



AKDENİZ UNIVERSITY
COMPUTER ENGINEERING

PROGRAMMING LANGUAGES
VISUAL BASIC

by
Fatma ERKAN
Tuğba GÜLER

Supervised by
Murat AK

Table of Contents

1. Visual Basic	3
1.1 Introduction.....	3
1.2 Authors	3
1.3 History of the Language	3
1.4 Which Languages Influenced It.....	4
1.5 Why was it Invented	5
1.6 When/Why Shall We Use it.....	6
1.7 How to Setup	7
1.7.1 Windows	7
1.7.2 Linux	12
1.7.3 Mac Os	13
1.8 Usage.....	17
1.9 Example Codes.....	22
1.10 Specific Things.....	29
1.10.1 Characteristic Features of the Language	29
1.10.2 Derivative Languages	30
References	32

1. VISUAL BASIC

1.1 INTRODUCTION

Visual Basic is a high-level programming language and fully compatible with Windows which is one of the most famous operating systems. It is derived from Basic (Beginners-All Purpose Symbolic Instruction Code) which is an easy and base programming language. Visual Basic is a high-level programming language that contains an object-oriented approach. It is developed for some reasons as easy to learn and use, enables RAD (Rapid Application Development) of GUI (Graphical User Interface), access databases with Data Access Objects, etc. Using drag and drop logic, it is possible to implement an application. With this feature, it contains tools collection to implement a program instead of writing codes line by lines using an editor to generate visually. In Visual Basic, implementation is done in a graphical environment. Developers click objects randomly, each object implemented independently. For this reason, each object has different actions.

An object is a thing that can be seen, used in object-oriented programming. Applications that are implemented by Windows environments use objects such as text boxes, buttons, etc. Every object in Visual Basic programming language is generated from a class. The class that belongs to an object includes the behavior and look of the object. An object that is created from its class is called an instance of the class.

1.2 AUTHOR(S)

Alan Cooper is an American software designer and programmer. He is recognized as “Father of Visual Basic”. He is a pioneer of the modern computing era. He created the programming language Visual Basic and wrote industry-standard books on design practice like “About Face.”

1.3 HISTORY OF THE LANGUAGE

The first version of VB 1.0 of Visual basic released in 1991. The creation of the user interface with drag and drop design inspired a beta generator developed by Alan Cooper on Cooper’s company Tripod.

Microsoft signed a contract with Cooper and its partners to create a Tripod on a programmable system for Windows 3.0. This system was developed under the code name Ruby, which has no relation to the Ruby Programming Language.

Tripod did not have any programming language at all. Microsoft then decided to use Ruby in combination with basic language to develop visual basic.

Ruby’s interface contributed to the “visual” component of the Visual Basic programming language. This was later combined with the Embedded BASIC engine that was developed for Microsoft’s abandoned “Omega” database system.

Version 5.0, in February 1997, Microsoft released a visual basic compatible only with 32-bit versions of Microsoft Windows. Programmers who prefer to write programs in 16-bit could do this in versions between 4.0 and 5.0. Besides, programs written in Visual Basic 5.0 can be easily converted to Version 4.0 programs. Version 5.0 also can compile with Windows native execution code and introduce custom user controls.

Visual Basic 6.0 was introduced in the middle of 1998. This version also came with many enhancements, including its stunning ability to build web-based applications. Extended support for Visual Basic 6.0 was stopped in March 2008. However, the basic parts of the Visual Basic 6 development environment work on all 32-bit Microsoft Windows, including Windows 8.1.

After the cessation of mainstream and extended support for Visual Basic 6.0 caused several programs to show concern. Community members then created users lobby and a petition was signed by them. The main purpose of this petition was to ensure the survival of the product. However, the petition did not achieve its purpose effectively.

1.4 WHICH LANGUAGES INFLUENCED IT

Visual basic was influenced by QuickBasic, Basic, Algol and Fortran languages. Figure 1 shows which languages are affected by each other.

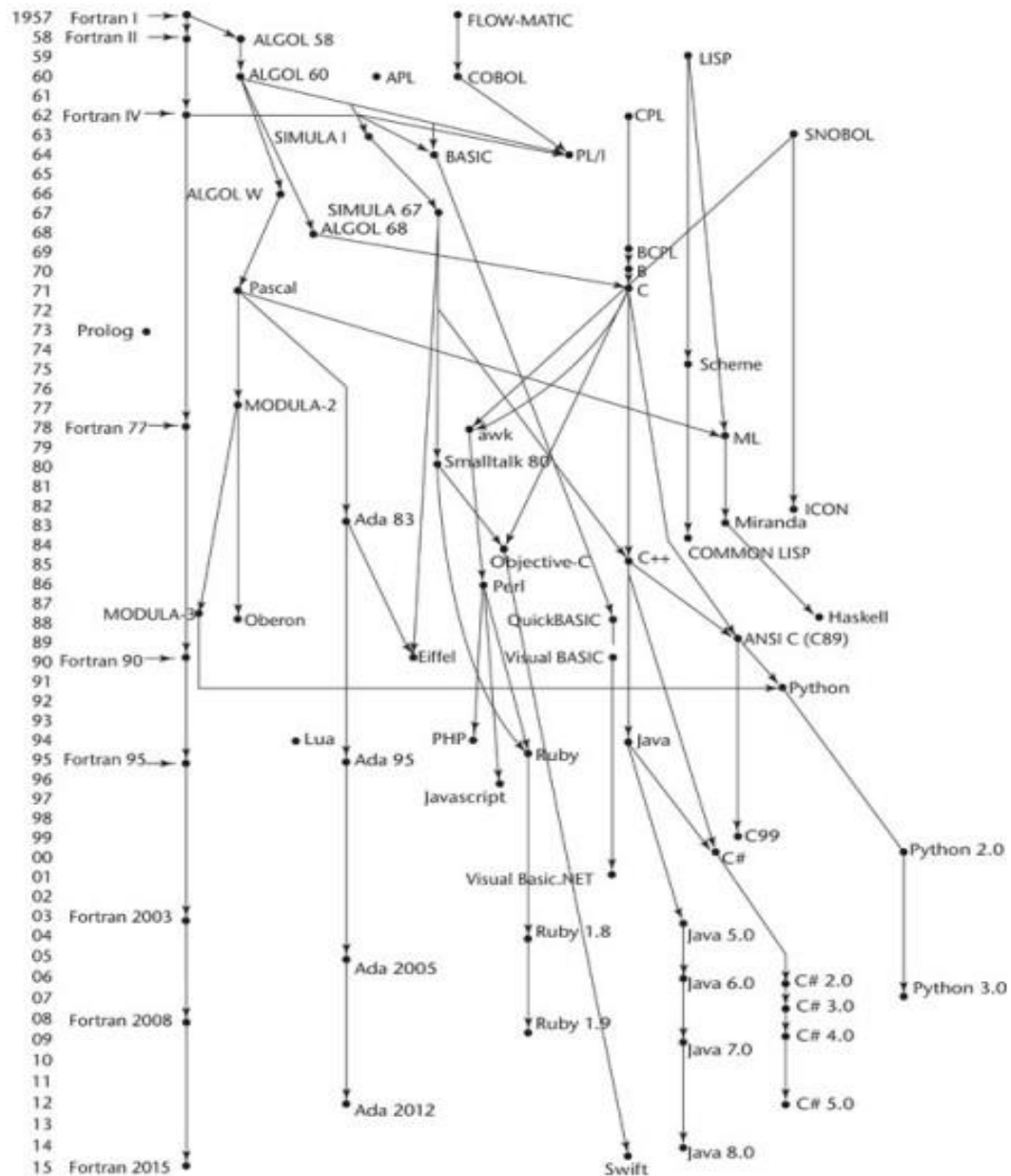


Figure 1. Table of languages influenced by each other

1.5 WHY WAS IT INVENTED

In 1988, Alan Cooper created a visual programming language which code called a "Ruby" that allowed Windows users to create shells like "Finder". It was called "a shell construction set". After Ruby was shown to Bill Gates, Microsoft bought it. Microsoft initially decided not to release the product as a shell for users. However, transform it into a professional development tool for the QuickBASIC programming language named Visual Basic. It is widely used for business application development for Windows computers.

The dynamically installable control facility, known as Cooper's "VBX" interface, was a well-known component of "Ruby". This innovation allowed any 3rd party developer to write a widget as a DLL, put it in the Visual Basic directory, and Visual Basic to find, communicate, and offer it to the user as a seamless part of the program.

1.6 WHEN/WHY SHALL WE USE IT

Visual Basic is considered one of the most powerful programming languages as Microsoft's first visual development tool. Compared to other computer programming languages like C, C ++, it is easy to learn and understand.

Visual Basic is a widely understood high-level programming language written using simple English-like words and syntax. It is an interpreted language; It can be run immediately after the code is written.

Having an interpreter makes it easier to use, as there is no need for the computer to compile the code, possibly finding errors to fix, before the compiled version can be run.

The Visual Basic programming language allows programmers to create software interface and codes in an easy-to-use graphical environment.

It allows programmers to quickly develop window-based applications. It also allows programmers to use ActiveX controls and various objects. It helps in accessing databases using ADO.

