



Azli 1920 | Azərbaycan Dövlət Neft və Sənaye Universiteti

**Faculty :** Information techniques and  
management

**Speciality:** Computer Engineering

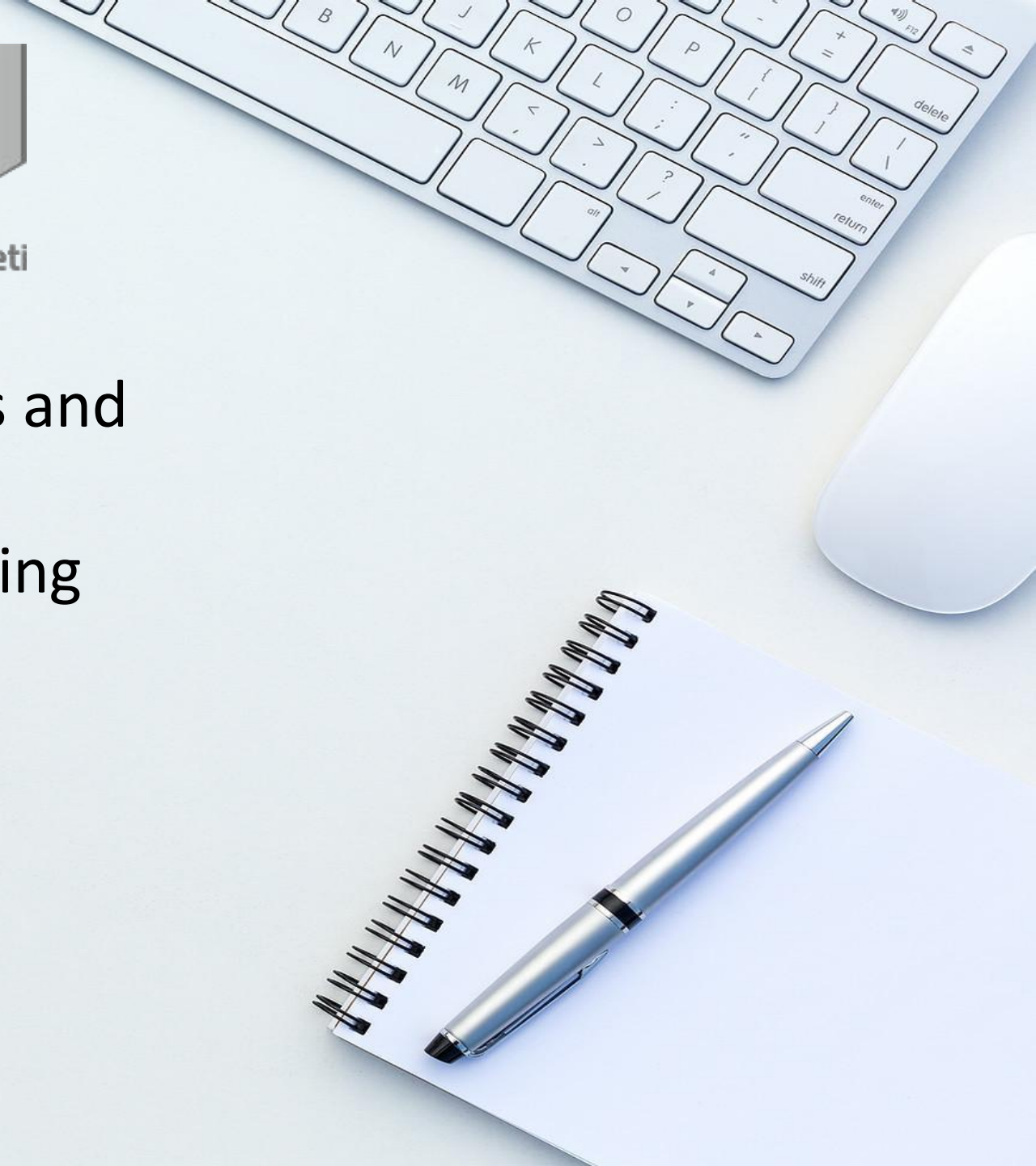
**Group:** 652.21E

**Subject:** Fundamentals of  
Programming

**Teacher:** Vagif Salimov

**Student:** Aysu Sadikhova

**Topic:** VB.NET



# *What is VB.Net? (General Overview)*

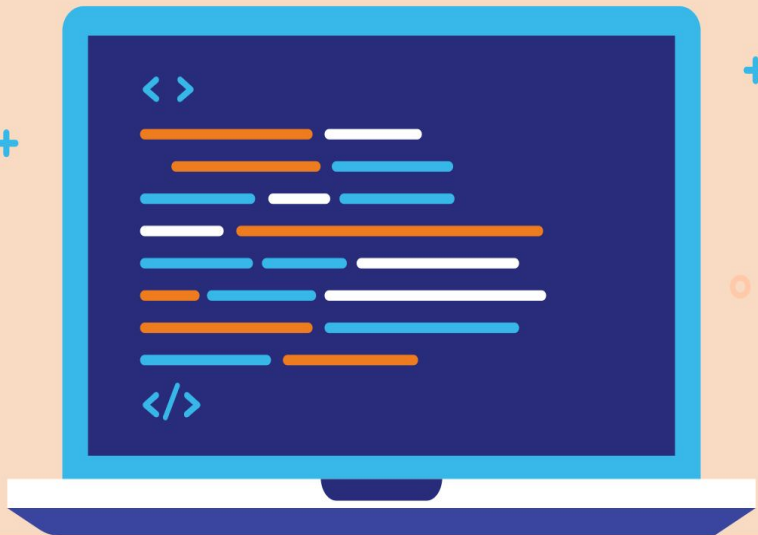
*VB.NET stands for Visual Basic.NET, and it is a computer programming language developed by Microsoft. It was first released in 2002 to replace Visual Basic 6. VB.NET is an object-oriented programming language. This means that it supports the features of object-oriented programming which include encapsulation, polymorphism, abstraction, and inheritance. The language was designed in such a way that it is easy to understand to both novice and advanced programmers.*



## VB.NET



# History of VB.NET



- *VB.NET is a multi-paradigm programming language developed by Microsoft on the .NET framework. It was launched in 2002 as a successor to the Visual Basic language. This was the first version of VB.NET and it relied on .NET version 1.0.*
- *In 2003, the second version of VB.NET, VB.NET 7.1, was released. This one relied on .NET version 1.1. This version came with a number of improvements including support for .NET Compact Framework and an improved reliability and performance of the .NET IDE.*
- *In 2005, VB.NET 8.0 was released. The .NET core portion was dropped from its name so as to distinguish it from the classical Visual Basic language. This version came with many features since Microsoft wanted this language to be used for rapid application developers. They also wanted to make it different from C# language*



