جامعة المستقبل
AL MUSTAQBAL UNIVERSITY
كلية الهندسة والتقنيات الهندسية

Communication Technical Engineering Department

2nd Stage

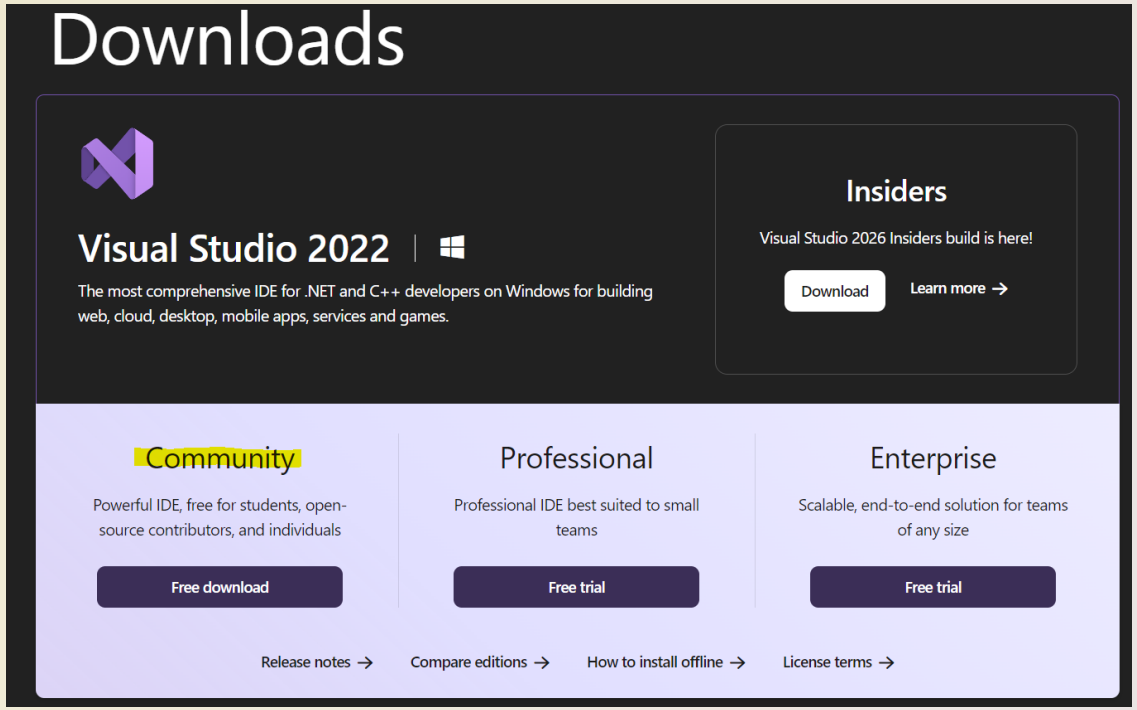
Visual Basic - UOMU0207036

Lecture 2 – Introduction to Visual Basic and its instructions

Dr. Mohammed Fadhil
PhD in Computer Networks
Email: mohammed.fadhil1@uomus.edu.iq

