initialise

 Process

Program

 Validate

Enter

data

Capture

form

 Output

 Input

Question 2013 f4

Figure 4

OUTPUT "Player One enter your chosen number: "

INPUT NumberToGuess

WHILE NumberToGuess < 1 OR NumberToGuess > 10 DO

 OUTPUT "Not a valid choice, please enter another number: "

 INPUT NumberToGuess

ENDWHILE

Guess 🡨 0

NumberOfGuesses 🡨 0

WHILE Guess <> NumberToGuess AND NumberOfGuesses < 5 DO

 OUTPUT "Player Two have a guess: "

 INPUT Guess

 NumberOfGuesses 🡨 NumberOfGuesses + 1

ENDWHILE

IF Guess = NumberToGuess

 THEN OUTPUT "Player Two wins"

 ELSE OUTPUT "Player One wins"

Test 1

 Test that your program works correctly by conducting the following test:

* Player One enters the number 0
* Player One enters the number 11
* Player One enters the number 5
* Player Two enters a guess of 5

 Test 2

 Test that your program works correctly by conducting the following test:

- Player One enters the number 6

- Player Two enters guesses of 1, 3, 5, 7, 10

Module Module1

 Sub Main()

 Dim NumberToGuess As Integer

 Dim guess As Integer

 Dim NumberofGuesses As Integer

 Console.WriteLine("Player One enter your chosen number: ")

 NumberToGuess = Console.ReadLine()

 Do While NumberToGuess < 1 Or NumberToGuess > 10

 Console.WriteLine("Not a valid choice, please enter another number: ")

 NumberToGuess = Console.ReadLine()

 Loop

 guess = 0

 NumberOfGuesses = 0

 Do While guess <> NumberToGuess And NumberOfGuesses < 5

 Console.WriteLine("Player Two have a guess: ")

 guess = Console.ReadLine()

 NumberOfGuesses = NumberOfGuesses + 1

 Loop

 If guess = NumberToGuess Then

 Console.WriteLine("Player Two wins")

 Else : Console.WriteLine("Player One wins")

 End If

 Console.ReadLine()

 End Sub

End Module

Roulette exercise: The task is to create betting software for roulette.

Task 1: Using the code below as a bas create a system that releases a random number between 1 and 36.

The Randomize function is used to create a seed number which is between 0 and 1. In the following code 2 \* Rnd() creates the value 0, 1 or 2 when displayed as an integer or whole number. If you add a number between 1 and three then you get a number of 1 to 3.

value = (2 \* Rnd()) + 1

Module Module1

 Sub Main()

 ' Initialize the random-number generator.

 Randomize()

 ' Generate random value between 1 and 6.

 Dim value As Integer

 value = (2 \* Rnd()) + 1

 Console.WriteLine(value)

 Console.ReadLine()

 End Sub

End Module

**Additional tasks**

Once you have the basic random number system in place You should take the number to bet on, and amount bet, from the user and then check whether it matches. You should then state the winnings or loss as an amount. This should be done as a simple bet on a single number with a 36 x win.

**Further tasks**

Once this has been set up you should introduce a wider variety of betting options including red and black.

## Roulette table layout



American style layout, French single zero wheel

The cloth covered betting area on a roulette table is known as the *layout*. The layout is either single zero or double zero. The European style layout has a single zero, and the American style layout is usually a double zero. The American style roulette table with a wheel at one end is now used in most casinos. The French style table with a wheel in the centre and a layout on either side is rarely found outside of Monte Carlo.

## Types of bets

### Inside bets

 **Straight (or Single)**

a single number bet. The chip is placed entirely on the middle of a number square.

 **Split**

a bet on two adjoining numbers, either on the vertical or horizontal (as in 14-17 or 8-9). The chip is placed on the line between these numbers.

**Street**

a bet on three numbers on a single horizontal line. The chip is placed on the edge of the line of a number at the end of the line (either the left or the right, depending on the layout).