- In 2008, VB 9.0 was introduced. This was released together with .NET 3.5. Some of the features added to this release of VB.NET included anonymous types, true conditional operator, Lambda expressions, extension methods, and type inference.
- In 2010, Microsoft released VB 2010 (code 10.0). They wanted to use a Dynamic Language Runtime for this release, but they opted for co-evolution strategy shared between VB.NET and C# to bring these languages closer to each other.
- In 2012, VB 2012 was release together with .NET 4.5. Its features included call hierarchy, iterators, asynchronous programming with “await” and “async” statements and the “Global” keyword in the “namespace” statements.
- In 2015, VB 2015 was released alongside Visual Studio 2015. The “?.” operator was introduced to do inline null checks. A string interpolation feature was also introduced to help in formatting strings inline.
- In 2017, VB 2017 was introduced alongside Visual Studio 2017. A better way of organizing source code in just a single action was introduced.

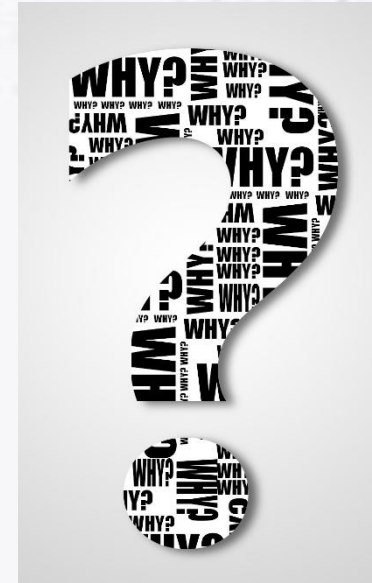


Microsoft  
**Visual Basic** 6.0

**VB**  
**.net**

# WHY VB.NET?

*Visual basic programming language allows programmers to create software interface and codes in an easy to use graphical environment. VB is the combination of different components that are used on forms having specific attributes and actions with the help of those components. On the one hand it allows programmers to develop windows based applications rapidly; on the other hand, it helps greatly in accessing data bases, using ADO while letting the programmers use ActiveX controls and various objects. While it is intended more to develop applications, it is also useful for games development for particular or limited purposes, unlike C++ that is more suitable for developing games.*



