Date	Dec 11 2024	Unit	3

Introduction to VB.NET

Introduction to VB.NET

VB.NET (Visual Basic .NET) is an object-oriented programming language from Microsoft. It is part of the .NET framework and is used for creating Windows, web, and mobile applications. VB.NET provides an Integrated Development Environment (IDE) called Visual Studio, which simplifies development with features like drag-and-drop controls, IntelliSense, and debugging tools.

Getting Started with VB.NET

Starting VB.NET

1. **IDE Overview**: Use Microsoft Visual Studio as the primary development environment for VB.NET.
2. **Creating a Shortcut**: Pin Visual Studio to the taskbar or desktop for quick access.
3. **Toolbar Navigation**:
    - **Auto-hide**: Use the pushpin icon to enable or disable auto-hide for toolbars.
    - **Docking/Undocking**: Drag toolbars to dock them to a specific area or float them by dragging them outside the IDE.

VB.NET Components Overview

1. **Forms**: The base for creating user interfaces. Forms can contain controls like buttons, labels, and text boxes.
2. **Properties Window**: Allows modification of control properties like `Text`, `Size`, `Color`, etc.
3. **Solution Explorer**: Displays the project structure, including forms, classes, and resources.

Basic VB.NET Keywords and Data Types

Keywords

VB.NET keywords include `Dim`, `If`, `For`, `While`, `End`, `Sub`, `Function`, etc.

The `Dim` keyword is used to **declare** variables.

`Dim` is a fundamental keyword in VB.NET used to declare variables.

It helps in defining the type of data the variable will hold and the scope of the variable.

It's a powerful keyword for managing variables within your code.

Data Types

- **Integer**: `Dim age As Integer = 25`
- **String**: `Dim name As String = "John"`
- **Boolean**: `Dim isActive As Boolean = True`
- **Double**: `Dim price As Double = 99.99`

VB.NET Statements and Control Structures

Conditional Statements

1. **If-Else Statement**

```vb
Module Program
    Sub Main()
        Dim num As Integer = 10
        If num > 5 Then
            Console.WriteLine("Number is greater than 5")
        Else
            Console.WriteLine("Number is 5 or less")
        End If
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
Number is greater than 5
```

1. **Select Case Statement**:

```vb
Module Program
    Sub Main()
        Dim day As Integer = 3
        Select Case day
            Case 1
                Console.WriteLine("Monday")
            Case 2
                Console.WriteLine("Tuesday")
            Case Else
                Console.WriteLine("Other Day")
        End Select
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
Other Day
```

2. **Switch and Choose** (Alternative methods):

```vb
Module Program
    Sub Main()
        Dim num As Integer = 2
        Dim result As String = Switch(num = 1, "One", num = 2, "Two",
        Console.WriteLine(result)
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
Two
```

More Example programs

If else

```
Module Program
    Sub Main()
        Console.WriteLine("Enter a number:")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        If num > 10 Then
            Console.WriteLine("Number is greater than 10")
        Else
            Console.WriteLine("Number is less than or equal to 10")
        End If

        Console.ReadLine()
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
Enter a number:
Number is less than or equal to 10
```

Select case statement

```
Module Program
    Sub Main()
        Console.WriteLine("Enter a number between 1 and 3:")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        Select Case num
            Case 1
                Console.WriteLine("You selected One")
            Case 2
                Console.WriteLine("You selected Two")
            Case 3
                Console.WriteLine("You selected Three")
            Case Else
                Console.WriteLine("Invalid selection")
        End Select

        Console.ReadLine()
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
Enter a number between 1 and 3:
Invalid selection
```

For-next loop

```vb
Module Program
    Sub Main()
        Console.WriteLine("For-Next Loop Example:")

        For i As Integer = 1 To 5
            Console.WriteLine("Iteration " & i)
        Next

        Console.ReadLine()
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
For-Next Loop Example:
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
```

For each next loop

```vb
Module Program
    Sub Main()
        Dim fruits As String() = {"Apple", "Banana", "Cherry"}

        Console.WriteLine("For Each-Next Loop Example:")
        For Each fruit As String In fruits
            Console.WriteLine(fruit)
        Next

        Console.ReadLine()
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
For Each-Next Loop Example:
Apple
Banana
Cherry
```

While loop

```vb
Module Program
    Sub Main()
        Dim counter As Integer = 1
        Console.WriteLine("While Loop Example:")

        While counter <= 5
            Console.WriteLine("Counter: " & counter)
            counter += 1
        End While

        Console.ReadLine()
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
While Loop Example:
Counter: 1
Counter: 2
Counter: 3
Counter: 4
Counter: 5
```

Do-while loop

```vb
Module Program
    Sub Main()
        Dim num As Integer = 1
        Console.WriteLine("Do-While Loop Example:")

        Do While num <= 5
            Console.WriteLine("Number: " & num)
            num += 1
        Loop

        Console.ReadLine()
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
Do-While Loop Example:
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
```

Array Example

```vb
Module Program
    Sub Main()
        Dim numbers As Integer() = {10, 20, 30, 40, 50}

        Console.WriteLine("Array Example:")

        For i As Integer = 0 To numbers.Length - 1
            Console.WriteLine("Element at index " & i & ": " & number
        Next

        Console.ReadLine()
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
Array Example:
Element at index 0: 10
Element at index 1: 20
Element at index 2: 30
Element at index 3: 40
Element at index 4: 50
```