Although it is mostly used to develop applications, it is also suitable for game development. Compared to other languages, Visual basic may be slower than other languages. However, it is flexible, and it is stated that things that are difficult in other languages are relatively easy and comfortable in a visual basic programming language.

Visual Basic is one of the most popular programming languages. It has many accessible resources.

Another thing that stands out in visual basic programming is that the VB structure is designed to allow programmers to create executable code - Exe files.

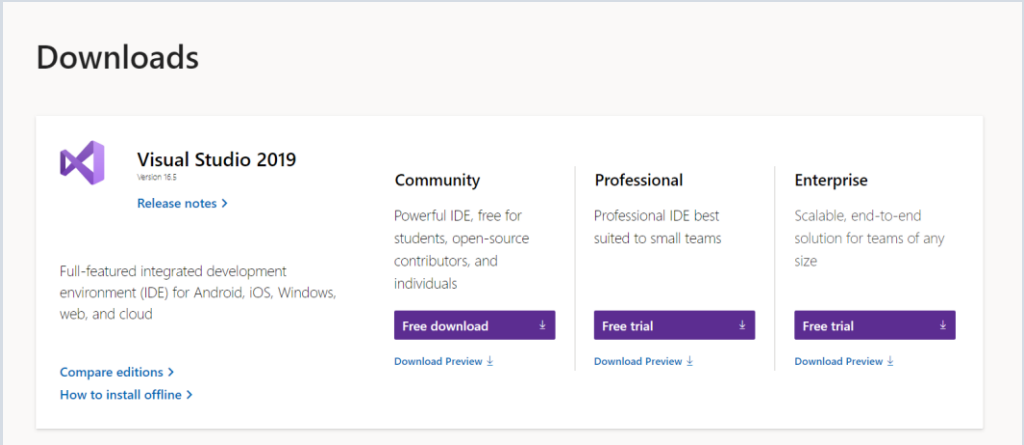
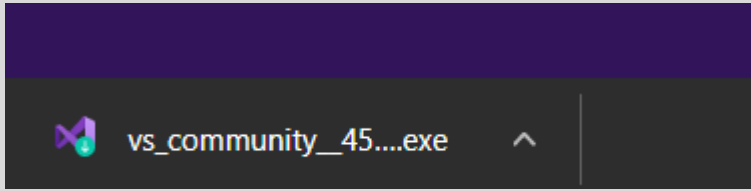
It enables programmers to develop programs that can be used as a front end in databases. Also, with the help of visual basic tools, someone can change abstract ideas into programs or all software and allow them to modify Programs appropriately.

Visual Basic is a language with a powerful front-end tool that can achieve simple and complex business needs effectively and efficiently.

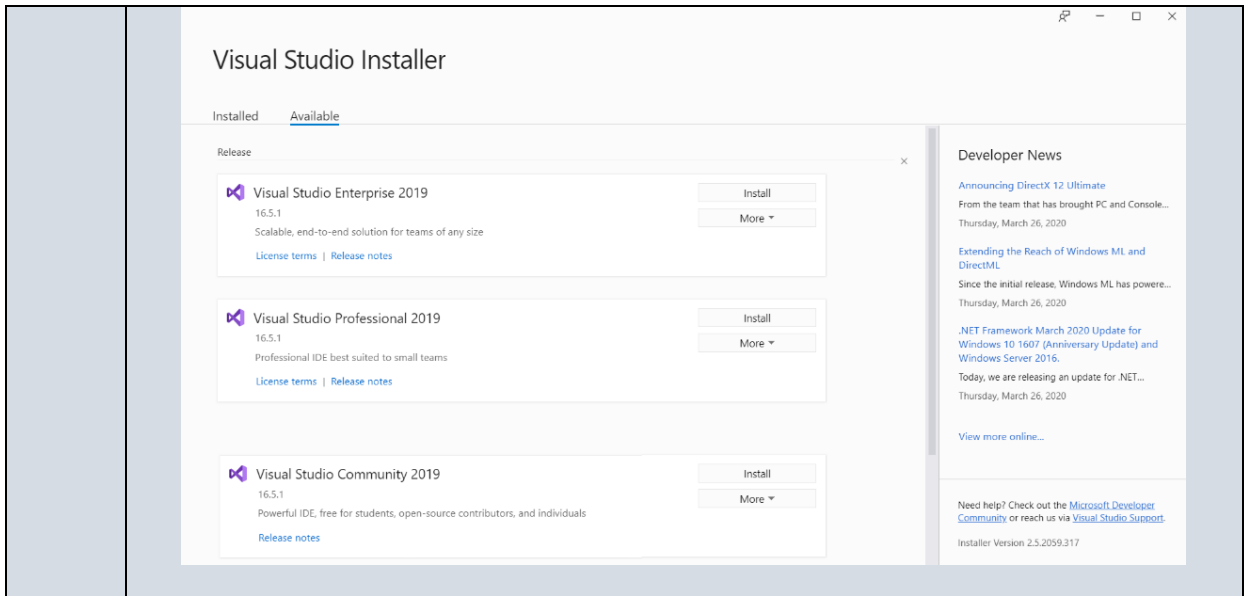
1.7 HOW TO SETUP

To create applications with Visual Basic programming language, it is needed to install the Visual Basic IDE (Integrated Development Environment). Visual Studio which is developed by Microsoft is the most known IDE for Visual Basic. Visual Studio IDE also provides to write code with C++, C#, F#, J#, and Visual Basic programming languages.

1.7.1 WINDOWS

S T E P 1	<p>Visual Studio is downloaded the link below. https://visualstudio.microsoft.com/downloads/</p>  <p>Visual Studio 2019 Community Edition is a free edition. However, Visual Studio 2019 Professional and Visual Studio 2020 Edition has 30 days free trial. The first step is to install the Visual Studio 2019 Community Edition.</p>
When clicking the “Free Download” button, the downloaded exe file is shown on the browser’s downloaded files.	S T E P 2
	 <p>The next step is to click the downloaded exe file.</p> <p>In the next screen, click the “Continue” button to accept Microsoft Privacy Statements and Microsoft Software License Terms.</p>

<div> <div>S</div> <div>T</div> <div>E</div> <div>P</div> <div>3</div> </div>	<div> <div> <div>Visual Studio Installer</div> <div> <div>Before you get started, we need to set up a few things so that you can configure your installation.</div> <div> <div>To learn more about privacy, see the Microsoft Privacy Statement.</div> <div>By continuing, you agree to the Microsoft Software License Terms.</div> </div> <div>Continue</div> </div> </div> </div>
<div>Visual Studio starts downloading the initial files. Download speed depends on internet connection.</div>	<div> <div>S</div> <div>T</div> <div>E</div> <div>P</div> <div>4</div> </div>
<div> <div>S</div> <div>T</div> <div>E</div> <div>P</div> <div>5</div> </div>	<div>In the next screen click the “Install” button which is located beside the Visual Studio Community Edition 2019.</div>



STEP 6

Visual Basic provides to develop applications for many platforms:

1. To implement a desktop application with Visual Basic, select “.Net desktop development”,
2. To create a Universal Windows Platforms, “Universal Windows Platform development” option is available,
3. To develop Office and SharePoint, select “Office/SharePoint development” as shown below.

Modifying — Visual Studio Community 2019 — 16.5.1

Workloads Individual components Language packs Installation locations

Gaming (2)

- ☐ **Game development with Unity**
Create 2D and 3D games with Unity, a powerful cross-platform development environment.
- ☒ **Game development with C++**
Use the full power of C++ to build professional games powered by DirectX, Unreal, or Cocos2d.

Other Toolsets (6)

- ☒ **Data storage and processing**
Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.
- ☒ **Data science and analytical applications**
Languages and tooling for creating data science applications, including Python and F#.
- ☐ **Visual Studio extension development**
Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.
- ☒ **Office/SharePoint development**
Create Office and SharePoint add-ins, SharePoint solutions, and VSTO add-ins using C#, VB, and JavaScript.
- ☐ **Linux development with C++**
Create and debug applications running in a Linux environment.
- ☒ **.NET Core cross-platform development**
Build cross-platform applications using .NET Core, ASP.NET Core, HTML/JavaScript, and Containers including Docker...

Location
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.

Total space required 475 MB

Installation details

- > Visual Studio core editor
- > ASP.NET and web development *
- > Universal Windows Platform development *
- > .NET Core cross-platform development
- > Desktop development with C++ *
- > Node.js development *
- > Game development with C++ *
- > Data science and analytical applications *
- > Data storage and processing *
- > Office/SharePoint development *

Included

- ✓ Office Developer Tools for Visual Studio
- ✓ .NET Framework 4.5 targeting pack
- ✓ .NET Framework 4.7.2 development tools
- ✓ Developer Analytics tools
- ✓ IntelliCode

Optional

- ☒ Visual Studio Tools for Office (VSTO)
- ☒ Web Deploy
- ☐ .NET Framework 4.6.1 development tools

Then click the “Modify” button to install it.

**S
T
E
P
7**

When the download is done, click the “Launch” button which is placed beside Visual Studio Community 2019.