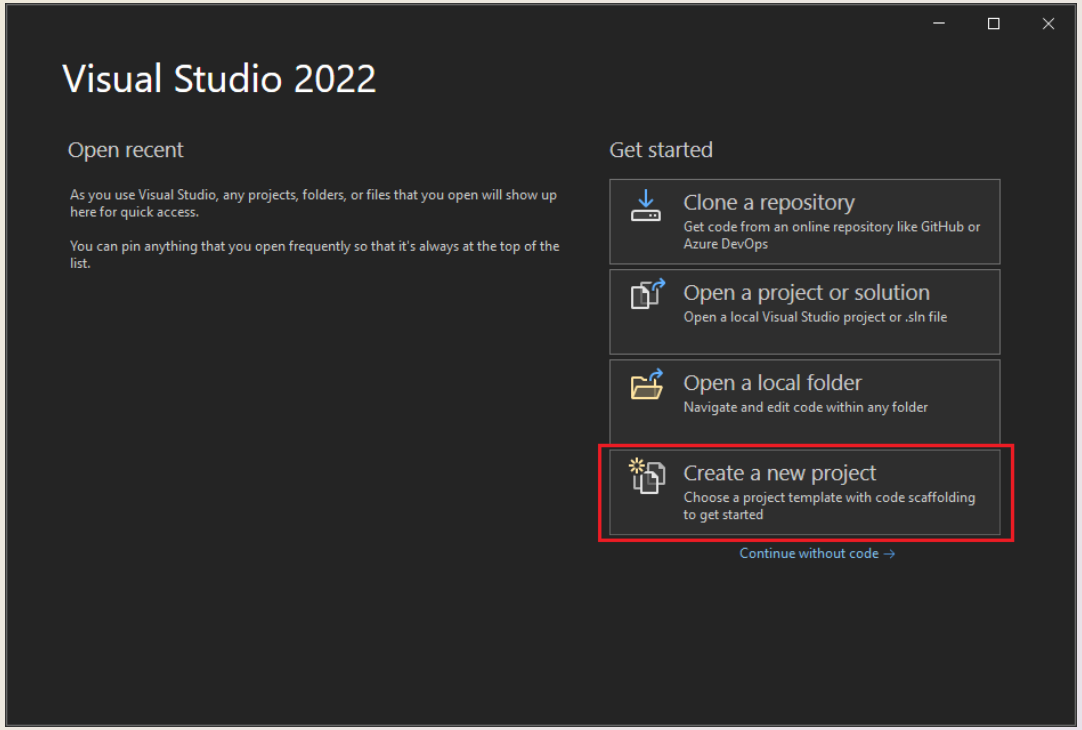
Create a simple Visual Basic console app

- **Prerequisites**
  - If you don't have Visual Studio, go to Visual Studio downloads to install it for free.