 **Corner (or Square)**

a bet on four numbers in a square layout (as in 11-12-14-15). The chip is placed at the horizontal and vertical intersection of the lines between the four numbers

**Six line (or Double Street)**

a bet on two adjoining streets, with the chip placed at the corresponding intersection, as if in between where two street bets would be placed

**Trio**

a bet on the intersecting point between 0, 1 and 2, or 0, 2 and 3 (single-zero layout only).

 **Basket (or the first four)**

(non-square corner) a bet on 0, 1, 2, and 3 (single-zero layout only).

**Top line**

a bet on 0, 00, 1, 2, and 3 (double-zero layout only). The chip is placed either at the corner of 0 and 1, or the corner of 00 and 3.

### Outside bets

 Outside bets typically have smaller payouts with better odds at winning.

 **1 to 18 (*Manque*)**

a bet on one of the first low eighteen numbers coming up.

 **19 to 36 (*Passe*)**

a bet on one of the latter high eighteen numbers coming up.

 **Red or black (*Rouge ou Noir*)**

a bet on which color the roulette wheel will show.

 **Even or odd (*Pair ou Impair*)**

a bet on an even or odd nonzero number.

 **Dozen bets**

a bet on the first (1-12, *Premiere douzaine* (P12)), second (13-24, *Moyenne douzaine* (M12)), or third group (25-36, *Dernière douzaine* (D12)) of twelve numbers.

**Column bets**

a bet on all 12 numbers on any of the three vertical lines (such as 1-4-7-10 on down to 34). The chip is placed on the space below the final number in this sequence.

Module Module1

 Sub Main()

 Dim BetNumber As Integer

 Dim BetAmount As Integer

 ' Initialize the random-number generator.

 Randomize()

 ' Generate random value between 1 and 6.

 Dim value As Integer

 Console.WriteLine("What Number")

 BetNumber = Console.ReadLine()

 value = (35 \* Rnd()) + 1

 Console.WriteLine("The value is below")

 Console.WriteLine(value)

 Console.WriteLine("What is the bet")

 BetAmount = Console.ReadLine()

 Console.WriteLine("You bet {0}", BetAmount)

 Console.ReadLine()

 If value = BetNumber Then

 BetAmount = BetAmount \* 36

 Console.WriteLine("You Win {0}", BetAmount)

 Else

 Console.WriteLine("You Lose {0}", BetAmount)

 End If

 Console.ReadLine()

 End Sub

End Module

Module Module1

 Sub Main()

 Dim countries(5) As String

 Dim myNum As Integer

 Console.WriteLine("Enter a number from 1 to 5")

 myNum = Console.ReadLine()

 countries(1) = "Scotland"

 countries(2) = "Japan"

 countries(3) = "Netherlands"

 countries(4) = "Spain"

 countries(5) = "Italy"

 Console.WriteLine("For a holiday you should go to {0}", countries(myNum))

 Console.ReadKey()

 End Sub

End Module

Tasks

1) Enter the data and run it. When you have done this change a country.

2) Write code that provides the name of a student when the seat number is input

3) Write code that allows the student in position 3 to be input by the user

4) Create a program that stores an array of 5 student names. The information should be entered by the user.

5) Write code that allows all the students to be input and then state which seat a student is in. You can do a subset of the students in the class e.g. one side of the class.

6) Write code that stores thirteen playing cards and then releases to randomly as is the case in 21 (pontoon)

7) Implement the same with a full deck of card and allow the user to draw up to five cards

Module Module1

 Sub Main()

 Dim countries(5) As String

 Dim myNum As Integer

 countries(1) = "Scotland"

 countries(2) = "Japan"

 countries(3) = "Netherlands"

 countries(4) = "Spain"

 countries(5) = "Italy"

 Console.WriteLine("Enter a country")

 countries(3) = Console.ReadLine()

 Console.WriteLine("Enter a number from one to five")

 myNum = Console.ReadLine()

 Console.WriteLine("For a holiday you should go to {0}", countries(myNum))

 Console.ReadKey()