Visual Studio Installer

Installed Available

Visual Studio Community 2017 15.9.17 Update available 15.9.21 View details	<input type="button" value="Update"/> <input type="button" value="Launch"/> <input type="button" value="More ▾"/>
Visual Studio Community 2019 16.5.1 Powerful IDE, free for students, open-source contributors, and individuals Release notes	<input type="button" value="Modify"/> <input type="button" value="Launch"/> <input type="button" value="More ▾"/>

Developer News

[Announcing DirectX 12 Ultimate](#)
From the team that has brought PC and Console...
Thursday, March 26, 2020

[Extending the Reach of Windows ML and DirectML](#)
Since the initial release, Windows ML has powere...
Thursday, March 26, 2020

[.NET Framework March 2020 Update for Windows 10 1607 \(Anniversary Update\) and Windows Server 2016](#)
Today, we are releasing an update for .NET...
Thursday, March 26, 2020

[View more online...](#)

Need help? Check out the [Microsoft Developer Community](#) or reach us via [Visual Studio Support](#).

Installer Version 2.5.2059.317

In the Visual Studio IDE, click the “Create new project” button to create a new project with Visual Basic programming language.


Visual Studio 2019

Open recent


Today


 AU.Optime.sln 3/28/2020 4:06 PM
D:\AU.Optime

This month

 PanCardView.sln 3/20/2020 2:50 PM
E:\Projects\VisualStudio-Projects\PanCardView

Older

 TemplateDeneme.sln 2/4/2020 12:55 PM
E:\Projects\VisualStudio-Projects\TemplateDeneme

 TemplateDeneme2.sln 2/4/2020 12:45 AM
E:\Projects\VisualStudio-Projects\TemplateDeneme2

 deneme.sln 1/29/2020 8:25 PM
D:\deneme

Older

 CV.sln 1/5/2020 7:25 PM

Get started



Clone or check out code

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

[Continue without code →](#)

S
T
E
P

8

Create a new project

Search for templates (Alt+S)




[Clear all](#)

Recent project templates

 Mobile App (Xamarin.Forms) C#


 ASP.NET Web Application (.NET Framework) C#

 Class Library (.NET Framework) C#

 ASP.NET Web Application (.NET Framework) Visual Basic

 ASP.NET Core Web Application C#

 Class Library (.NET Core) C#

 Console App (.NET Core) C#



WPF App (.NET Framework)

Windows Presentation Foundation client application

Visual Basic Windows Desktop

[New](#)



Windows Forms App (.NET Framework)

A project for creating an application with a Windows Forms (WinForms) user interface

Visual Basic Windows Desktop

[New](#)



Windows Forms Control Library (.NET Framework)

A project for creating controls to use in Windows Forms (WinForms) applications

Visual Basic Windows Desktop Library

[New](#)



WPF Browser App (.NET Framework)

Windows Presentation Foundation browser application

Visual Basic Windows Desktop

[New](#)



WPF User Control Library (.NET Framework)

Windows Presentation Foundation user control library

Visual Basic Windows Desktop Library

[New](#)

[Back](#)

[Next](#)

S
T
E
P

9


In the Create new project screen, select the programming language which is Visual Basic that is located below the search bar. There are many project templates to create applications for different platforms.

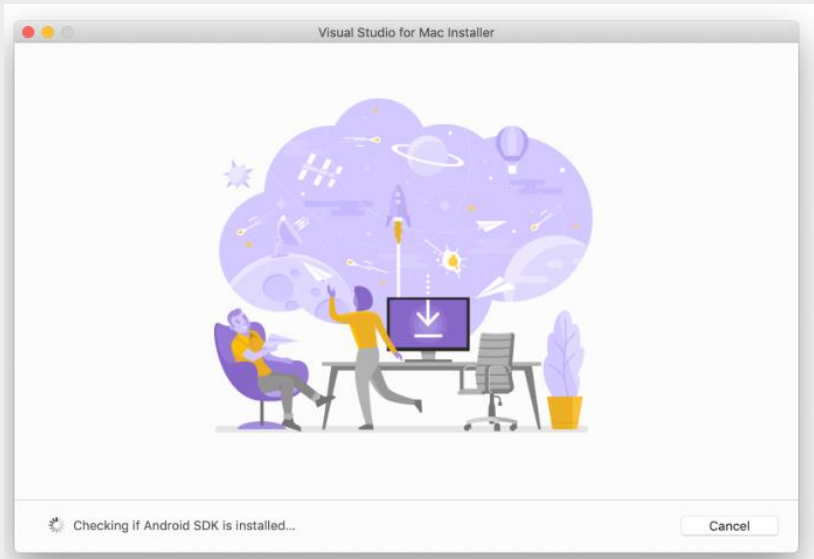
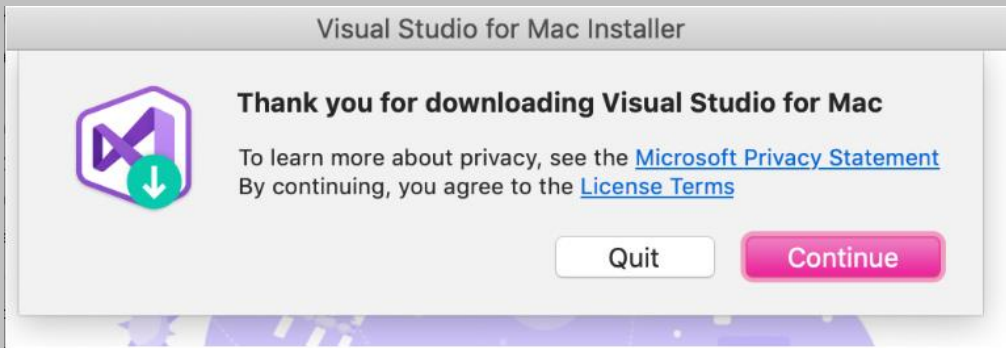
1.7.2 LINUX

S T E P 1	<p>To compile codes that are written by Visual Basic programming language, Mono Visual Basic Compiler, and Visual Basic Runtime are required in Linux operating system. These requirements are available in “Mono 1.2.3”. Packages can be downloaded at the link below</p> <p>https://www.mono-project.com/download/stable/#download-lin</p> <p>To download for Ubuntu 18.04 the commands are used as shown below to package repository hosts that the packages are needed.</p> <pre> sudo apt install gnupg ca-certificates sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF echo "deb https://download.mono-project.com/repo/ubuntu stable-bionic main" sudo tee /etc/apt/sources.list.d/mono-official-stable.list sudo apt update </pre> <p>To install for Ubuntu 16.04, the shared commands are used as listed below to repository hosts.</p> <pre> sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 3FA7E0328081BFF6A14DA29AA6A19B38D3D831EF sudo apt install apt-transport-https ca-certificates echo "deb https://download.mono-project.com/repo/ubuntu stable-xenial main" sudo tee /etc/apt/sources.list.d/mono-official-stable.list sudo apt update </pre>
<p>To install Mono to compile code the command is used.</p> <pre> sudo apt install mono-devel </pre>	S T E P 2
S T E P	<p>To install code which is available in the git from the module “mono-basic”,</p> <pre> \$./configure </pre>

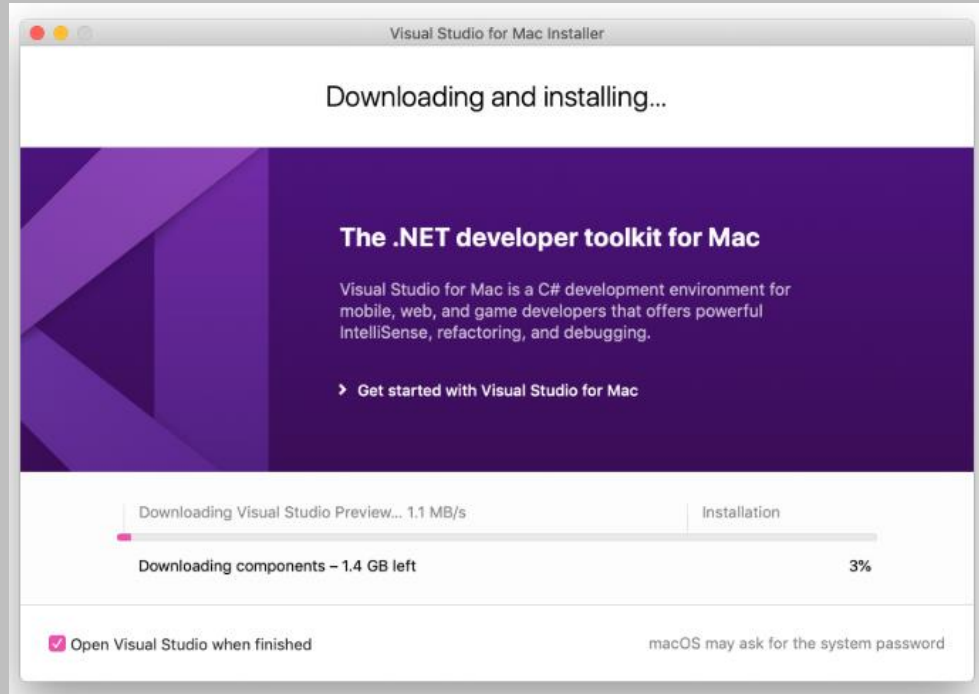
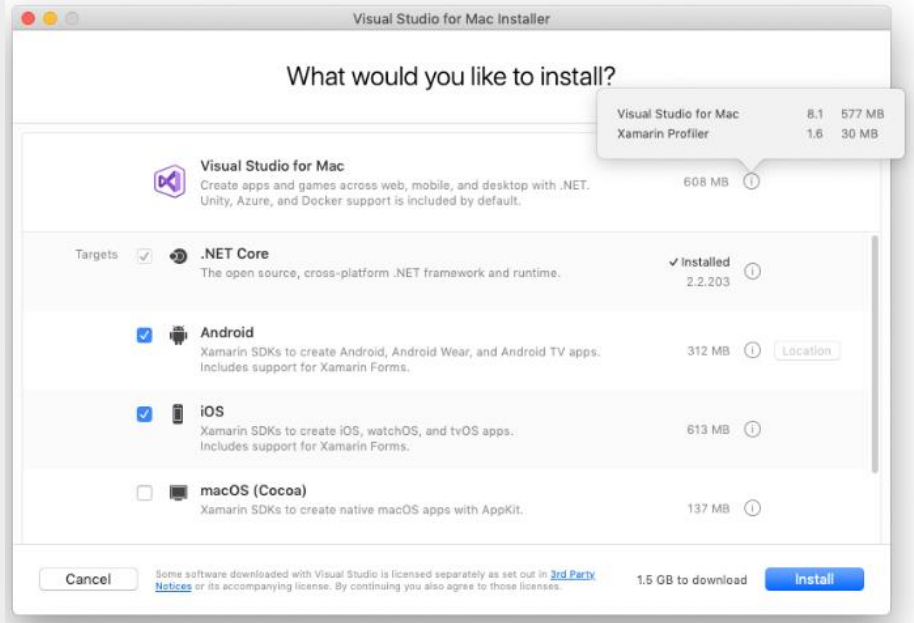
3	<code>\$ make && sudo make install</code>	STEP 4
	commands are used.	
Then the command “vbnc” is available to build applications that are written by Visual Basic programming language.		