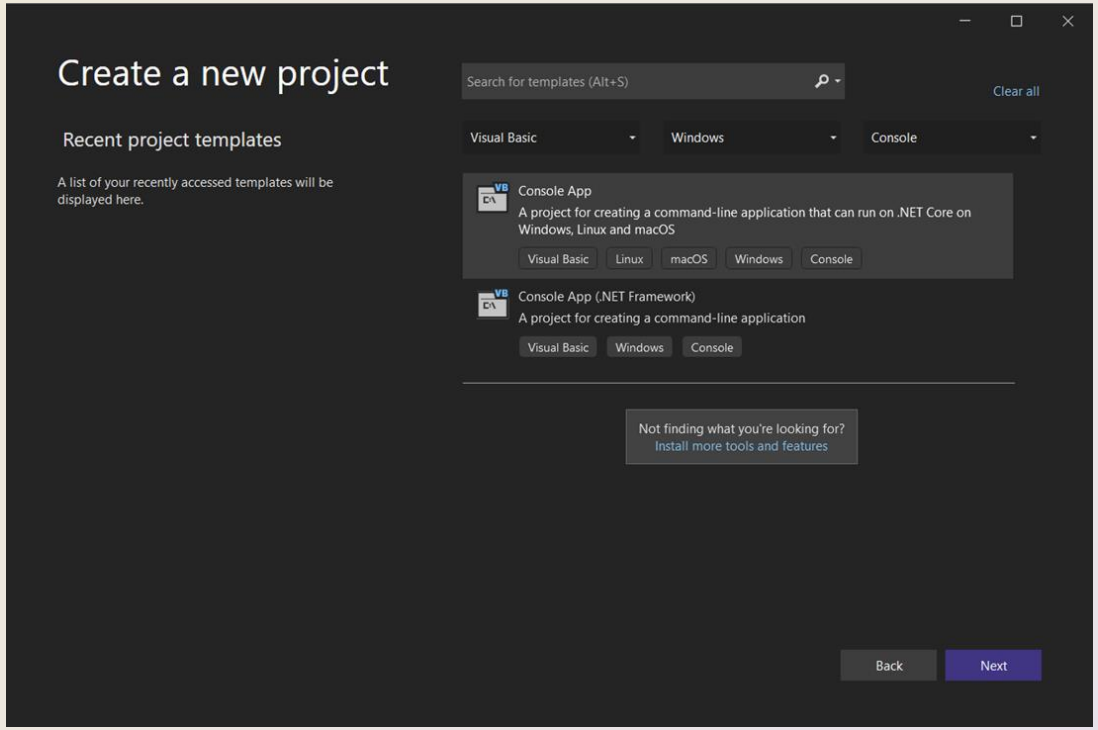
Create a project

- First, you create a Visual Basic app project. The default project template includes all the files you need for a runnable app.
  - Open Visual Studio.
  - On the start window, choose **Create a new project**.



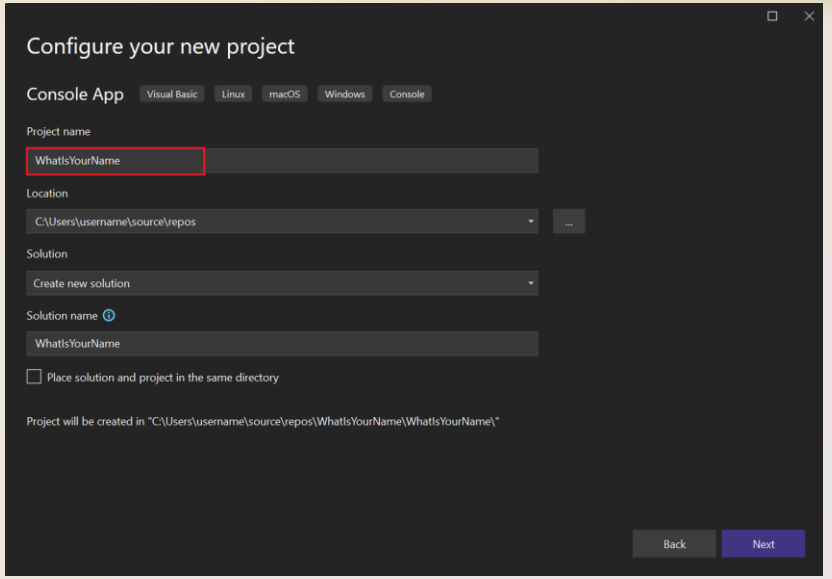
Create a project (cont.)

- In the Create a new project window, choose Visual Basic from the language list. Next, choose Windows from the platform list and Console from the project types list.
- After you apply the language, platform, and project type filters, choose the Console App template, and then choose Next.

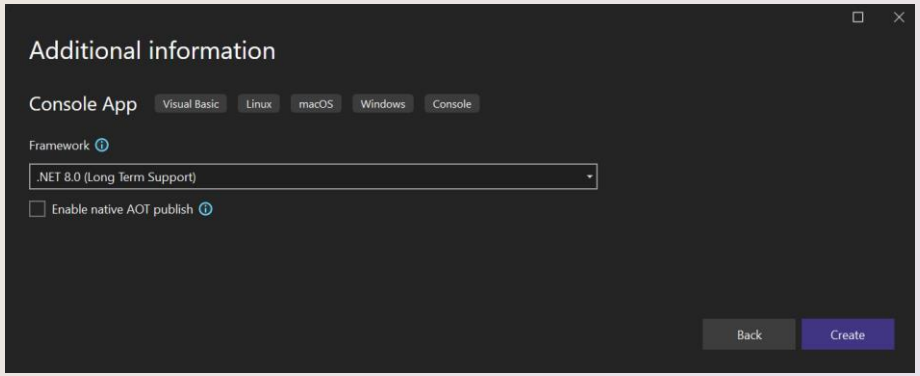


Create a project (cont.)

- In the Configure your new project window, enter WhatIsYourName in the Project name box. Then, choose Next.