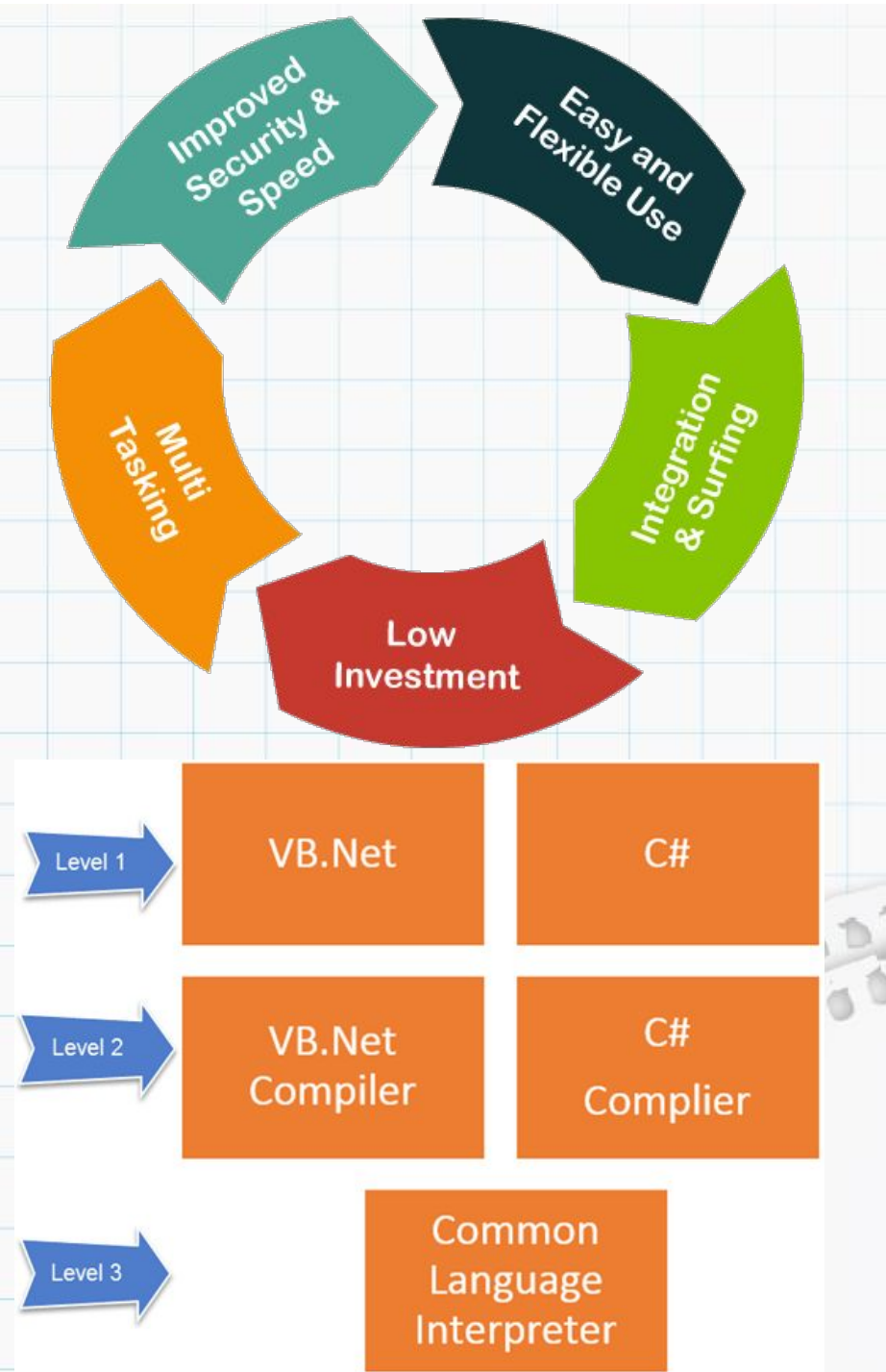
# Features of VB.NET

- VB.NET is not case sensitive like other languages such as C++ and Java.
- It is an object-oriented programming language. It treats everything as an object.
- Automatic code formatting, XML designer, improved object browser etc.
- Garbage collection is automated.
- Support for Boolean conditions for decision making.
- Simple multithreading, allowing your apps to deal with multiple tasks simultaneously.
- Simple generics.
- A standard library.
- Events management.
- References. You should reference an external object that is to be used in a VB.NET application.
- Attributes, which are tags for providing additional information regarding elements that have been defined within a program.
- Windows Forms- you can inherit your form from an already existing form.