1.7.3 MAC OS

STEP 1	<p>To start developing with Visual Basic programming language in macOS, download the Visual Studio Installer from the link below.</p> <p>https://visualstudio.microsoft.com/vs/mac/</p> <div data-bbox="386 810 1412 1310"> <h2>Visual Studio for Mac</h2> <p>Develop apps and games for iOS, Android, and web using .NET</p> <p>Read more about activating your license</p> <p>Download Visual Studio for Mac ↓</p> </div>	STEP 2
	<p>After clicking the “Download Visual Studio for Mac” button, find the “VisualStudioforMacInstaller.dmg” file in the downloaded files.</p> <div data-bbox="621 1486 979 1841">  </div>	

STEP 3	<p>The warning screen may be presented which is about downloading the application from the internet. Open it.</p> 	
	<p>Visual Studio for Mac Installer will be asked to accept Microsoft Privacy Statement and License Terms. Click “Continue”.</p> 	STEP 4
	<p>There are some options to the installer screen. To use Visual Basic programming language, select the .Net Core framework then click the button that is called “Install”.</p>	

S
T
E
P
5



S
T
E
P
6

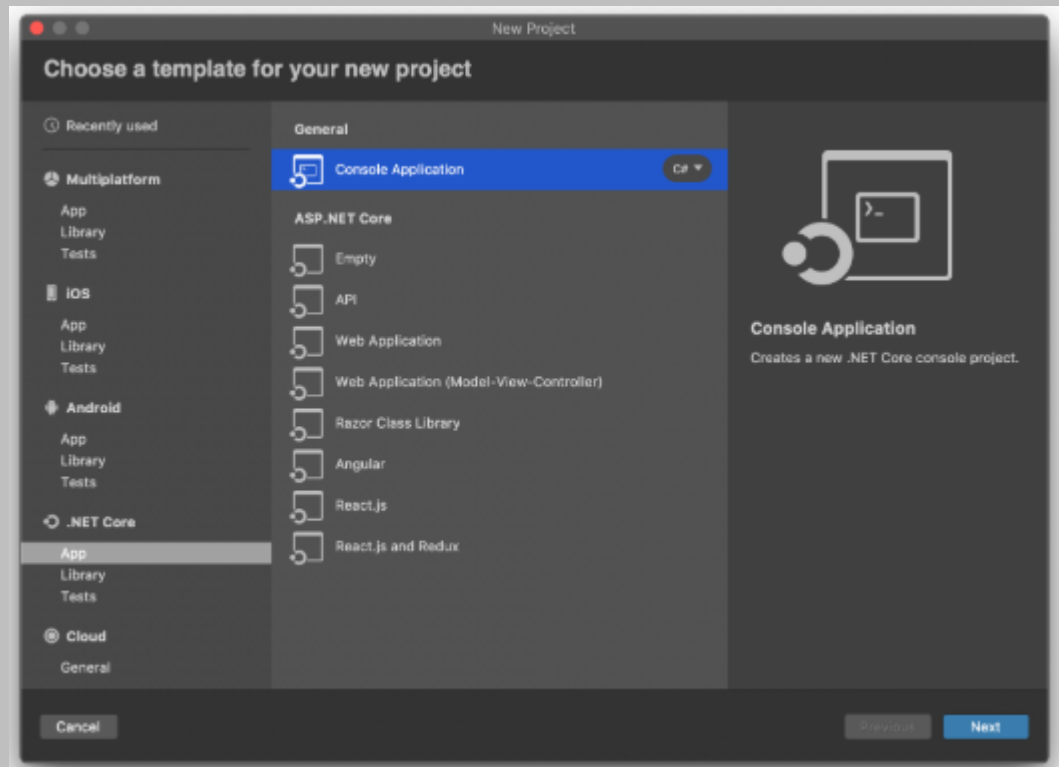
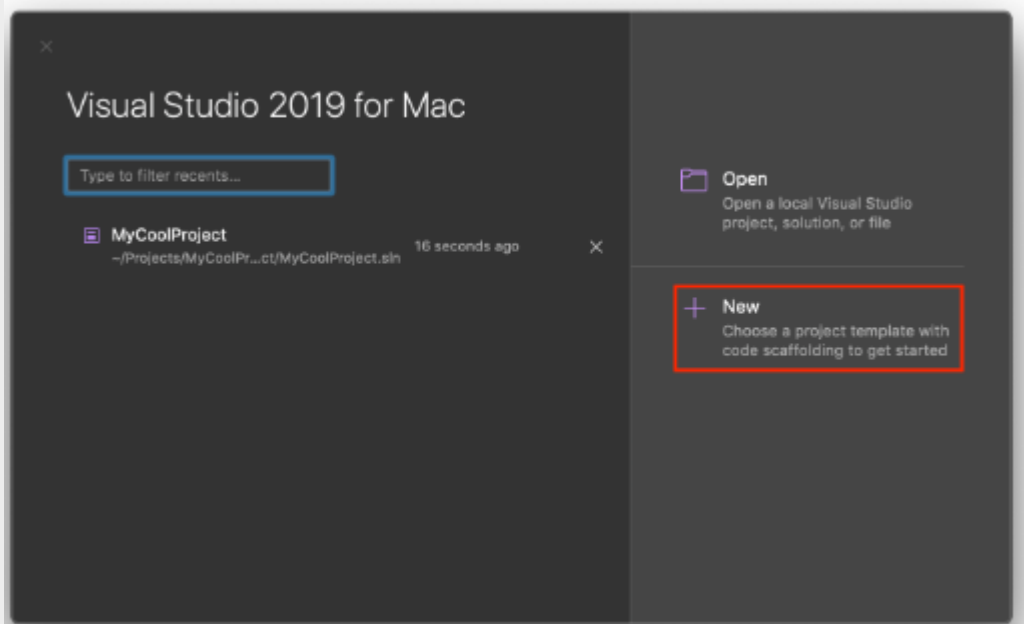
The installer screen displays the progress of installing and downloading.

To create an application in macOS with Visual Basic programming language open the Visual Studio 2019 for Mac IDE.

Then click the “New” button to create a new project.

S
T
E
P

7



S
T
E
P

8

Select the project template to develop an application for different platforms and the right side of the project template there is a dropdown menu which enables to change programming language. After selecting a template, change programming language with Visual Basic.

1.8 USAGE

Visual Studio 2019 provides a Toolbox window to drag and drop controls to the project which is developed using Visual Basic programming language. To open the Toolbox window, click “Toolbox” on the View menu as shown in Figure 1.7.1. Using the Toolbox window, GUI of the application will be created easily.