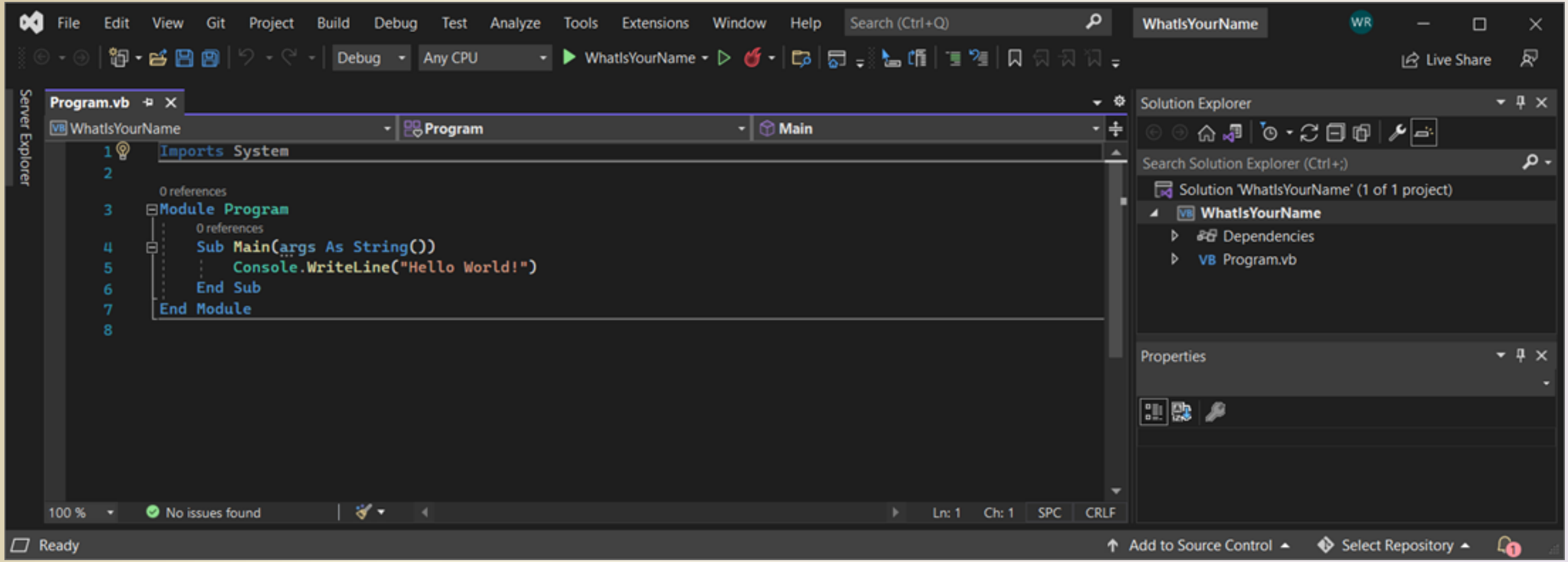
- In the Additional information window, .NET 8.0 should already be selected for your target framework. If not, select .NET 8.0. Then, choose Create.



Run the app

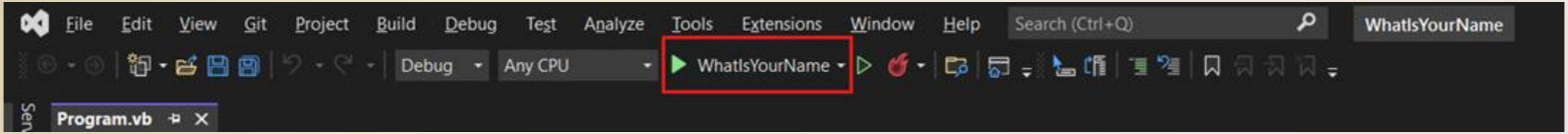
- After you select your Visual Basic project template and name your project, Visual Studio creates a *Program.vb* file. The default code calls the WriteLine method to display the literal string "Hello World!" in the console window.
- There are two ways to run this code, inside Visual Studio in *debug mode*, and from your computer as a regular *standalone* app.

