Government Science College, Valod

Second Year B.Sc. (Computer Science) Sem-3

Subject: VB.NET

Unit-2

Practical Sheet-I

1. **Program to demonstrate the usage of option Explicit option.**

```
Option Explicit Off
Imports System.Console
Module Module1
    Sub Main()
        x = 10
        y = 10
        WriteLine("Sum : " & (x + y))
        ReadLine()
    End Sub
End Module
```

Run the program and note output of program.

Now, change Option Explicit Off to on and observe the behavior of program. Try to Run the Program.

Note down your conclusion.

2. **Program to demonstrate the usage of Imports statement and Option Compare option.**

```
Imports System.Console
Module Module1
    Dim x As String
    Dim y As String
    Sub Main()
        x = "GSCV"
        y = "Gscv"
        WriteLine("Textual Compare : " & (x = y))
        ReadLine()
    End Sub
End Module
```

Run the program and note output of program.

Now, insert Option Compare Text above Imports System.Console and Run the program and note down conclusion.

3. **Program to demonstrate the usage of Option Strict option for type conversion.**

```
Imports System.Console
Module Module1
    Dim x As Integer
    Dim y As Single = 12.55
    Sub Main()
        x = y
        WriteLine("Textual Compare : " & x)
        ReadLine()
    End Sub
End Module
```

Run the program and note output of program.

Now, insert Option Strict On above the Imports Statement and observe the behavior. Try to Run the Program.

Note down your conclusion.

4. Program to create and use of Enumerations.

```
Module Enumeration
    Enum Days
        Sunday = 1
        Monday = 2
        Tuesday = 3
        Wednesday = 4
        Thursday = 5
        Friday = 6
        Saturday = 7
    End Enum
    Sub Main()
        System.Console.WriteLine("Friday is day " & Days.Friday)
    End Sub
End Module
```

Run the Program and note down your output.

Create a program that creates the Enumeration of Months of Year and print the number of days for particular months.
Enum Name: Year values January = 31 February = 28 ……

5. Program that demonstrates the default values of all types of data types.

```
Imports System.Console
Module DataTypes
    Sub main()
        Dim i As Integer
        Dim bt As Byte
        Dim s As Short
        Dim l As Long
        Dim c As Char
        Dim bool As Boolean
        Dim si As Single
        Dim d As Double
        Dim dec As Decimal
        Dim dt As Date
        Dim obj As Object
        Dim str As String
        WriteLine("Default Values of various Data types")
        WriteLine("Integer : " & i)
        WriteLine("Byte : " & bt)
        WriteLine("Short : " & s)
        WriteLine("Long : " & l)
        WriteLine("Char : " & c)
        WriteLine("Boolean : " & bool)
        WriteLine("Single : " & si)
        WriteLine("Double : " & d)
        WriteLine("Decimal : " & dec)
        WriteLine("Date : " & dt)
        WriteLine("Object : " & obj)
        WriteLine("String : " & str)
        ReadKey()
    End Sub
End Module
```

Run the Program and note down your output.

6. Program to demonstrate the use of conversion function.

```
Option Strict On
Imports System.Console
Module Conversion
    Sub Main()
        Dim dblData As Double
        Dim intData As Integer
        dblData = 3.14159
        intData = dblData
        WriteLine("intData = " & Str(intData))
        ReadKey()
    End Sub
End Module
```

Run the Program and note your output.

Now, Replace intData = dblData statement with intData = CInt(dblData) observe your output.

Note down your Conclusion.