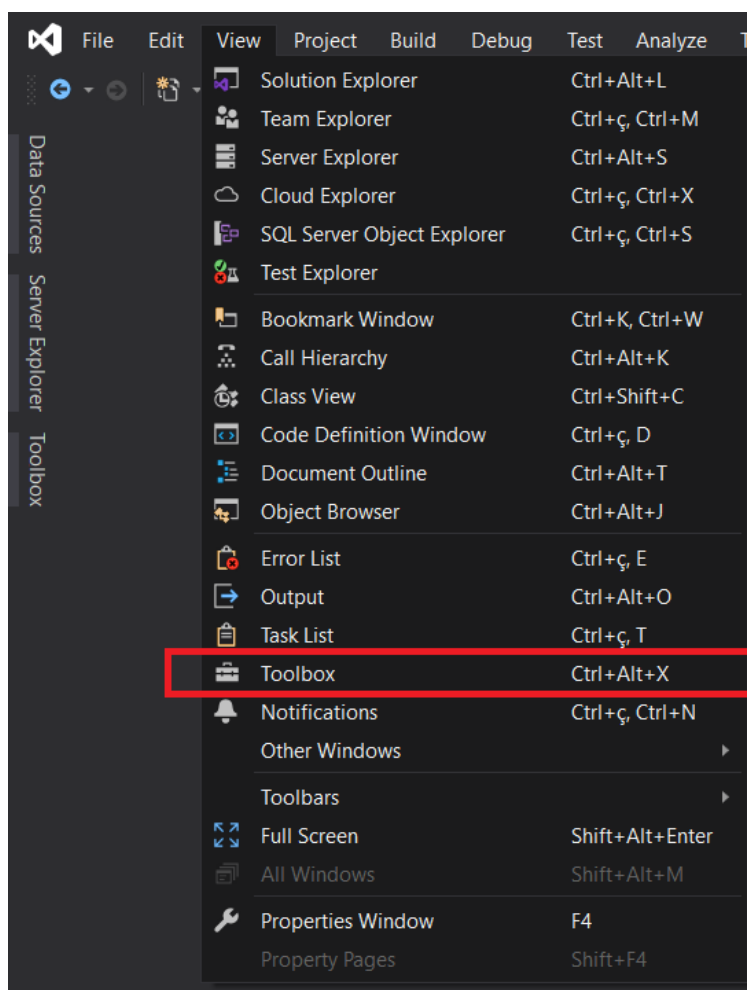

















Figure 1.7.1: Toolbox Window

In the Toolbox window, controls are located as seen in Figure 1.7.2. Controls, which are known as objects, are added dragging selected objects with a mouse pointer. When dragging the control to the form, Visual Basic is automatically created its instance to the Form's class. Many controls occurred in the Toolbox. Most commons are Button, CheckBox, ComboBox, Label, ListBox, MonthCalendar, ProgressBar, RadioButton, TextBox etc.

Button	Icon	Description
	TextBox	It contains text that can be changed by the user.
	Label	It allows to have text that is shown to the user, but the user cannot change.
	ComboBox	It provides to draw a combination Listbox and text box. Users can select one of the items from the list.
	ListBox	It is used to display many items at the same time which a user can select. It has a scroll if many items occur.
	CheckBox	It is a box to indicate whether something is false or true. It is possible to select multiple choices.
	OptionButton (RadioButton)	It allows for displaying multiple choices. However, the user can select only one of them.
	ToggleButton	It is a button to toggle on and off.
	Select Objects (Pointer)	It is the item that does not draw control. When it is selected, it can only resize or move a control.
	Frame	It provides to build a group for controls. To group controls, first, drag the frame to the Windows Form. Then drag controls inside of the frame.
	CommandButton	To create a button to choose to carry out a command.
	TabStrip	It provides to define multiple pages for the same form.
	MultiPage	It presents multiple screens of knowledge.
	ScrollBar	It allows to graphical tool to navigate through a long list of items or used to large amounts of data.
	SpinButton	It is used with another control to decrement or increment numbers.
	Image	It is used to display images from a bitmap, icon or metafile. The image will be displayed Image control.

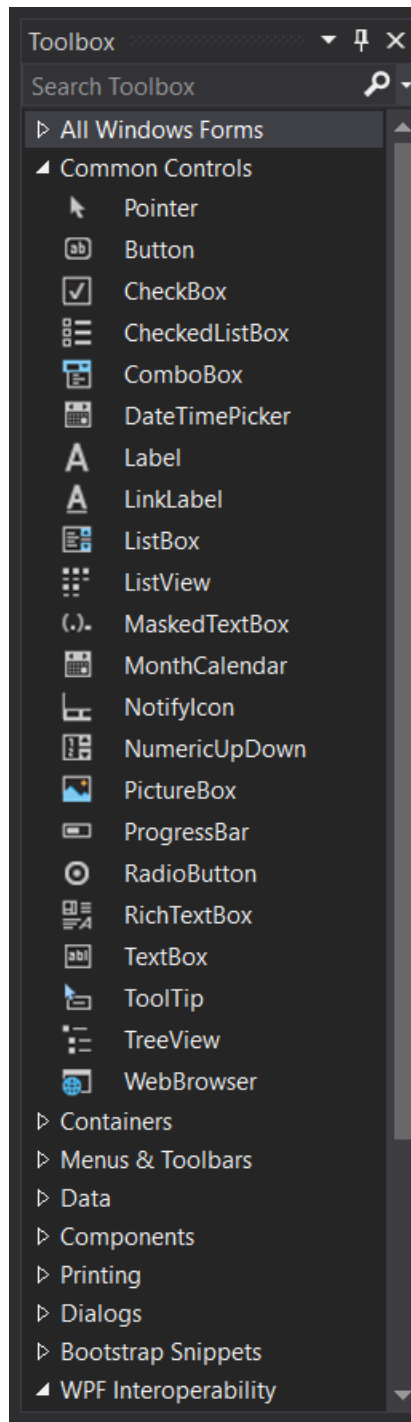


Figure 1.7.2 Toolbox controls

The designer window is where the GUI of application is created. A windows form is shown in the designer window in Figure 1.7.3. The form is an object to create user interfaces. The form is an instance of Windows Form class.

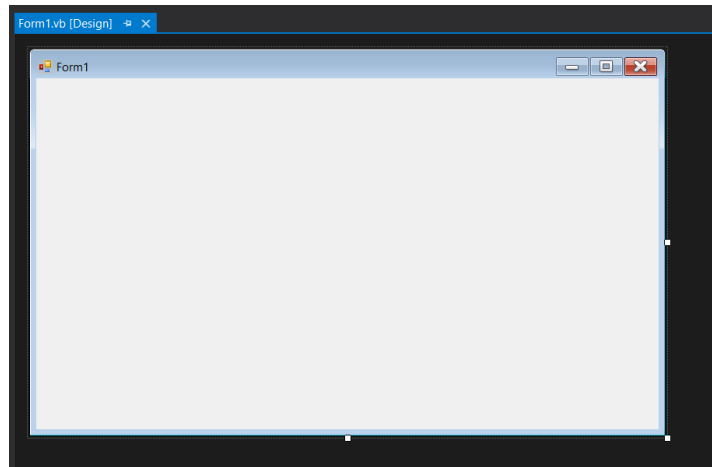


Figure 1.7.3 Designer Window

Each control has attributes that identify behavior and appearance. These attributes called properties that are listed in the Properties window shown in Figure 1.7.4.

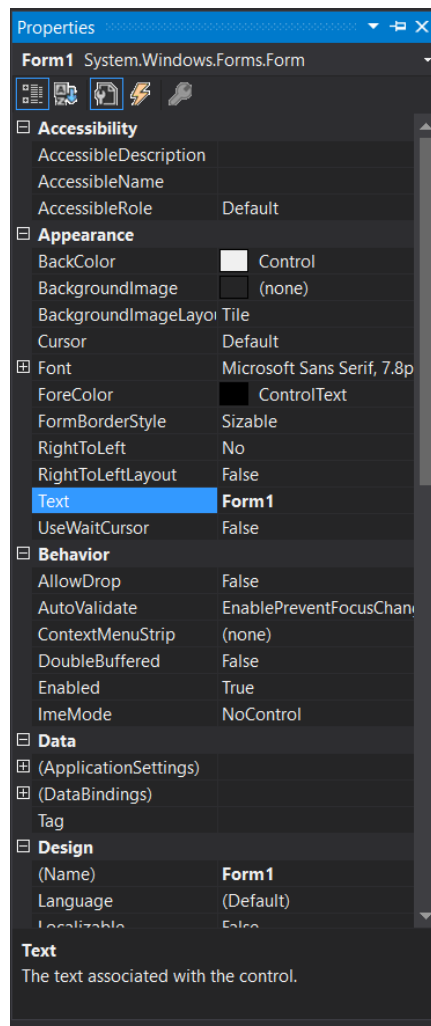


Figure 1.7.4 Properties Window

To change the properties of a control, select the control in the designer window and open the properties window. General attribute examples are Name, Text, Text Color, Background Color, Font, ForeColor, etc.

Example “Login Form” is shown in Figure 1.7.5. It is created using two Labels to show text called “Email” and “Password”. To get information from the user, there are two TextBox exist. One of them is for “Email” and the other one is for “Password”. Besides, there is a Login button that exists to click.