 End Sub

End Module

Module Module1

 Sub Main()

 Dim countries(5) As String

 Dim myNum As Integer

 Dim x As Integer

 For x = 1 To 5

 Console.WriteLine("Enter a country")

 countries(x) = Console.ReadLine()

 Next

 Console.WriteLine("Enter a number from one to five")

 myNum = Console.ReadLine()

 Console.WriteLine("For a holiday you should go to {0}", countries(myNum))

 Console.ReadKey()

 End Sub

End Module

Module Module1

 Sub Main()

 Dim countries(5) As String

 Dim myNum As Integer

 Dim value As Integer

 ' Initialize the random-number generator.

 Randomize()

 ' Generate random value between 1 and 6.

 value = (4 \* Rnd()) + 1

 Console.WriteLine(value)

 myNum = value

 countries(1) = "Scotland"

 countries(2) = "Japan"

 countries(3) = "Netherlands"

 countries(4) = "Spain"

 countries(5) = "Italy"

 'Console.WriteLine("Enter a number from one to five")

 ' myNum = Console.ReadLine()

 Console.WriteLine("The country is {0}", countries(myNum))

 Console.ReadKey()

 End Sub

End Module

Task: Write a program that releases a random playing card from a full deck

**Multi Dimensional Array Example**

Module Module1

 Sub Main()

 Dim countries(5, 5) As String

 Dim myNum As Integer

 Dim value As Integer

 ' Initialize the random-number generator.

 Randomize()

 ' Generate random value between 1 and 6.

 value = (4 \* Rnd()) + 1

 Console.WriteLine(value)

 myNum = value

 countries(1, 1) = "Scotland"

 countries(2, 1) = "Japan"

 countries(3, 1) = "Netherlands"

 countries(4, 1) = "Spain"

 countries(5, 1) = "Italy"

 countries(1, 2) = "6m"

 countries(2, 2) = "12m"

 countries(3, 2) = "2m"

 countries(4, 2) = "1m"

 countries(5, 2) = "55m"

 'Console.WriteLine("Enter a number from one to five")

 ' myNum = Console.ReadLine()

 Console.WriteLine("The country is {0}", countries(myNum, 1))

 Console.WriteLine("The country is {0}", countries(myNum, 2))

 Console.ReadKey()

 End Sub

End Module

Task: Write a computer program that allows you to input the names of the countries of the European Union and display their populations.

Task: Write a program that takes student scores into an array and allows you to access them.

(#18a) Reading from a text file (2)

Reading from a text file is a useful facility for a computer system. Data can be stored in a text file. When the program starts it can be stored in an array. From there the data can be used in a similar way to it being stored in a database. In the case of telephone call records it would be possible to read the data from a set of call records and produce a bill for a customer.

The following code reads from a text file

Module Module1

 Sub Main()

 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("C:\Users\jbarwick3.319\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()

 End Sub

End Module

Task1: Create a program that reads the student’s names from a text file and displays them onto the screen

Task2: Create a program that takes student names from a text file and stores them into an array. When the information is in the array print the student name to the screen.

The following code reads from a text file into an array. You will need to produce a text file called diary.txt – which is below.

one

two

three

four

five

It contains the numbers 1 to 5 which are displayed on the screen when the 1,2,3, 4 or 5 are pressed.

Module Module1

 Sub Main()

 Dim objStreamReader As IO.StreamReader

 Dim strLine As String

 Dim numbers(5) As String

 Dim count As Integer

 Dim DisplayNumber As Integer

 'Pass the file path and the file name to the StreamReader constructor.

 objStreamReader = New IO.StreamReader("C:\Users\yourname\diary.txt")

 'Read the first line of text.

 strLine = objStreamReader.ReadLine

 'Continue to read until you reach the end of the file.

 count = 0

 Do While Not strLine Is Nothing

 count = count + 1

 numbers(count) = strLine

 'Write the line to the Console window.