Run the app in debug mode

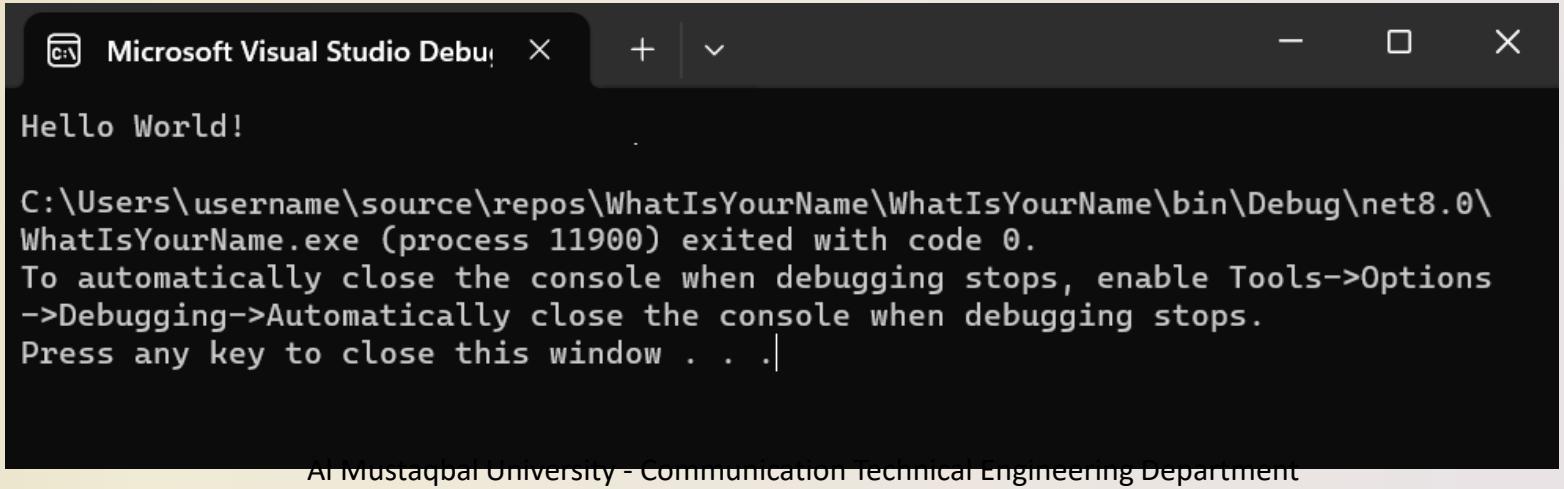


Run the app in debug mode

- Select the **WhatIsYourName** button or press **F5** to run the default code in Debug mode.



- When the app runs in the Microsoft Visual Studio Debug Console, "Hello World!" displays. Press any key to close the debug console window and end the app:



Run the app as a standalone

- To see the output outside of Visual Studio, in a system console window, build and run the executable (.exe file).
  - In the **Build** menu, choose **Build Solution**.
  - In **Solution Explorer**, right-click on **WhatIsYourName** and choose **Open File in File Explorer**.
  - In **File Explorer**, navigate to the *bin\Debug\net8.0* directory and run *WhatIsYourName.exe*.
  - The *Main* procedure terminates after its single statement executes and the console window closes immediately. To keep the console visible until the user presses a key, see the next section.