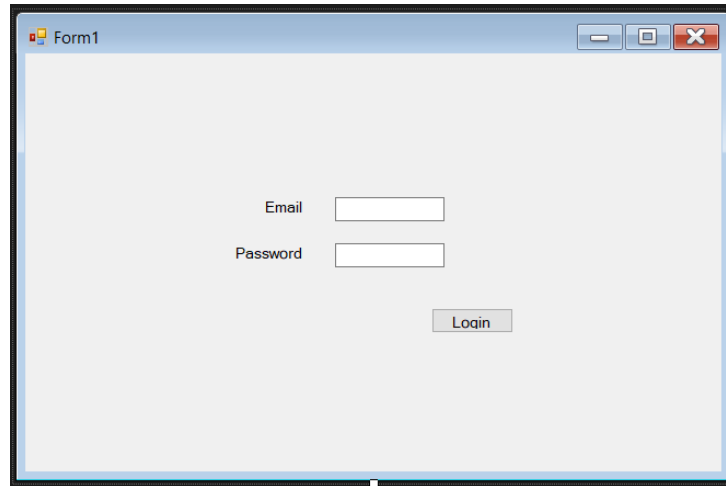


Figure 1.7.5 Login Form Example

To get an event of any control double click to control and it is automatically generated in the file with .vb extension as shown in Figure 1.7.6. The event belongs to Login Button in the Form.

```
1 reference
Public Class Form1
    0 references
    Private Sub btnLogin_Click(sender As Object, e As EventArgs) Handles btnLogin.Click
        ...
    End Sub
End Class
```

Figure 1.7.6 Visual Basic file of the Form.

In addition to this, we can call other controls with their names as shown in Figure 1.7.7. TextBox of Email is called as a statement “txtEmail.Text” where the “txtEmail” is the name of the control which can be changed in the Properties Window. With using “.Text” property, it is possible to get the text of the Email TextBox. Then it is controlled in the if-else statement to show the message to the user.

```

1 reference
Public Class Form1
    0 references
    Private Sub btnLogin_Click(sender As Object, e As EventArgs) Handles btnLogin.Click
        Dim email As String = txtEmail.Text
        Dim password As String = txtPassword.Text

        If email = "fatma" And password = "erkan" Then
            MessageBox.Show("Successful!")
        Else
            MessageBox.Show("Please try again!")
        End If
    End Sub
End Class

```

Figure 1.7.7 Login Button event example

1.9 EXAMPLE CODES

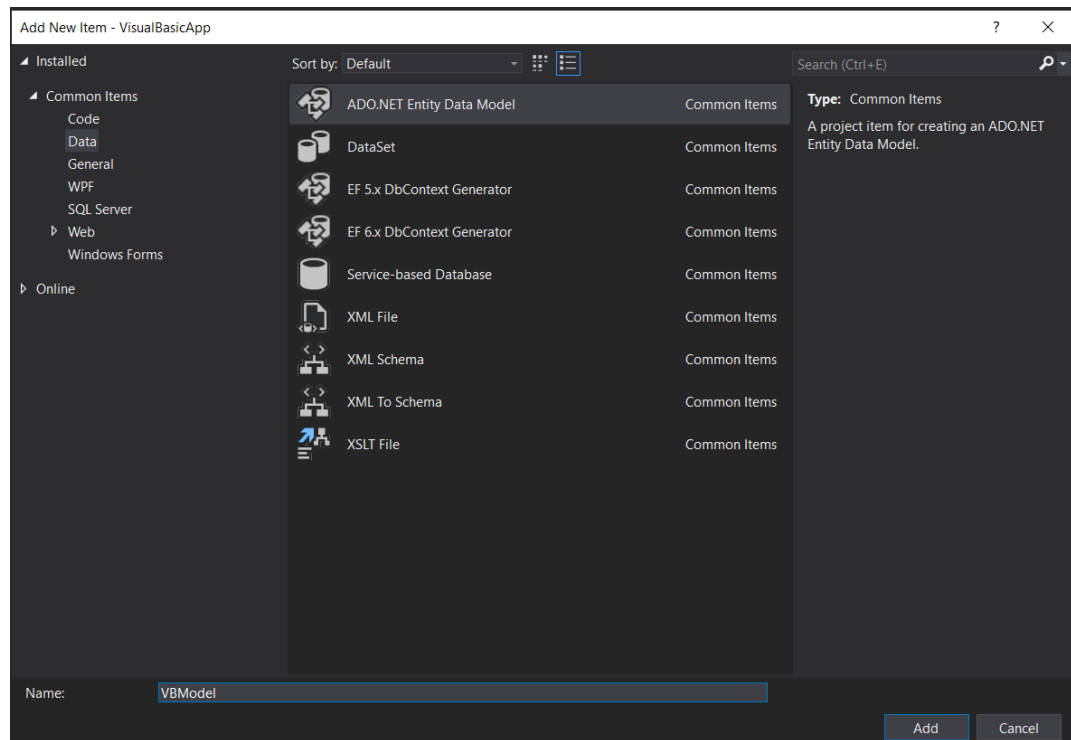


Figure 1.9.1

In Figure 1.9.1, the database model was added to the project using Ado.Net Entity Data Model.

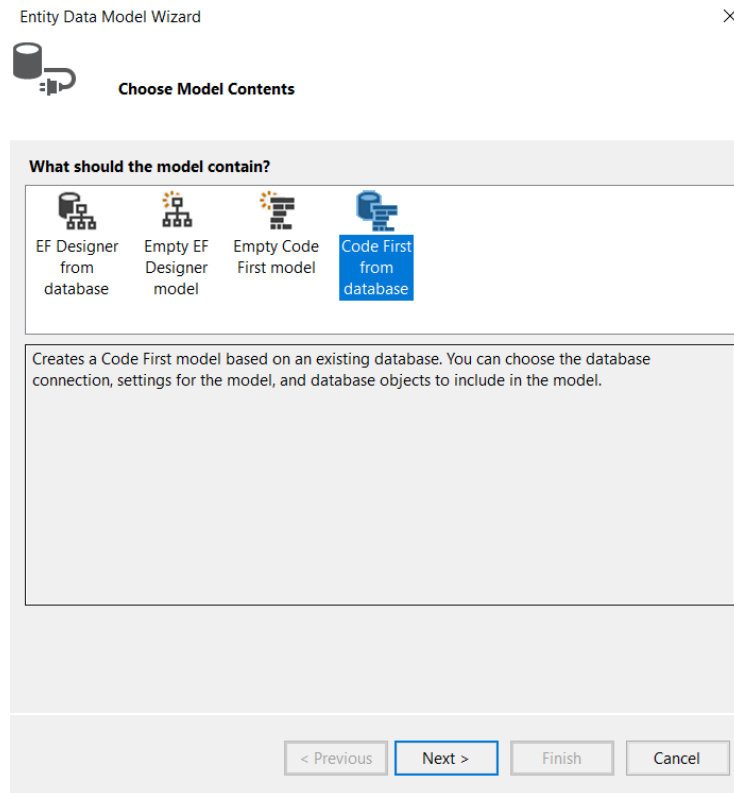


Figure 1.9.2

Figure 1.9.2 shows the Code First approach of the database connection.

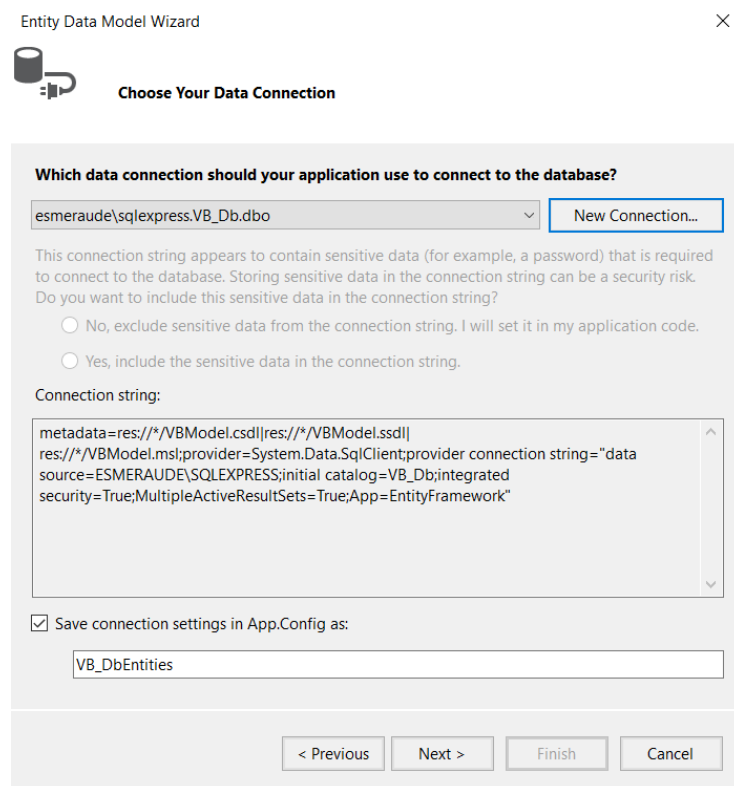


Figure 1.9.3

Figure 1.9.3 is where connectionString generated to get database information.