 Console.WriteLine(strLine)

 'Read the next line.

 strLine = objStreamReader.ReadLine

 Loop

 'Close the file.

 Console.WriteLine("Type a number")

 DisplayNumber = Console.ReadLine()

 DisplayNumber = DisplayNumber

 Console.WriteLine(numbers(DisplayNumber))

 Console.ReadLine()

 objStreamReader.Close()

 Console.ReadLine()

 End Sub

End Module

(#18b) The following code writes to a text file (2)

Module Module1

 Sub Main()

 Dim objStreamWriter As IO.StreamWriter

 'Pass the file path and the file name to the StreamWriter constructor.

 objStreamWriter = New IO.StreamWriter("C:\Users\Jbarwick\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()

 End Sub

End Module

Task: Write a program that asks for 5 names and then writes them to a text file

Task: Write a program that asks for the names of countries in the European Union and writes them to a text file.

**Roulette Array Task**

Task 1: The first task is to write a program that places a set of roulette wheel spins into an array. This should be done automatically from a file with 5 numbers. You should write code to display the numbers in the array that will all be between 1 and 36

Task 2: The second task is to write a computer program that allows a user to bet on a number between 1 and 36 and displays a win of 36 times the amount if they win.

Task 3: The third task is to integrate these two programs so that you run the number of the wheel spin and that number is used to determine whether there is a win and the amount.

Module Module1

 Sub Main()

 Dim objStreamReader As IO.StreamReader

 Dim strLine As String

 Dim numbers(5) As String

 Dim count As Integer

 Dim DisplayNumber As Integer

 Dim bet As Integer

 Dim roulettenumber As Integer

 Dim win As Integer

 Dim amountbet As Integer

 'Pass the file path and the file name to the StreamReader constructor.

 objStreamReader = New IO.StreamReader("C:\Users\edwardjoyce\spin.txt")

 'Read the first line of text.

 strLine = objStreamReader.ReadLine

 'Continue to read until you reach the end of the file.

 count = 0

 Do While Not strLine Is Nothing

 count = count + 1

 numbers(count) = strLine

 'Write the line to the Console window.

 Console.WriteLine(strLine)

 'Read the next line.

 strLine = objStreamReader.ReadLine

 Loop

 'Close the file.

 For x = 1 To 5

 Console.WriteLine("Type the numebr of the spin")

 DisplayNumber = Console.ReadLine()

 Console.WriteLine(numbers(DisplayNumber))

 roulettenumber = numbers(DisplayNumber)

 Console.WriteLine("Type a number bet on")

 bet = Console.ReadLine()

 Console.WriteLine("Type an amount bet")

 amountbet = Console.ReadLine()

 Console.WriteLine("Displaynum ie the nuber on the roulette wheel is {0}", roulettenumber)

 Console.WriteLine("Number you bet on is {0} ", bet)

 Console.WriteLine("Bet is {0}", amountbet)

 win = 36 \* amountbet

 If bet = roulettenumber Then

 Console.WriteLine("You Win {0}", win)

 Else

 Console.WriteLine("You Lose")

 End If

 Next

 objStreamReader.Close()

 Console.ReadLine()

 End Sub

End Module

**String to integer conversion**

Students of VB.Net sometimes request string to integer conversions. The following code shows how this is done using CInt

Module Module1

 Sub Main()

 Dim myNumber As String

 Dim grounded As String

 Dim mytext As Double

 myNumber = 1

 grounded = 1

 mytext = myNumber + grounded

 Console.WriteLine("Numis {0}", mytext)

 Dim agrounded As Integer = CInt(grounded)

 Dim amyNumber As Integer = CInt(myNumber)

 Dim amytext As Double

 ' Convert.ToDouble(grounded)

 'Convert.ToDouble(mytext)

 amytext = amyNumber + agrounded

 Console.WriteLine("Numis {0}", amytext)

 Console.ReadLine()

 End Sub

End Module

**Task: In the gap explain each line of this code**

Module Module1

 Sub Main()

