

Table of Contents



- [1. Hello World \(Basic Syntax\)](#)
- [2. Variables & Data Types](#)
- [3. Constants](#)
- [4. Operators](#)
- [5. Conditionals](#)
- [6. Select Case \(Switch\)](#)
- [7. Loops](#)
 - [For Loop](#)
 - [While Loop](#)
 - [Do-While Loop](#)
- [8. Arrays](#)
- [9. Functions](#)
- [10. Subroutines](#)
- [11. Classes & Objects](#)
- [12. Properties \(Encapsulation\)](#)
- [13. Inheritance](#)
- [14. Polymorphism](#)
- [15. Interfaces](#)
- [16. Exception Handling](#)
- [17. User Input](#)
- [18. File Handling](#)
 - [Writing to a File](#)
 - [Reading from a File](#)
- [19. Lists](#)
- [20. Dictionaries](#)
- [21. Asynchronous Programming](#)
- [Conclusion](#)

1. Hello World (Basic Syntax)

Module Program

```
Sub Main()
```

```
    Console.WriteLine("Hello, World!") ' Output text to console  
    Console.ReadLine() ' Pause execution
```

```
End Sub
End Module
```

- Module → Defines a module (container for procedures).
- Sub Main() → Entry point of the program.
- Console.WriteLine() → Outputs text.

2. Variables & Data Types

```
Dim age As Integer = 25      ' Integer
Dim price As Double = 9.99   ' Decimal number
Dim grade As Char = "A"     ' Single character
Dim name As String = "John"  ' Text
Dim isActive As Boolean = True ' Boolean
```

- Dim → Declares a variable.
- Integer, Double, Char, String, Boolean → Common data types.

3. Constants

```
Const PI As Double = 3.14159 ' Cannot be changed
```

4. Operators

Operator	Description	Example
+	Addition	a + b
-	Subtraction	a - b
*	Multiplication	a * b
/	Division	a / b
\	Integer Division	10 \ 3 (Output: 3)
Mod	Modulus	a Mod b
^	Exponentiation	2 ^ 3 (Output: 8)

5. Conditionals

```
Dim num As Integer = 10
```

```
If num > 5 Then
    Console.WriteLine("Greater than 5")
ElseIf num = 5 Then
    Console.WriteLine("Equal to 5")
Else
    Console.WriteLine("Less than 5")
End If
```

6. Select Case (Switch)

```
Dim grade As Char = "A"c

Select Case grade
    Case "A"c
        Console.WriteLine("Excellent!")
    Case "B"c
        Console.WriteLine("Good!")
    Case Else
        Console.WriteLine("Invalid grade")
End Select
```

7. Loops

For Loop

```
For i As Integer = 1 To 5
    Console.WriteLine(i)
Next
```

While Loop

```
Dim count As Integer = 0
While count < 5
    Console.WriteLine(count)
    count += 1
End While
```

Do-While Loop

```
Dim num As Integer = 0
Do
    Console.WriteLine(num)
    num += 1
Loop While num < 5
```

8. Arrays

```
Dim numbers() As Integer = {1, 2, 3, 4, 5}
Console.WriteLine(numbers(0)) ' Outputs 1
```

9. Functions

```
Function Add(a As Integer, b As Integer) As Integer
    Return a + b
End Function
```

```
Console.WriteLine(Add(3, 4)) ' Output: 7
```

10. Subroutines

```
Sub Greet(name As String)
    Console.WriteLine("Hello, " & name)
End Sub
```

```
Greet("Alice")
```

- Function → Returns a value.
- Sub → Does not return a value.

11. Classes & Objects

```
Class Car
    Public Brand As String

    Public Sub New(b As String)
```

```
        Brand = b
    End Sub
End Class
```

```
Dim myCar As New Car("Toyota")
Console.WriteLine(myCar.Brand)
```

- Class → Defines a blueprint.
- Object → Created using New.

12. Properties (Encapsulation)

```
Class Person
    Private _name As String

    Public Property Name() As String
        Get
            Return _name
        End Get
        Set(value As String)
            _name = value
        End Set
    End Property
End Class
```

```
Dim p As New Person()
p.Name = "Alice"
Console.WriteLine(p.Name) ' Output: Alice
```

13. Inheritance

```
Class Animal
    Public Sub MakeSound()
        Console.WriteLine("Some sound")
    End Sub
End Class
```

```
Class Dog
```

```

    Inherits Animal

    Public Sub Bark()
        Console.WriteLine("Woof!")
    End Sub
End Class

Dim myDog As New Dog()
myDog.MakeSound() ' Inherited method
myDog.Bark() ' Dog's own method

```

14. Polymorphism

```

Class Animal
    Overridable Sub Speak()
        Console.WriteLine("Animal sound")
    End Sub
End Class

Class Dog
    Inherits Animal

    Overrides Sub Speak()
        Console.WriteLine("Bark!")
    End Sub
End Class

Dim myAnimal As Animal = New Dog()
myAnimal.Speak() ' Output: Bark!

```

15. Interfaces

```

Interface IAnimal
    Sub MakeSound()
End Interface

Class Dog
    Implements IAnimal

```

```
        Public Sub MakeSound() Implements IAnimal.MakeSound
            Console.WriteLine("Bark!")
        End Sub
    End Class
```

```
Dim myDog As New Dog()
myDog.MakeSound()
```

16. Exception Handling

```
Try
    Dim x As Integer = 10 / 0 ' Error
Catch ex As Exception
    Console.WriteLine("Error: " & ex.Message)
Finally
    Console.WriteLine("This runs no matter what.")
End Try
```

17. User Input

```
Console.Write("Enter your name: ")
Dim userName As String = Console.ReadLine()
Console.WriteLine("Hello, " & userName)
```

18. File Handling

Writing to a File

```
System.IO.File.WriteAllText("test.txt", "Hello World!")
```

Reading from a File

```
Dim content As String = System.IO.File.ReadAllText("test.txt")
Console.WriteLine(content)
```

19. Lists

```
Imports System.Collections.Generic
```

```
Dim names As New List(Of String) From {"Alice", "Bob"}  
names.Add("Charlie")  
Console.WriteLine(names(2)) ' Outputs Charlie
```

20. Dictionaries

```
Imports System.Collections.Generic
```

```
Dim ages As New Dictionary(Of String, Integer)  
ages("Alice") = 25  
ages("Bob") = 30
```

```
Console.WriteLine(ages("Alice")) ' 25
```

21. Asynchronous Programming

```
Imports System
```

```
Imports System.Threading.Tasks
```

```
Module Program
```

```
    Async Function DoWork() As Task  
        Await Task.Delay(2000)  
        Console.WriteLine("Work Done")  
    End Function
```

```
    Async Sub Main()  
        Await DoWork()  
        Console.WriteLine("Main Finished")  
    End Sub
```

```
End Module
```

Conclusion

This Visual Basic .NET Cheat Sheet covers:

- Basic Syntax
- Variables, Loops, Conditionals
- Object-Oriented Programming (Classes, Inheritance, Polymorphism)
- Exception Handling & File Handling
- Collections (Lists, Dictionaries)
- Asynchronous Programming

This serves as a quick reference guide to get started with VB.NET programming.