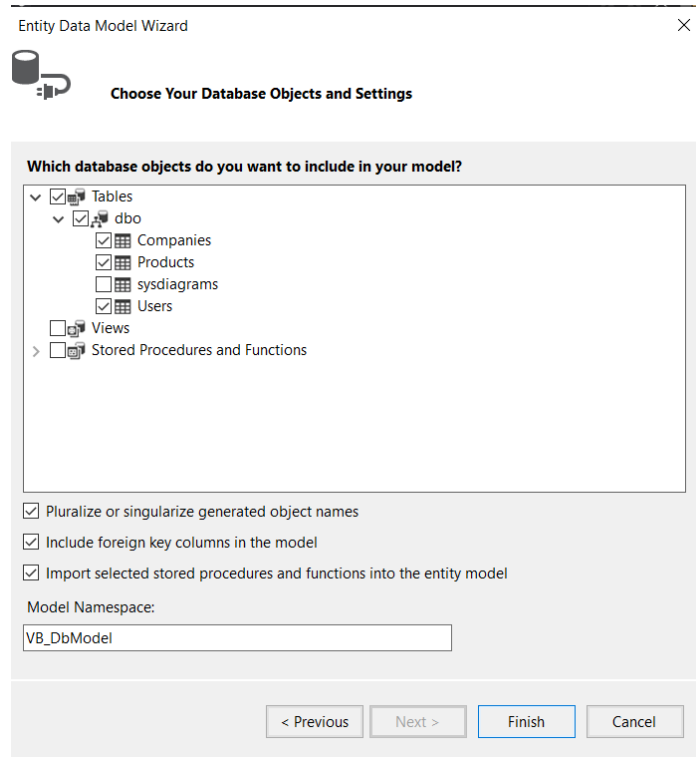


Figure 1.9.4

In Figure 1.9.4 Database tables are shown. To use these tables in the project, select tables.

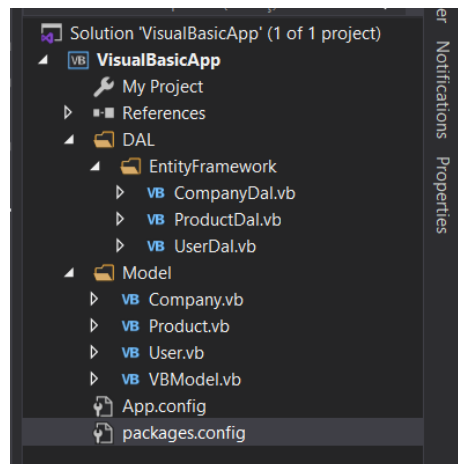


Figure 1.9.5

Figure 1.9.5 shows that the views of tables in the project which is implemented by Visual Basic programming language. On the other hand, there is a folder that contains DAL which is called "Data Access Layer". DAL is configured for the CRUD operations (Create/Read/Update/Delete) of the tables.


```

1 Imports System
2 Imports System.Collections.Generic
3 Imports System.ComponentModel.DataAnnotations
4 Imports System.ComponentModel.DataAnnotations.Schema
5 Imports System.Data.Entity.Spatial
6
7 Partial Public Class Product
8     Public Property ProductId As Integer
9
10    <StringLength(100)>
11    Public Property ProductName As String
12
13    Public Property CompanyId As Integer?
14
15    Public Overridable Property Company As Company
16 End Class

```

Figure 1.9.6

Figure 1.9.6 contains a Product class that is the view of the Product table in the project.

```

1 Imports System
2 Imports System.Data.Entity
3 Imports System.ComponentModel.DataAnnotations.Schema
4 Imports System.Linq
5
6 Partial Public Class VBModel
7     Inherits DbContext
8
9     Public Sub New()
10         MyBase.New("name=VBModel")
11     End Sub
12
13     Public Overridable Property Companies As DbSet(Of Company)
14     Public Overridable Property Products As DbSet(Of Product)
15     Public Overridable Property Users As DbSet(Of User)
16
17     Protected Overrides Sub OnModelCreating(ByVal modelBuilder As DbModelBuilder)
18     End Sub
19 End Class

```

Figure 1.9.7

In Figure 1.9.7 there is a VBModel class that is inherited from DbContext class. DbContext class provides to do operations with the database automatically. The class also contains table names as property.

```

1 Imports System
2 Imports System.Collections.Generic
3 Imports System.ComponentModel.DataAnnotations
4 Imports System.ComponentModel.DataAnnotations.Schema
5 Imports System.Data.Entity.Spatial
6
7 Partial Public Class User
8     Public Sub New()
9         Companies = New HashSet(Of Company)()
10     End Sub
11
12     Public Property UserId As Integer
13
14     <StringLength(100)>
15     Public Property Email As String
16
17     <StringLength(100)>
18     Public Property Password As String
19
20     Public Overridable Property Companies As ICollection(Of Company)
21 End Class

```

Figure 1.9.8

Properties of the User table is shown in Figure 1.9.8.

```

VisualBasicApp Company UserId
1 Imports System
2 Imports System.Collections.Generic
3 Imports System.ComponentModel.DataAnnotations
4 Imports System.ComponentModel.DataAnnotations.Schema
5 Imports System.Data.Entity.Spatial
6
7 Partial Public Class Company
8     Public Sub New()
9         Products = New HashSet(Of Product)()
10    End Sub
11
12    4 references
13    Public Property CompanyId As Integer
14    <StringLength(100)>
15    4 references
16    Public Property CompanyName As String
17    1 reference
18    Public Property UserId As Integer?
19    0 references
20    Public Overridable Property User As User
21    1 reference
22    Public Overridable Property Products As ICollection(Of Product)
23 End Class

```

Figure 1.9.9

The view of the Company table is shown in Figure 1.9.9.

```

1 reference
1 Public Class CompanyDal
2     Public model As VBModel
3
4     0 references
5     Public Function Add(company As Company)
6         model.Companies.Add(company)
7         model.SaveChanges()
8     End Function
9
10    0 references
11    Public Function Remove(companyId As Integer)
12        Dim companyModel As Company = model.Companies.Find(companyId)
13        model.Companies.Remove(companyModel)
14        model.SaveChanges()
15    End Function
16
17    0 references
18    Public Function Update(company As Company)
19        Dim editToCompany As Company = model.Companies.Find(company.CompanyId)
20        editToCompany.CompanyName = company.CompanyName
21        model.SaveChanges()
22    End Function
23
24    0 references
25    Public Function GetCompany(companyId As Integer) As Company
26        Dim company As Company = model.Companies.Find(companyId)
27        Return company
28    End Function
29
30    3 references
31    Public Function GetUserCompany(userId As Integer) As Company
32        Dim company As Company = From comp In model.Companies.ToList()
33            Where comp.UserId = userId
34            Select (comp.CompanyId, comp.CompanyName)
35        Return company
36    End Function
37 End Class

```

Figure 1.9.10

Figure 1.9.10 contains CompanyDAL class to define company table operations.

```

1 1 reference
2 Public Class ProductDal
3     Public model As VBModel
4
5     0 references
6     Public Function Add(product As Product)
7         model.Products.Add(product)
8         model.SaveChanges()
9     End Function
10
11     0 references
12     Public Function Remove(productId As Integer)
13         Dim productModel As Product = model.Products.Find(productId)
14         model.Products.Remove(productModel)
15         model.SaveChanges()
16     End Function
17
18     0 references
19     Public Function Update(product As Product)
20         Dim editToProduct As Product = model.Products.Find(product.ProductId)
21         editToProduct.ProductName = product.ProductName
22         model.SaveChanges()
23     End Function
24
25     0 references
26     Public Function GetProduct(productId As Integer) As Product
27         Dim product As Product = model.Products.Find(productId)
28         Return product
29     End Function
30
31     1 reference
32     Public Function GetProducts(companyId As Integer) As List(Of Product)
33         Dim products As List(Of Product) = From prod In model.Products.ToList()
34                                         Where prod.CompanyId = companyId
35                                         Select (prod.ProductId, prod.ProductName)
36         Return products
37     End Function
38 End Class

```

Figure 1.9.11

Figure 1.9.11 contains the ProductDAL class.

```

1 Public Class UserDal
2     Public model As VBModel
3
4     Public Function Add(user As User)
5         model.Users.Add(user)
6         model.SaveChanges()
7     End Function
8
9     Public Function Remove(userId As Integer)
10        Dim userModel As User = model.Users.Find(userId)
11        model.Users.Remove(userModel)
12        model.SaveChanges()
13    End Function
14
15    Public Function Update(user As User)
16        Dim editToUser As User = model.Users.Find(user.UserId)
17        editToUser.Email = user.Email
18        editToUser.Password = user.Password
19        model.SaveChanges()
20    End Function
21
22    0 references
23    Public Function GetUser(userId As Integer) As User
24        Dim user As User = model.Users.Find(userId)
25        Return user
26    End Function
27
28    1 reference
29    Public Function Login(email As String, password As String) As User
30
31        Dim queryResults = From user In model.Users.ToList()
32                        Where user.Password = password
33                        Where user.Email = email
34                        Select (user.UserId, user.Email, user.Password)
35
36        Return queryResults
37    End Function
38 End Class

```

Figure 1.9.12

In Figure 1.9.12, the UserDal class is located.