 Dim objStreamReader As IO.StreamReader

 Dim strLine As String

 objStreamReader = New IO.StreamReader("C:\Users\yourname\diary.txt")

 strLine = objStreamReader.ReadLine

 Do While Not strLine Is Nothing

 Console.WriteLine(strLine)

 strLine = objStreamReader.ReadLine

 Loop

 objStreamReader.Close()

 Console.ReadLine()

 End Sub

End Module

Task BD (a) Binary to denary converter program

The task is to write a program that converts from binary to denary and the reverse. You should begin by writing cost that converts and four digit binary number into a denary number. Then you should then extend the code to convert a binary number of up to eight digits into a denary number

Once you have completed this you should look at how to deal with the binary numbers of between one and eight digits and modify the program so that it calculates the length of the string that is input and then provides the conversion. Finally you should convert numbers that are above 256. Ultimately you should have software that provides a conversion between binary and denary

Task BD (b) Denary to binary conversion program

The task is to write a program that converts from denary to binary. You should begin by writing cost that converts a number up to 16 into a binary number. Then you extend the code to deal with a number up to 255. Once you have done this you should extend the code to deal with larger numbers.

Module Module1

 Sub Main()

 Dim MyBinnum As String

 Dim MyDenary As Integer

 Dim BinArray(7) As Integer

 Console.WriteLine("Enter a number in binary")

 MyBinnum = Console.ReadLine()

 Console.WriteLine(MyBinnum.Length)

 Console.ReadLine()

 BinArray(0) = (MyBinnum.Substring(0, 1))

 Console.WriteLine(BinArray(0))

 BinArray(1) = (MyBinnum.Substring(1, 1))

 Console.WriteLine(BinArray(1))

 BinArray(2) = (MyBinnum.Substring(2, 1))

 Console.WriteLine(BinArray(2))

 BinArray(3) = (MyBinnum.Substring(3, 1))

 Console.WriteLine(BinArray(3))

 BinArray(4) = (MyBinnum.Substring(4, 1))

 Console.WriteLine(BinArray(4))

 BinArray(5) = (MyBinnum.Substring(5, 1))

 Console.WriteLine(BinArray(5))

 BinArray(6) = (MyBinnum.Substring(6, 1))

 Console.WriteLine(BinArray(6))

 BinArray(7) = (MyBinnum.Substring(7, 1))

 Console.WriteLine(BinArray(7))

 Console.ReadLine()

 MyDenary = (BinArray(0) \* 128) + (BinArray(1) \* 64) + (BinArray(2) \* 32) + (BinArray(3) \* 16) + (BinArray(4) \* 8) + (BinArray(5) \* 4) + (BinArray(6) \* 2) + (BinArray(7) \* 1)

 Console.WriteLine(MyDenary)

 Console.ReadLine()

 End Sub

End Module

The Caesar Cipher

The Caesar cipher is a basic substitution cipher where one character is converted into another. Mid is required, illustrated below

Dim TestString As String = "Mid Function Demo"

' Returns "Mid".

Dim FirstWord As String = Mid(TestString, 1, 3)

' Returns "Demo".

Dim LastWord As String = Mid(TestString, 14, 4)

' Returns "Function Demo".

Dim MidWords As String = Mid(TestString, 5)

Module Module1

 Sub Main()

 Dim myText As String

 Dim a As String

 Dim count As Integer

 Dim longString As String

 longString = "Text: "

 count = 1

 Console.WriteLine("What is the text to encrypt")

 myText = Console.ReadLine

 For x = 1 To 9

 a = Mid(myText, count, 1)

 count = count + 1

 If a = "a" Then

 a = "z"

 ElseIf a = "b" Then

 a = "a"

 ElseIf a = "c" Then

 a = "b"

 ElseIf a = "d" Then

 a = "c"

 ElseIf a = "e" Then

 a = "d"

 ElseIf a = "f" Then

 a = "e"

 ElseIf a = "g" Then

 a = "f"

 ElseIf a = "h" Then

 a = "g"