Basic calculator program

```vb
Module Program
    Sub Main()
        Console.WriteLine("Enter first number:")
        Dim num1 As Double = Convert.ToDouble(Console.ReadLine())

        Console.WriteLine("Enter second number:")
        Dim num2 As Double = Convert.ToDouble(Console.ReadLine())

        Console.WriteLine("Select operation: (+, -, *, /)")
        Dim operation As String = Console.ReadLine()

        Dim result As Double

        Select Case operation
            Case "+"
                result = num1 + num2
            Case "-"
                result = num1 - num2
            Case "*"
                result = num1 * num2
            Case "/"
                If num2 <> 0 Then
                    result = num1 / num2
                Else
                    Console.WriteLine("Error: Division by zero")
                    Exit Sub
                End If
            Case Else
                Console.WriteLine("Invalid operation")
                Exit Sub
        End Select

        Console.WriteLine("Result: " & result)
        Console.ReadLine()
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
Enter first number:
Enter second number:
Select operation: (+, -, *, /)
Invalid operation
```

Even odd program

```vb
Module Program
    Sub Main()
        Console.WriteLine("Enter a number to check if it's even or od
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())

        If num Mod 2 = 0 Then
            Console.WriteLine("The number is even.")
        Else
            Console.WriteLine("The number is odd.")
        End If

        Console.ReadLine()
    End Sub
End Module
```

STDIN

3

Output:

```
Enter a number to check if it's even or odd:
The number is odd.
```

Fibonacci Sequence

```
Module Program
    Sub Main()
        Dim n As Integer
        Console.WriteLine("Enter a number to generate Fibonacci serie
        n = Convert.ToInt32(Console.ReadLine())

        Dim a As Integer = 0
        Dim b As Integer = 1
        Dim c As Integer

        Console.WriteLine("Fibonacci Sequence:")

        For i As Integer = 1 To n
            Console.WriteLine(a)
            c = a + b
            a = b
            b = c
        Next

        Console.ReadLine()
    End Sub
End Module
```

STDIN

3

Output:

```
Enter a number to generate Fibonacci series:
Fibonacci Sequence:
0
1
1
```

Factorial Program

```vb
Module Program
    Sub Main()
        Console.WriteLine("Enter a number to find its factorial:")
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim factorial As Integer = 1

        For i As Integer = 1 To num
            factorial *= i
        Next

        Console.WriteLine("Factorial of " & num & " is: " & factorial
        Console.ReadLine()
    End Sub
End Module
```

STDIN

```
3
```

Output:

```
Enter a number to find its factorial:
Factorial of 3 is: 6
```

String reverse

```vb
Module Program
    Sub Main()
        Console.WriteLine("Enter a string to reverse:")
        Dim input As String = Console.ReadLine()
        Dim reversed As String = ""

        For i As Integer = input.Length - 1 To 0 Step -1
            reversed &= input(i)
        Next

        Console.WriteLine("Reversed String: " & reversed)
        Console.ReadLine()
    End Sub
End Module
```

STDIN

```
xbitlabs
```

Output:

```
Enter a string to reverse:
Reversed String: sbaltibx
```

Sum of the digits of a number

```vb
Module Program
    Sub Main()
        Console.WriteLine("Enter a number to find the sum of its digi
        Dim num As Integer = Convert.ToInt32(Console.ReadLine())
        Dim sum As Integer = 0

        While num > 0
            sum += num Mod 10
            num \= 10
        End While

        Console.WriteLine("Sum of digits: " & sum)
        Console.ReadLine()
    End Sub
End Module
```

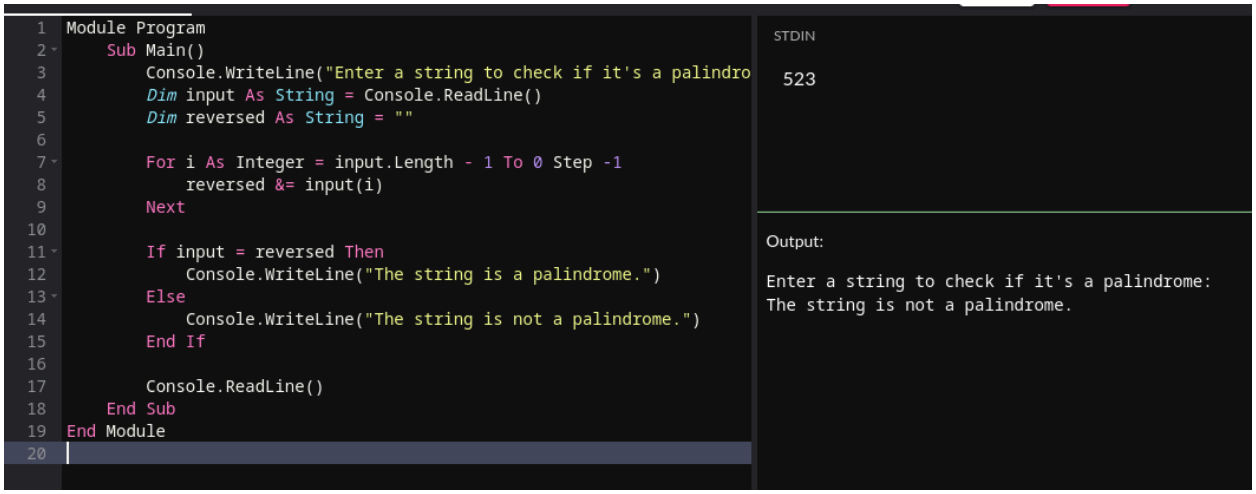
STDIN

```
523
```

Output:

```
Enter a number to find the sum of its digits:
Sum of digits: 10
```

String Reverse program



These console-based programs demonstrate key programming concepts in VB.NET, including basic control flow, loops, arrays, and simple mathematical operations. You can expand these programs with more advanced features as you get comfortable with the syntax and logic.

For GUI based applications when we work in text editor e.g vs code
MessageBox.Show($"Number: {num}")

Can be used instead of Console.writeline()

END