- **Assemblies:** An EXE or a DLL file that forms the basis of deployment, version control, reuse, and security permissions of an application.
- **Namespaces:** The components of assemblies, namespaces primarily organize the objects present in the assemblies. An assembly can contain more than one namespace.
- **Adding references:** You need to add a reference to the external object that you want to use in your current application.
- **Attributes:** The tags that are used to provide additional information about the elements defined in a Visual Basic .NET program. Some of the most common uses of attributes are
  - To explain COM properties for classes, interfaces, and methods
  - To explain assemblies
  - To specify security requirements of methods
  - To specify features required to enforce security



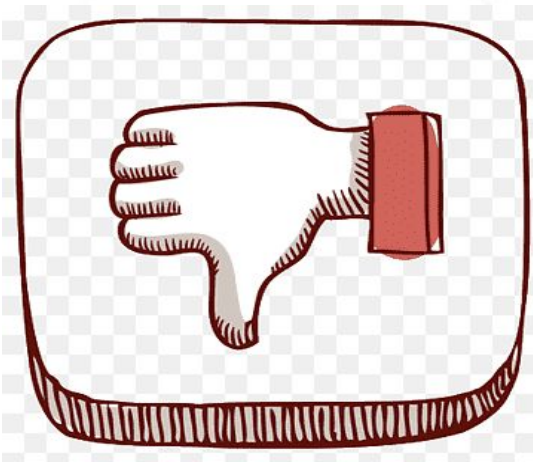
# Advantages of VB.NET

- Your code will be formatted automatically.
- You will use object-oriented constructs to create an enterprise-class code.
- You can create web applications with modern features like performance counters, event logs.
- You can create your web forms with much ease through the visual forms designer.
- You can connect your applications to other applications created in languages that run on the .NET framework.
- You will enjoy features like docking, automatic control anchoring, and in-place menu editor all good for developing web applications.

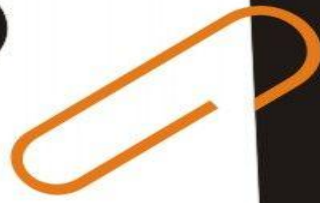




# Disadvantages of VB.NET



- VB.NET cannot handle pointers directly. This is a significant disadvantage since pointers are much necessary for programming. Any additional coding will lead to many CPU cycles, requiring more processing time. Your application will become slow.
- VB.NET is easy to learn. This has led to a large talent pool. Hence, it may be challenging to secure a job as a VB.NET programmer.



## *Pre-Requisite for VB.Net*

*It is a programming language which is very much based on other two Microsoft technology which is BASIC and Visual Basic programming languages, so if someone who has a basic understanding of these programming languages, then it is quite easier and fun to learn VB.Net programming language. Now, after knowing in-depth about VB.Net, it is worth knowing about its scope.*

## *Scope*

*At first, instance, if we say, VB.Net has a high scope. To be more precise, but only VB.Net does not have much scope. VB.Net, although it still exists in the Top 10 programming language, it will be beneficial to learn VB.Net. However, alone learning and practicing VB.Net will not be much helpful. Once you learned VB.Net, then once should learn other Microsoft technology like C# and .Net, which will be much easier to learn and understand. Together all these Microsoft technologies will be in great demand.*



# Syntax

In VB.Net methodology, a program consists of various objects that interact with each other by means of actions. The actions that an object may take are called methods. Objects of the same kind are said to have the same type or, more often, are said to be in the same class. When we consider a VB.Net program, it can be defined as a collection of objects that communicate via invoking each other's methods. Let us now briefly look into what do class, object, methods and instance variables mean.

**Object** – Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors - wagging, barking, eating, etc. An object is an instance of a class.

**Class** – A class can be defined as a template/blueprint that describes the behaviors/states that objects of its type support.