7. Program to demonstrate the use of Conversion functions.

```
Option Strict On
Imports System.Console
Module Conversion
    Sub Main()
        Dim Number As String = "123456789.987654321"
        Dim intNum As Integer
        Dim bt As Byte
        Dim st As Short
        Dim dec As Decimal
        Dim sngNum As Single
        Dim dblNum As Double
        Dim lngNum As Long
        Dim bool As Boolean
        intNum = CInt(Number)
        'bt = CByte(Number)
        'st = CShort(Number)
        dec = CDec(Number)
        sngNum = CSng(Number)
        dblNum = CDbl(Number)
        lngNum = CLng(Number)
        bool = CBool(Number)
        WriteLine("Byte No. " & Str(bt))
        WriteLine("Short No. : " & Str(st))
        WriteLine("Integer No. : " & Str(intNum))
        WriteLine("Decimal No. : " & Str(dec))
        WriteLine("Single No. : " & Str(sngNum))
        WriteLine("Double No. : " & Str(dblNum))
        WriteLine("Long No. : " & Str(lngNum))
        WriteLine("Boolean. : " & Str(bool))
        ReadKey()
    End Sub
End Module
```

Run the Program and note down your output.

Now, Remove comment of highlighted lines and try to run the program, observe the behavior of program.

Try out Converting Number to Date, Object, too.

Note down your conclusion.

8. Program to demonstrate the use of CType() and Str() function.

```
Option Strict On
Imports System.Console
Module Conversion
    Sub Main()
        Dim Number As String = "123456789.987654321"
        Dim intNum As Integer
        Dim bt As Byte
        Dim st As Short
        Dim dec As Decimal
        Dim sngNum As Single
        Dim dblNum As Double
        Dim lngNum As Long
        Dim bool As Boolean

        intNum = CType(Number, Integer)
        dec = CType(Number, Decimal)
        sngNum = CType(Number, Single)
        dblNum = CType(Number, Double)
        lngNum = CType(Number, Long)
        bool = CType(Number, Boolean)
        WriteLine("Byte No. " & Str(bt))
        WriteLine("Short No. : " & Str(st))
        WriteLine("Integer No. : " & Str(intNum))
        WriteLine("Decimal No. : " & Str(dec))
        WriteLine("Single No. : " & Str(sngNum))
        WriteLine("Double No. : " & Str(dblNum))
        WriteLine("Long No. : " & Str(lngNum))
        WriteLine("Boolean. : " & Str(bool))
        ReadKey()
    End Sub
End Module
```

Run the Program and note down your output.

Try out Converting Number to Date, Object, too.

Note down your conclusion.

9. Program to demonstrate the use Asc() and Chr() function to convert character to character code and vice-versa.

```
Imports System.Console
Module DataConversion
    Sub main()
```

```
        Dim chr1 As Char = "A"
        Dim chr2 As Char = "a"
        Dim intChar As Integer = 66
        WriteLine("ASCII value of A : " & Asc(chr1))
        WriteLine("ASCII value of A : " & Asc(chr2))
        WriteLine("Character Whose ASCII code 66 is : " & Chr(intChar))
        WriteLine("Character Whose ASCII code 55 is : " & Chr(55))
        ReadKey()
    End Sub
End Module
```

Run the Program and note down your output.

Try out to find out ASCII code for other characters using Asc(). Also find characters for specific Number using chr().

10. Program to demonstrate the use of Val(), UCase(), LCase() function of String.

```
Imports System.Console
Module DataConversion
    Sub main()
        Dim st As String = "Government Science College, Valod"
        WriteLine("String in Upper Case : " & UCase(st))
        WriteLine("String in Lower Case : " & LCase(st))

        Dim strNum As String = "12345"
        Dim intNum As Integer = Val(strNum)
        WriteLine("Number in Integer : " & intNum)
        intNum = intNum + 10
        Dim strNum2 As String = Str(intNum)
        WriteLine("Number in String : " & strNum2)
        strNum2 = strNum2 + "Hello"
        WriteLine("Number in String : " & strNum2)
        ReadKey()
    End Sub
End Module
```