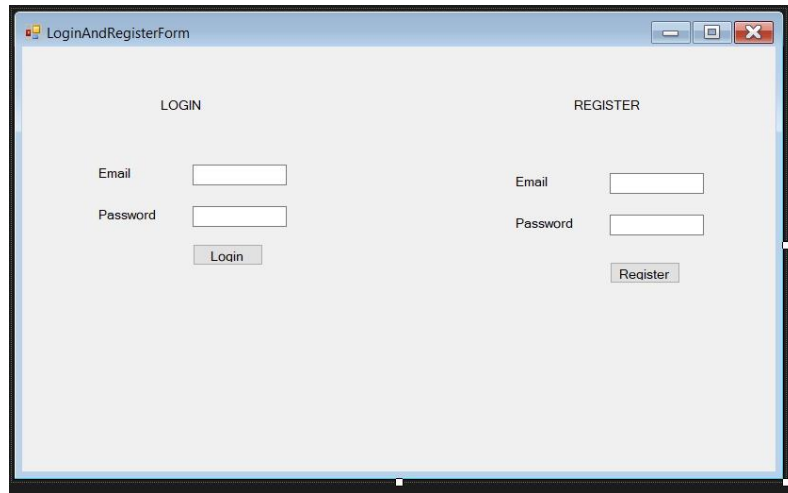


Figure 1.9.13

Login and Register form is shown in Figure 1.9.13. It was created using the Toolbox window of Visual Studio 2019. There are 6 labels to the shown user and 4 textboxes to get information from the user. The login button is located to control login information of the user and the Register button is located to add a user to the user table in the database. Backend codes are shown in Figure 1.9.14

```

1 2 references
2 Public Class LoginAndRegisterForm
3     Dim efUserDAL As UserDal
4     Public Shared userId As Integer
5
6     Private Sub btnLogin_Click(sender As Object, e As EventArgs) Handles btnLogin.Click
7         Dim user As User = efUserDAL.Login(txtLoginEmail.Text, txtLoginPassword.Text)
8         If user.UserId = 0 Then
9             lblInfo.Text = "User cannot found! Please try again."
10        Else
11            MessageBox.Show("Welcome!")
12            userId = user.UserId
13            Dim changeForm As New CompaniesForm
14            changeForm.Show()
15        End If
16    End Sub
17
18    Private Sub btnRegister_Click(sender As Object, e As EventArgs) Handles btnRegister.Click
19        Dim newUser As User
20        newUser.Email = txtRegisterEmail.Text
21        newUser.Password = txtRegisterPassword.Text
22        efUserDAL.Add(newUser)
23        MessageBox.Show("Welcome!")
24        userId = newUser.UserId
25        Dim changeForm As New CompaniesForm
26        changeForm.Show()
27    End Sub
28 End Class

```

Figure 1.9.14

After user login, the Companies Form will be open which contains CompanyName label to show the company name of the user. And then, there is a ListView to show the Product name of the company. Its form is shown in Figure 1.9.15 and the Visual Basic codes are shown in Figure 1.9.16

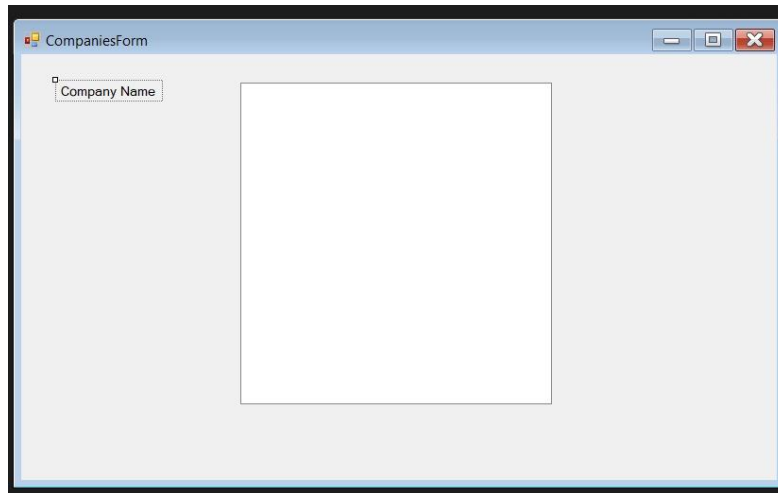


Figure 1.9.15

```

1 Public Class CompaniesForm
2     Dim efCompanyDAL As CompanyDal
3     Dim efProductDAL As ProductDal
4
5     Dim userId As Integer = LoginAndRegisterForm.userId
6
7     0 references
8     Public Sub Main()
9         lblCompanyId.Text = efCompanyDAL.GetUserCompany(userId).CompanyId
10        lblCompanyName.Text = efCompanyDAL.GetUserCompany(userId).CompanyName
11
12        For Each item As Product In efProductDAL.GetProducts(efCompanyDAL.GetUserCompany(userId).CompanyId)
13            lstProducts.Items.Add(item.ProductName)
14        Next
15    End Sub
16 End Class

```

Figure 1.9.16

In the example codes part, a small application was developed to show how Visual Basic codes are generated, how to connect the Database model with the project and how Forms are designed.

1.10 SPECIFIC THINGS

1.10.1 CHARACTERISTIC FEATURES OF THE LANGUAGE

Visual Basic is based on characteristics of BASIC.

- As in the previous BASIC, there is no line number, the code is grouped into subroutines or methods. → Sub ... End Sub.
- Code Expressions do not contain a terminating character other than a line break (carriage return/line feed). Versions since at least VB 3.0 have allowed expressions to be indirectly multi-line by combining strings, or explicitly multi-line using the underscore character (_) at the end of the lines.
- Code comments are made with a single apostrophe (') character. → 'This is a comment
- Loop statement blocks begin and end with keywords → Do ... Loop, While ... End While, For ... Next.
- Multiple variables cannot be assigned. A = B = C does not mean that all values are equal.

- The Boolean data type is stored as a two's complement signed integer. The numerical value of the Boolean constant True is -1. In this structure, -1 evaluates all-1s as binary (Boolean value of True) and 0 all-0s (Boolean value of False).
- Logical and bitwise operators are unified. This is different from some C-derivative languages, for example, Perl with separate logical and bitwise operators. This is one of BASIC's traditional features.
- Variable array base. Arrays are declared by specifying the upper and lower boundaries similar to Pascal and Fortran. It is also possible to use the Option Base statement to set the default lower bound. Using the Option Base statement can confuse when reading Visual Basic code and is best avoided by always explicitly specifying the lower boundary of the array. This lower limit is not limited to 0 or 1 because it can also be set via notification. In this way, both the lower and upper bounds can be programmed. In languages limited to subscript, the lower bound of the array is not variable.
- Relatively strong integration with the Windows operating system and Component Object Model.
- The banker's rounding is rounded as the default behavior when converting real numbers to integers with the Round function. → ? Round (2,5, 0) 2,? The round gives (3.5, 0) 4.
- Integers are automatically raised to the reals in expressions containing the normal division operator (/), so dividing one integer with another produces an intuitively accurate result. VB provides a specific integer division operator (\) that is truncated.
By default, if a variable is not declared or no type of declaration character is specified, the variable is of the Variant type. However, this can be replaced with DefType statements such as DefInt, DefBool, DefVar, DefObj, DefStr. There are 12 DefType statements in total offered by Visual Basic 6.0. The default type may be overridden for a specific declaration by using a special suffix character on the variable name (# for Double, ! for Single, & Long, % for Integer, \$ for String, and @ for Currency).

1.10.2 DERIVATIVE LANGUAGES

- Visual Basic itself is derived heavily from BASIC.
- Microsoft has developed derivatives of Visual Basic for use in scripting.
- Visual Basic subsequently has been replaced with a .NET platform version.

❖ Visual Basic for Applications (VBA)

It is a structure developed to create automation within software such as Microsoft Office.

❖ VB Script

VBScript is the default language for Active Server Pages.

Active Server Pages (ASP) is the standard language of the web Platform.

It is based on the Visual Basic syntax.

ASP and VB Script are not the same thing as ASP.NET.

ASP.NET is in a similar category with Visual Basic.Net.

❖ **Visual Basic .NET**

Although Visual Basic.NET is the heir to the Visual Basic 6.0 Platform, it is a different structure and runs on the .NET Framework.

This Platform is a step to strengthen the Object-Oriented features of the Visual Basic language.

However, it can be said that the user-friendly features of the Visual Basic language that make it an advanced language are compromised.

Apart from these;

- OpenOffice Basic is a Visual Basic compatible interpreter arising from the StarOffice office suite.
- The Gambas is a free software programming language inspired by Visual Basic for the Linux operating system.
It is not a Visual Basic clone, but it can convert Visual Basic programs to the Gambas.
- WinWrap Basic is a third-party VBA variant used with various software that programmers can use to create a macro facility in their programs.
- LotusScript is a VBA variant available in Lotus SmartSuite and Lotus Notes.
- Later versions of the Corel WordPerfect Office apply to access to VBA as one of the macro/scripting languages.

Other major languages are CorelScript and PerfectScript earlier versions of Microsoft Word use a variant of Visual Basic called WordBasic.

REFERENCES

1. https://en.wikipedia.org/wiki/Visual_Basic
2. <https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/>
3. <https://docs.microsoft.com/en-us/dotnet/visual-basic/>
4. <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/concepts/>
5. <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/program-structure/program-structure-and-code-conventions>
6. <https://docs.microsoft.com/en-us/dotnet/visual-basic/programming-guide/language-features/>
7. https://en.wikipedia.org/wiki/Alan_Cooper
8. <https://www.cleverism.com/skills-and-tools/visual-basic/>
9. <http://ceng.eskisehir.edu.tr/emrekacmaz/bil158/VBTutorial.pdf>
10. <https://www.thoughtco.com/what-is-visual-basic-3423998>
11. https://simple.wikipedia.org/wiki/Visual_Basic
12. https://en.wikipedia.org/wiki/Alan_Cooper
13. <https://www.linkedin.com/pulse/visual-basic-importance-programming-language-fareed/>