**Methods** – A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.

**Instance Variables** – Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.

A Rectangle Class in VB.Net



# EXAMPLE

```
Imports System
Public Class Rectangle
    Private length As Double
    Private width As Double

    'Public methods
    Public Sub AcceptDetails()
        length = 4.5
        width = 3.5
    End Sub

    Public Function GetArea() As Double
        GetArea = length * width
    End Function

    Public Sub Display()
        Console.WriteLine("Length: {0}", length)
        Console.WriteLine("Width: {0}", width)
        Console.WriteLine("Area: {0}", GetArea())
    End Sub

    Shared Sub Main()
        Dim r As New Rectangle()
        r.Acceptdetails()
        r.Display()
        Console.ReadLine()
    End Sub
End Class
```

*RESULT:*

Length: 4.5  
Width: 3.5  
Area: 15.75

# Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. VB.Net is rich in built-in operators and provides following types of commonly used operators –

- Arithmetic Operators (mathematical operations such as **subtraction, addition, multiplication, division**, etc.)
- Comparison Operators (is used to compare the value of two variables or operands for the various condition such as greater, less than or equal, etc.)
- Logical/Bitwise Operators (work with Boolean (true or false) conditions/ perform the various logical operations such as And, Or, Not, etc.)
- Bit Shift Operators (used to perform the bit shift operations on binary values)
- Assignment Operators (used to assign the value to variables)
- Miscellaneous Operators



# Arithmetic Operators

&

# Comparison Operators

Name	Operator	Description
Addition	+	Sum of two numbers.
Subtraction	−	Difference of two numbers.
Multiplication	*	Product of two numbers.
Division	/	Division of two numbers, returns float value.
Division	\	Division of two numbers, returns integer value. (Decimal part removed)
Modulus	MOD	Remainder of the division of two numbers.
Exponent	^	Exponential of a number

Name	Operator	Description
Equals to	=	Returns True if values are equal
Not Equal to	<>	Returns True if values are not equal
Greater than	>	Returns True is value is greater
Greater than or equals to	>=	Returns True is value is equal or greater
Less than:	<	Returns True is value is smaller
Less than:	<=	Returns True if value is equal or smaller

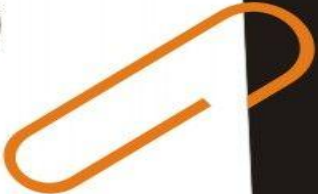


## Example to understand all the arithmetic operators



```
Module operators
Sub Main()
Dim a As Integer = 21
Dim b As Integer = 10
Dim p As Integer = 2
Dim c As Integer
Dim d As Single c = a + b
Console.WriteLine
("Line 1 - Value of c is {0}",
c) c = a - b
Console.WriteLine
("Line 2 - Value of c is {0}",
c) c = a * b
Console.WriteLine
("Line 3 - Value of c is {0}",
c) d = a / b
Console.WriteLine
("Line 4 - Value of d is {0}",
d) c = a \ b
Console.WriteLine
("Line 5 - Value of c is {0}",
c) c = a Mod b
Console.WriteLine
("Line 6 - Value of c is {0}",
c) c = b ^ p
Console.WriteLine
("Line 7 - Value of c is {0}",
c) Console.ReadLine()
End Sub
End Module
```

## RESULT:



```
Line 1 - Value of c is 31
Line 2 - Value of c is 11
Line 3 - Value of c is 210
Line 4 - Value of d is 2.1
Line 5 - Value of c is 2
Line 6 - Value of c is 1
Line 7 - Value of c is 100
```

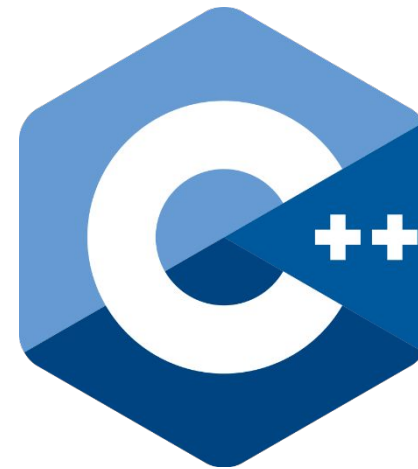
# *Difference Between VB.NET and Visual Studio*

## *VB.NET vs C++ performance*

- *Visual Basic is an event driven programming language, while Visual Studio is a Software development tool (Integrated development Environment). So **there is no point in comparing programming language with IDE.** Visual Studio has a component for Visual Basic.*
- *C++ compiler does do a lot more optimization (even more than the C#/VB.Net compilers + JIT combined), and is less abstracted, which means you can typically, with enough effort, write C++ code that is **faster than** the equivalent managed code.*