11. Program to demonstrate the use of functions that check for data types like IsDate(), IsNumeric(), etc.

```
Imports System.Console
Module CheckDataType
    Sub main()
        Dim testVar As Object
        WriteLine("testVar is Nothing ?: " & IsNothing(testVar))
        testVar = "53"
        ' The following call to IsNumeric returns True.
        WriteLine("53 is Numeric ?: " & IsNumeric(testVar))
        testVar = "459.95"
        ' The following call to IsNumeric returns True.
        WriteLine("459.95 is Numeric ?: " & IsNumeric(testVar))
        testVar = "45 Help"
        ' The following call to IsNumeric returns False.
        WriteLine("45 Help is Numeric ?: " & IsNumeric(testVar))
        testVar = {1, 2, 3}
        ' The following call to IsArray returns True.
        WriteLine("{1, 2, 3} is Array ?: " & IsArray(testVar))
        'Assign current date to testVar
        testVar = Date.Today
        'Display Current Date
        WriteLine("Todays's Date : " & testVar)
        'The Following call to isDate returns True.
        WriteLine(testVar & " is Date ?: " & IsDate(testVar))
        'The Following call to isReference returns False.
        WriteLine(testVar & " is Reference ?: " & IsReference(testVar))
        'The Following call to isReference returns False.
        'Check whether testVar has some value or not.
        WriteLine(testVar & " is Nothing ?: " & IsNothing(testVar))
        ReadKey()
    End Sub
End Module
```

12. **Program to demonstrate the use of If…Else Statement.**

```
Imports System.Console
Module Conditionals
    Dim intNum As Integer = 49
    Sub main()
        If intNum > 50 Then
            WriteLine("No. is Greater than 50.")
        Else
            WriteLine("No. is Less than 50.")
        End If
        ReadKey()
    End Sub
End Module
```

Note down the output.

Change the value of intNum to 50 and observe the output whether it is true or not.

Make Correction according that display correct output.

13. **Program to demonstrate the use of If..ElseIf..Else Statement.**

```
Imports System.Console
Module Conditionals
    Dim intNum As Integer = 101
    Sub main()
        If intNum > 50 Then
            WriteLine("No. is Greater than 50.")
        ElseIf intNum > 100 Then
            WriteLine("No. is Greater than 100.")
        End If
        ReadKey()
    End Sub
End Module
```

Note down the output.

Make Currection according that display correct output.

14. **Program to demonstrate the use of Logical Operators.**

```
Imports System.Console
Module Conditionals
    Dim a As Integer = 51
    Dim b As Integer = 101
    Sub main()
        If a > 50 And b > 50 Then
            'Executes When both conditions are true simulataneously
            'Display Output as both a and b have values greater than 50
            WriteLine("Both a and b are above 50 ")
        Else
            'Executes When Both Condition is false or one of them is true, but
not both
            WriteLine("Both a and b are above not above 50, but posibility for
one of them ")
        End If
        If a > 100 And b > 100 Then
            WriteLine("Both a and b are above 100 ")
        Else
            'Display Output, beacuse a is 51 while b is 101, so only b is
greater than 100
            WriteLine("Both a and b are above not above 100, but posibility
for one of them ")
        End If
        If a > 100 Or b > 100 Then
            'Excutes when one of conditions are true
            WriteLine("Both a and b are above 100 or one of them")
        Else
            'Executes when all conditions are false
            WriteLine("Both a and b are below 100")
        End If
        If Not a > 50 Then
            WriteLine("a is less than 50")
        Else
            WriteLine("a is greater than 50")
        End If
        ReadKey()
    End Sub
End Module
```

Note down the output.

Try out to change the values of a and b to understand the behavior of IF..Else and Logical operators like And, Or and Not.

Note down your observations.

15. Program to demonstrate the use of Readline() function to take input from user.

```
Imports System.Console
Module Conditionals
    Sub main()
        Dim RollNo As Integer
        Dim Name As String
        Dim Percentage As Single
        Dim dob As String
        Write("Enter Your Roll No. : ")
        RollNo = Integer.Parse(ReadLine())
        Write("Enter Your Name : ")
        Name = ReadLine()
        Write("Enter Your Percentage : ")
        Percentage = Single.Parse(ReadLine())
        Write("Enter Your Date of Birth (DD/MM/YYYY) :")
        dob = ReadLine()
        WriteLine("Roll No. : " & RollNo)
        WriteLine("Name : " & Name)
        WriteLine("Percentage : " & Percentage)
        Dim dtDob As Date = Date.Parse(dob)
        WriteLine("Date of Birth : " & dtDob.ToString("dd/MM/yyyy"))

        ReadKey()
    End Sub
End Module
```

Write a program to find total, percentage and class of student from Marks of five different Subjects. Input marks from user.