 ElseIf a = "i" Then

 a = "h"

 ElseIf a = "j" Then

 a = "i"

 ElseIf a = "k" Then

 a = "j"

 ElseIf a = "l" Then

 a = "k"

 ElseIf a = "m" Then

 a = "l"

 ElseIf a = "n" Then

 a = "m"

 ElseIf a = "o" Then

 a = "n"

 ElseIf a = "p" Then

 a = "o"

 ElseIf a = "q" Then

 a = "p"

 ElseIf a = "r" Then

 a = "q"

 ElseIf a = "s" Then

 a = "r"

 ElseIf a = "t" Then

 a = "s"

 ElseIf a = "u" Then

 a = "t"

 ElseIf a = "v" Then

 a = "u"

 ElseIf a = "w" Then

 a = "v"

 ElseIf a = "x" Then

 a = "w"

 ElseIf a = "y" Then

 a = "x"

 ElseIf a = "z" Then

 a = "y"

 ElseIf a = " " Then

 a = " "

 End If

 Console.WriteLine("The converted character is {0}", a)

 longString = longString + a

 Next

 Console.WriteLine(longString)

 Console.ReadLine()

 End Sub

End Module

**Additional tasks**

Noughts and crosses

Connect Four

Battleships

**Frequency analysis code**

Module Module1

 Sub Main()

 Dim myText As String

 Dim a As String

 Dim count As Integer

 Dim longString As String

 Dim acount, bcount, ccount, dcount, ecount, fcount, gcount, hcount As Integer

 Dim icount, jcount, kcount, lcount, mcount, ncount, ocount, pcount As Integer

 Dim qcount, rcount, scount, tcount, ucount, vcount, wcount, xcount, ycount, zcount As Integer

 longString = "Text: "

 count = 1

 Console.WriteLine("What is the text to encrypt")

 myText = Console.ReadLine

 For x = 1 To 500

 a = Mid(myText, count, 1)

 count = count + 1

 If a = "a" Then

 acount = acount + 1

 ElseIf a = "b" Then

 bcount = bcount + 1

 ElseIf a = "c" Then

 ccount = ccount + 1

 ElseIf a = "d" Then

 dcount = dcount + 1

 ElseIf a = "e" Then

 ecount = ecount + 1

 ElseIf a = "f" Then

 fcount = fcount + 1

 ElseIf a = "g" Then

 gcount = gcount + 1

 ElseIf a = "h" Then

 hcount = hcount + 1

 ElseIf a = "i" Then

 icount = icount + 1

 ElseIf a = "j" Then

 jcount = jcount + 1

 ElseIf a = "k" Then

 kcount = kcount + 1

 ElseIf a = "l" Then

 lcount = lcount + 1

 ElseIf a = "m" Then

 mcount = mcount + 1

 ElseIf a = "n" Then

 ncount = ncount + 1

 ElseIf a = "o" Then

 ocount = ocount + 1

 ElseIf a = "p" Then

 pcount = pcount + 1

 ElseIf a = "q" Then

 qcount = qcount + 1

 ElseIf a = "r" Then

 rcount = rcount + 1

 ElseIf a = "s" Then

 scount = scount + 1

 ElseIf a = "t" Then

 tcount = tcount + 1

 ElseIf a = "u" Then

 ucount = ucount + 1

 ElseIf a = "v" Then

 vcount = vcount + 1

 ElseIf a = "w" Then

 wcount = wcount + 1

 ElseIf a = "x" Then

 xcount = xcount + 1

 ElseIf a = "y" Then

 ycount = ycount + 1

 ElseIf a = "z" Then

 zcount = zcount + 1

 ElseIf a = " " Then

 a = " "

 End If

 'longString = longString + a

 Next

 'Console.WriteLine(longString)

 Console.WriteLine("The number of a's is {0}", acount)

 Console.WriteLine("The number of b's is {0}", bcount)

 Console.WriteLine("The number of c's is {0}", ccount)