- *Visual Basic tends to be Microsoft oriented; C++ is generic.*
- *C++ and VB are two completely different languages and have quite a few fundamental differences (managed vs. unmanaged being a major one that comes to mind...). That said, if you are just doing some basic windows programming or web development (in ASP.NET) I would stick with VB.Net, there really isn't much reason to struggle with trying to learn C++ for that.*



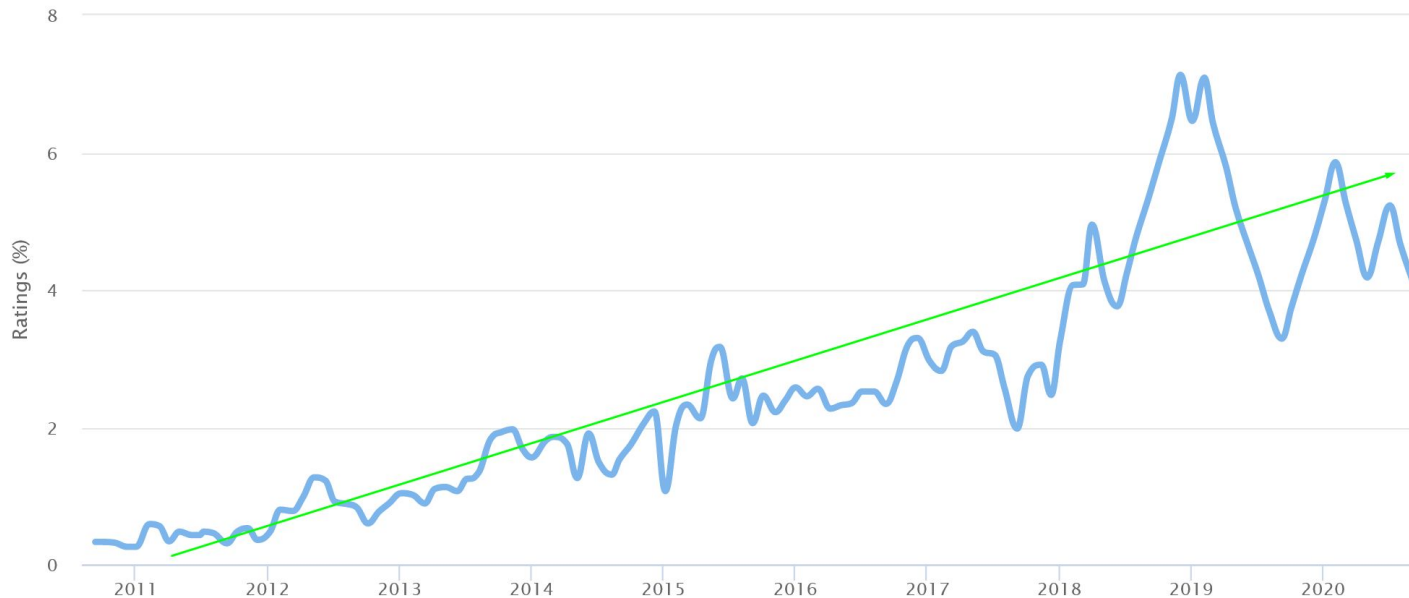


## IS VB.NET DEAD IN 2021?

Visual Basic (VB.NET) will continue to be supported by Microsoft. (**It's not dead.**) The language will no longer have new features added to it.

TIOBE Index for Visual Basic

Source: [www.tiobe.com](http://www.tiobe.com)



Rank	Change	Language	Share	Trend
14	↑	Scala	1.19 %	-0.1 %
15	↑↑	Kotlin	1.14 %	+0.4 %
16	↓↓	Visual Basic	1.13 %	-0.1 %
17	↑	Go	1.04 %	+0.3 %



# *References*

[https://www.tutorialspoint.com/vb.net/vb.net\\_overview.htm](https://www.tutorialspoint.com/vb.net/vb.net_overview.htm)

<https://www.javatpoint.com/vb-net>

[https://www.tutorialspoint.com/vb.net/vb.net\\_basic\\_syntax.htm](https://www.tutorialspoint.com/vb.net/vb.net_basic_syntax.htm)

<https://www.guru99.com/vb-net-introduction-features.html>

<https://www.c-sharpcorner.com/blogs/advantages-of-vb-net-c-sharpnet1>

<https://stackoverflow.com/questions/2307530/keeping-vb-net-vs-learning-c>

<https://www.mrlacey.com/2021/03/vbnet-is-dead-one-not-dead.html>



# Thanks for your attention

*thanks for coming!  
we appreciated.*