16. Program to demonstrate the use of Select..Case Statement.

```
Imports System.Console
Module Conditionals
    Sub main()
        Dim choice As Integer
        Dim num1 As Integer
        Dim num2 As Integer
        WriteLine("Enter Number 1 : ")
        num1 = Integer.Parse(ReadLine())
        WriteLine("Enter Number 2 : ")
        num2 = Integer.Parse(ReadLine())

        WriteLine("1. Addition of Two Numbers.")
        WriteLine("2. Subtraction of Two Numbers.")
        WriteLine("3. Multiplication of Two Numbers.")
        WriteLine("4. Division of Two Numbers.")
        WriteLine("5. Integer Division of Two Numbers.")
        WriteLine("6. Modulus(Remainder) of Two Numbers.")
        Write("Enter Your Choice : ")
        choice = Integer.Parse(ReadLine())
        Select Case choice
            Case 1
                WriteLine("Addition of Two Numbers : " & (num1 + num2))
            Case 2
                WriteLine("Subtration of Two Numbers : " & (num1 - num2))
            Case 3
                WriteLine("Multiplication of Two Numbers : " & (num1 * num2))
            Case 4
                WriteLine("Division of Two Numbers : " & (num1 / num2))
            Case 5
                WriteLine("Integer Division of Two Numbers : " & (num1 \ num2))
            Case 6
                WriteLine("Modulus(Remainder) of Two Numbers : " & (num1 Mod num2))
            Case Else
                WriteLine("Invalid Choice. Please Enter choice between 1 to 6.")
        End Select
        ReadKey()
    End Sub
End Module
```

Run the Program and try out all the choice from 1 to 6. Also check choice as 7.

17. Program to demonstrate the use of While Loop.

```
Imports System.Console
Module loops
    Sub main()
        Dim no As Integer = 10
        Dim counter As Integer = 0
        While counter < no
            Write(" " & counter)
            counter = counter + 1
        End While
        ReadKey()
    End Sub
End Module
```

Run the Program and understand the behavior of While loop.

If you notice that it doesn’t print last number 10, make necessary correction for it.

Accept no and counter from user as 50 and 10 respectively and run the program.

Now, make counter = counter + 1 as counter = counter + 2, note down the output.

Write a Program that print series from 1 to 10 in reverse order.

Write a Program to print odd numbers from 1 to 50.

Write a Program to print Even numbers from 1 to 50.

18. Program to demonstrate the use of For Loop.

```
Imports System.Console
Module loops
    Sub main()
        Dim no As Integer = 10
        For counter = 0 To no
            Write(" " & counter)
        Next
        ReadKey()
    End Sub
End Module
```

Run the program and note down output.

Observe the code and difference between While loop and for loop.

19. Program to demonstrate the use of For Loop with Step Option.

```
Imports System.Console
Module loops
    Sub main()
        Dim no As Integer = 10
        For counter = 0 To no Step
2
            Write(" " & counter)
        Next
        ReadKey()
    End Sub
End Module
```

```
Imports System.Console
Module loops
    Sub main()
        Dim no As Integer = 0
        For counter = 10 To no Step -2
            Write(" " & counter)
        Next
        ReadKey()
    End Sub
End Module
```

Run the Program and check the difference and understand the role of step in for…loop.

Above three programs of for..loop note down conclusion.

20. **Program to demonstrate the alternate syntax of Do..While Loop.**

```
Imports System.Console
Module loops
    Sub main()
        Dim no As Integer = 10
        Dim counter = 0
        Do
            counter = counter + 1
            Write(" " & counter)
        Loop While counter < no
        ReadKey()
    End Sub
End Module
```

```
Imports System.Console
Module loops
    Sub main()
        Dim no As Integer = 10
        Dim counter = 0
        Do While counter < no
            counter = counter + 1
            Write(" " & counter)
        Loop
        ReadKey()
    End Sub
End Module
```

Both program are same but do loop have variations.