 Console.WriteLine("The number of d's is {0}", dcount)

 Console.WriteLine("The number of e's is {0}", ecount)

 Console.WriteLine("The number of f's is {0}", fcount)

 Console.WriteLine("The number of g's is {0}", gcount)

 Console.WriteLine("The number of h's is {0}", hcount)

 Console.WriteLine("The number of i's is {0}", icount)

 Console.WriteLine("The number of j's is {0}", jcount)

 Console.WriteLine("The number of k's is {0}", kcount)

 Console.WriteLine("The number of l's is {0}", lcount)

 Console.WriteLine("The number of m's is {0}", mcount)

 Console.WriteLine("The number of n's is {0}", ncount)

 Console.WriteLine("The number of o's is {0}", ocount)

 Console.WriteLine("The number of p's is {0}", pcount)

 Console.WriteLine("The number of q's is {0}", qcount)

 Console.WriteLine("The number of r's is {0}", rcount)

 Console.WriteLine("The number of s's is {0}", scount)

 Console.WriteLine("The number of t's is {0}", tcount)

 Console.WriteLine("The number of u's is {0}", ucount)

 Console.WriteLine("The number of v's is {0}", vcount)

 Console.WriteLine("The number of w's is {0}", wcount)

 Console.WriteLine("The number of x's is {0}", xcount)

 Console.WriteLine("The number of y's is {0}", ycount)

 Console.WriteLine("The number of z's is {0}", zcount)

 Console.ReadLine()

 End Sub

End Module

**Caesar Cipher using an array**

Module Module1

 Sub Main()

 Dim myLetter As String

 Dim alphaNumber(26, 2) As String

 Dim myText As String

 Dim count As Integer

 Dim chosenLetter As String

 Dim chosenPosition As Integer

 Dim givenLetter As Char

 myText = "abcdefghijklmnopqrstuvwxyz"

 count = 0

 For x = 1 To 26

 count = count + 1

 myLetter = Mid(myText, count, 1)

 'Console.WriteLine("My letter is {0}", myLetter)

 alphaNumber(count, 1) = myLetter

 Next

 Console.WriteLine("Enter a character")

 givenLetter = Console.ReadLine()

 count = 0

 For x = 1 To 26

 count = count + 1

 If givenLetter = alphaNumber(count, 1) Then

 Console.WriteLine("The number asociated with this letter is {0}", count)

 Console.WriteLine("The Ceaser substitution is {0}", alphaNumber(count + 1, 1))

 'indentation above changed to fit on line

 End If

 Next

 Console.ReadLine()

 End Sub

End Module

**Homework: Breaking a code**

Exercise: The task is to produce computer code that will assist with breaking any encrypted code that uses a Caeser cipher. There are a number of levels to this task. You should select the level you wish to complete and then seek to create working code for that level. The first task

The task relates to taking a piece of encrypted code and crack it. An example would be the second line below

i have the encrypted code

j ibwf uif fodszqufe dpef

You have to create tools to help crack the code. You can create your own encrypted code which is longer, however the above code should be sufficient

Level 1

You should create a piece of code to count the number of time every letter appears in a piece of text. You can read the code provided but should build your own version from scratch.

Level 2

You should write code that does level 1 but also looks for i and a in the text and then uses that to determine the shift.

Level 3

You should write code that does the task at level 1 and 2 and then looks for another word, for example ‘the’ to determine the shift between the two options from i or a.

Level 4

Your code should take any code encrypted using a Caesar cipher, no matter how many letters shifted and decrypt it.