21. **Program to demonstrate the difference between While loop and Do… While loop.**

```
Imports System.Console
Module loops
    Sub main()
        Dim no As Integer = 1
        While no > 10
            WriteLine()
        End While
        WriteLine("Exit From While
Loop")
        ReadKey()
    End Sub
End Module
```

```
Imports System.Console
Module loops
    Sub main()
        Dim no As Integer = 1
        Do
            WriteLine("No is Greater
than 10")
        Loop While no > 10
        WriteLine("Exit From While
Loop")
        ReadKey()
    End Sub
End Module
```

Execute both program and note down the output. Understand the difference between the execution behavior of body of loop.

22. **Program to demonstrate the use of Arrays.**

```
Imports System.Console
Module Arrays
    Sub main()
        Dim intArray(5) As Integer
        intArray(0) = 5
        intArray(1) = 10
        intArray(2) = 15
        intArray(3) = 20
        intArray(4) = 25
        intArray(5) = 30
        WriteLine("Length (Size) of Array : " & intArray.Length)
        WriteLine("First Element of Array : " & intArray(0))
        WriteLine("3rd Element of Array : " & intArray(2))
        WriteLine("Last Element of Array : " & intArray(intArray.Length - 1))
        'You can Change value of any index
        intArray(3) = 100
        WriteLine("4th Element of Array : " & intArray(3))
        'Display All the Elements of Array
        For i = 0 To intArray.Length - 1
            Write(" " & intArray(i))
        Next
        ReadKey()
    End Sub
End Module
```

23. Program to demonstrate to set the values to array and retrieve from it.

```
Imports System.Console
Module Arrays
    Sub main()
        Dim intArray(5) As Integer
        'Accept Array Element from User
        WriteLine("Enter Array Element one by one")
        For i = 0 To intArray.Length - 1
            Write("Enter Element at " & i & " : ")
            intArray(i) = Integer.Parse(ReadLine())
        Next
        'Display Array Elements
        WriteLine("Array Elements Are...")
        For i = 0 To intArray.Length - 1
            Write(" " & intArray(i))
        Next
        ReadKey()
    End Sub
End Module
```

24. Program to demonstrate the Dynamic array using ReDim Statement.

```
Imports System.Console
Module Arrays
    Sub main()
        Dim intArray(5) As Integer
        'Accept Array Element from User
        WriteLine("Enter Array Element one by one")
        For i = 0 To intArray.Length - 1
            Write("Enter Element at " & i & " : ")
            intArray(i) = Integer.Parse(ReadLine())
        Next

        'Display Array Elements
        WriteLine("Array Elements Are...")
        For i = 0 To intArray.Length - 1
            Write(" " & intArray(i))
        Next
        WriteLine()
        ReDim intArray(10)
        'Notice that first 5 elements are lost on redim
        WriteLine("Array Elements Are...")
        For i = 0 To intArray.Length - 1
            Write(" " & intArray(i))
        Next
        ReadKey()
    End Sub
End Module
```

Execute the Program and note down the output. Also understand the usage of Redim and its impact on output.

Now, change highlighted line with ReDim Preserve intArray(10) and check output again.

Add after sub main()

```
Write("Enter Total no. of Elements : ")
Dim intArray(ReadLine()) As Integer
```

Write a Program to print Array Element in Reverse order.

25. **Program to find Simple Interest or Compound Interest based on user choice. It is a combination of Do…While statement and Select Case Statement.**

```
Imports System.Console
Module Arrays
    Sub main()
        Dim choice As Integer
        Dim p As Integer
        Dim r As Integer
        Dim n As Integer
        Dim interest As Double
        Do
            Clear()
            Write("Enter Principal Amount : ")
            p = Integer.Parse(ReadLine())
            Write("Enter Rate of Interest : ")
            r = Integer.Parse(ReadLine())
            Write("Enter Duration : ")
            n = Integer.Parse(ReadLine())
            Clear()
            WriteLine("-------------------------------------")
            WriteLine("1. Simple Interest.")
            WriteLine("2. Compound Interst.")
            WriteLine("-------------------------------------")
            Write("Enter Your Choice : ")
            choice = Integer.Parse(ReadLine())
            Clear()
            Select Case choice
                Case 1
                    'Simple Interest
                    interest = p * r * n / 100
                Case 2
                    'Compound Interest
                    interest = (p * Math.Pow((1 + r / 100), n)) - p
                Case Else
                    WriteLine("Invalid Choice")
            End Select
            WriteLine("----------------------------")
            WriteLine("Principal Amount : " & p)
            WriteLine("Rate of Interest : " & r)
            WriteLine("Duration : " & n)
            WriteLine("Interest : " & interest)
            WriteLine("----------------------------")
            WriteLine("Do you want to Continue....(y/n) : ")
        Loop While Char.Parse(ReadLine()) = "y"
        End
    End Sub
End Module
```

26. **Program to demonstrate the use of Date functions.**

```
Imports System.Console
Module Arrays
    Sub main()
        Dim today As Date = Date.Now
        WriteLine("Today's Date : " & today)
        today = Date.Today
        WriteLine("Today's Date : " & today)
        WriteLine("Indian Format : " & today.ToString("dd/MM/yy"))
        Dim dob As Date
        Write("Enter Date of Birth (dd/mm/yy): ")
        dob = Date.Parse(ReadLine())
        'Find Your Age
        WriteLine("Age : " & Math.Round(DateDiff(DateInterval.Day, dob, today)
/ 365))
        WriteLine("Retrirement Date : " & DateAdd(DateInterval.Year, 60, dob))
        WriteLine("U.S. Format : " & Date.Now.ToString("ddd, MMM dd, yyy"))
        ReadKey()
    End Sub
End Module
```

Try out other formats of date to display current date.

Don’t try to write whole program at once, instead try one string function at a time and observe the working of function.

27. **Program to demonstrate the use of String Functions.**

```
Imports System.Console
Module Strings
    Sub main()
        Dim st1 As String = "  Government Science College Valod "
        WriteLine("Original String : " & st1)
        WriteLine("Length of String : " & st1.Length)
        'Remove space from Left side
        WriteLine("Left Trim : " & LTrim(st1))
        'Remove space from both sides
        WriteLine("Trimmed String : " & Trim(st1))
        WriteLine("Length of Trimmed String : " & Trim(st1).Length)
        WriteLine("SubString : " & st1.Substring(2, 10))
        WriteLine("Replaced String : " & st1.Replace("Science", "Commerce &
Arts"))
        WriteLine("Remove Government From String : " & st1.Remove(2, 10))
        'Extract Words from Strings with separator space
        Dim strArray() As String = st1.Split(" ")
        Dim st2 As String = "Hello"
        Dim st3 As String = "World"
        WriteLine("Concatenated String : " & st2 + st3)
        st2 = String.Concat(st2, st3)
        WriteLine("Concatenated String : " & st2)
        WriteLine("Inserted Space after Hello : " & st2.Insert(5, " "))
        WriteLine("String in Words of Array")
        For i = 0 To strArray.Length - 1
            WriteLine(strArray(i))
        Next
        st2 = "Hello"
        st3 = "hello"
        'Compare String Textually
        If (String.Compare(st2, st3)) Then
            WriteLine("Both are Equal")
        Else
            WriteLine("Both are not Equal")
        End If
        st2 = st3

        'Compare String at Binary Level
        If (st2 = st3) Then
            WriteLine("Both are Equal")
        Else
            WriteLine("Both are not Equal")
        End If

        'Compare String Contents, Compare Text
        If st1.Equals(st2) Then
            WriteLine("Both are Equal")
        Else
            WriteLine("Both are not Equal")
        End If
        WriteLine("Search Science From Original String : " &
st1.IndexOf("Science"))

        ReadKey()
    End Sub
End Module
```