**Section B**

The algorithm, represented below using the pseudo code in Figure 4, describes code to determine the winner of a game of chance playing against a computer. The user has to choose 1 or 2. The system will produce a random outcome. If the user gains correct scores more than half the time they win, otherwise the system wins

**Figure 4**

OUTPUT "You have 5 chances to play the system. If you match the system choice three times you will win"

WHILE COUNT < 5

OUTPUT "Enter your first choice: 1 or 2"

INPUT UserChoice

IF ChoiceOnetwo < 1 OR ChoiceOnetwo > 2 Then

OUTPUT "Not a valid choice, choose again"

COUNT = COUNT -1

INPUT UserChoice

END IF

SystemChoice <-- random number either one or two

IF UserChoice = SystemChoice THEN

 PlayerScore = Score + 1

ELSE SystemScore = SystemScore + 1

END IF

COUNT + COUNT + 1

END IF

LOOP

IF SystemScore > PlayerScore Then

OUTPUT "The Computer Wins"

ELSE OUTPUT "You Win !"

END IF

What you need to do

 Write a program for the above algorithm

 Test 1 test with the numbers 1, 2, 1, 2, 1

 Test 2 test with the numbers 1,2,3,3,3,3,1,2,1

 Save the program to a Word file

 Modify the program to give the user the choice of Red or Black rather than 1 or 2

**Evidence that you must provide**

Your program source code (13 marks)

Screen capture showing the result of test 1 (2 marks)

Screen capture showing the result of test 2 (2 marks)

Modified code to give the user the choice of Red or Black rather than 1 or 2 (3 marks)

**Programming challenge: Calculate the date of Easter Sunday**

Easter falls on the Sunday following the first full moon following the 21st March. The next full moon is on Jan 24th at 01.46. There are on average 29.53 days between full moons. The task is to calculate the date of Easter this year.

Your program must calculate the date of all full moons between today and Easter and display them

**Stretch task**

Produce a calendar to show Easter for every year for any year input. The program should show the dates of full moons used to calculate this. The date of the full moon must be calculated by the system and not input from a table.

Programming Challenge: Cashpoint

The task is to write code for a cashpoint system.

The customers starting balance is £253.17

Screen 1: Please enter your PIN

The code needs to take a PIN code and then offer two choices. The first is a balance. The second is the option to withdraw money.

The customer is given a maximum of three attempts to withdraw. If they fail to enter a correct PIN they are refused and cannot come back and receive money

Screen 2:

Select the option require

1. Withdraw funds

2. Balance

If the funds are withdrawn the balance diminishes. If the customer attempts to withdraw more funds that they have the system should only issue the money available.

The cashpoint only has £10 and £20 notes

Stretch task

The stretch task is to have a cashpoint with multiple customers. There should be 5 customers with the following balances: 17.25, 1045.25, -103.46, 23,542.18, 1.54

**Using Mod**

Mod is used to produce remainders after division: 7 Mod 2 produces 1

Type in the following code to understand how Mod works

Module Module1

 Sub Main()

 Dim TestNumber As Integer

 TestNumber = 7 Mod 2

 Console.WriteLine(TestNumber)

 Console.ReadLine()

 End Sub

End Module

Using Mod to find the first day of the week

The first day of 2016 was Friday January 1st 2016

1st Feb was the 32 day of the year. You can use Mod to work out the number of days after Friday that it was. If you use the following code you can see that it is the 4th day after Friday, i.e. a Monday.

Module Module1

 Sub Main()

 Dim TestNumber As Integer

 TestNumber = 32 Mod 7

 Console.WriteLine(TestNumber)

 Console.ReadLine()

 End Sub

End Module

Module Module1

 Sub Main()

 Dim DayofMonth As Integer

 Dim Month As Integer

 Console.WriteLine("What is the Month")

 Month = Console.ReadLine()

 Console.WriteLine("What is the Day in the Month")

 DayofMonth = Console.ReadLine()

 If Month = 1 Then

 DayofMonth = DayofMonth Mod 7

 Console.WriteLine(DayofMonth)

 Console.ReadLine()

 End If

 If Month = 2 Then

 DayofMonth = DayofMonth + 31

 DayofMonth = DayofMonth Mod 7

 Console.WriteLine(DayofMonth)

 Console.ReadLine()

 End If

 End Sub

End Module

**Task: produce software which will give the day of the week for any day in 2016 when the month and the day are entered. Then extend this for any year in the next decade.**