Add code to ask for user input

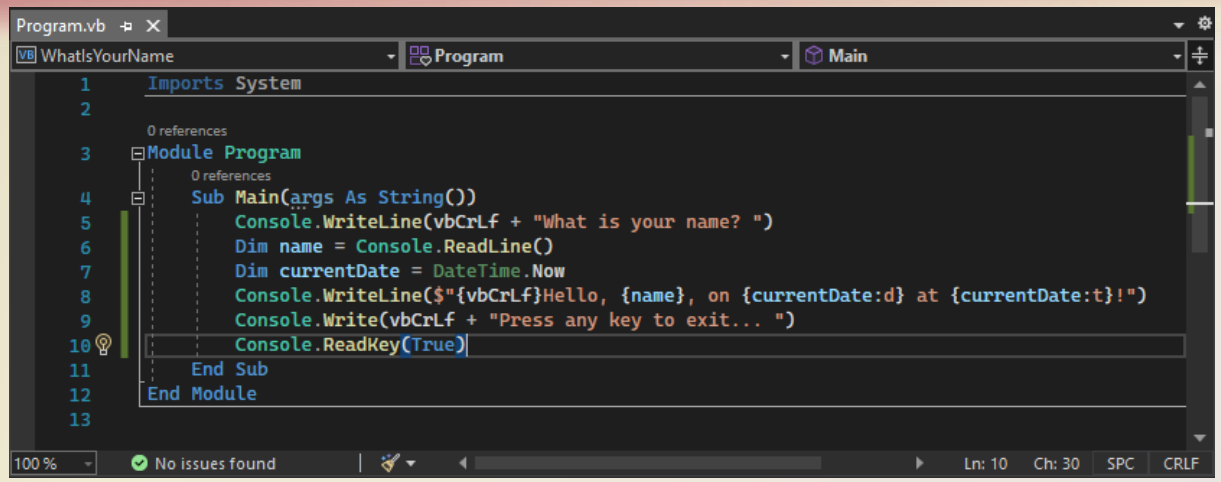
- Next, you add Visual Basic code that prompts you for your name and then displays it along with the current date and time. In addition, you add code that pauses the console window until the user presses a key.
  - Enter the following Visual Basic code after the Sub Main(args As String()) line and before the End Sub line, replacing the **WriteLine** line:

- **Write and WriteLine write a string to the console.**
- **ReadLine reads input from the console, in this case a string.**
- **DateTime represents a datetime, and Now returns the current time.**
- **ReadKey() pauses the app and waits for a keypress.**

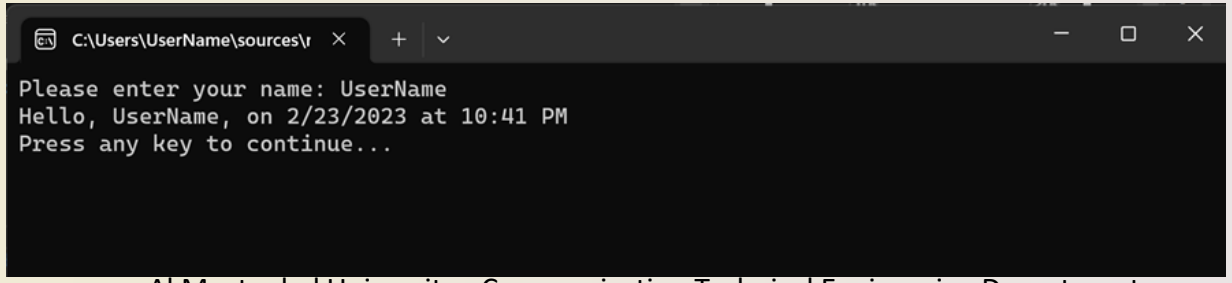
VB Copy

```
Console.Write("Please enter your name: ")
Dim name = Console.ReadLine()
Dim currentDate = DateTime.Now
Console.WriteLine($"Hello, {name}, on {currentDate:d} at {currentDate:t}")
Console.Write("Press any key to continue...")
Console.ReadKey(True)
```

Add code to ask for user input



- Select the **WhatIsYourName** button or press **F5** to build and run your app in Debug mode.
- When the debug console window opens, enter your name. Your console window should look similar to the following screenshot:



Extra credit: Add two numbers

VB Copy

```vb
Module Program
    Public num1 As Integer
    Public num2 As Integer
    Public answer As Integer
    Sub Main(args As String())
        Console.Write("Type a number and press Enter")
        num1 = Console.ReadLine()
        Console.Write("Type another number to add to it and press Enter")
        num2 = Console.ReadLine()
        answer = num1 + num2
        Console.WriteLine("The answer is " & answer)
        Console.Write("Press any key to continue...")
        Console.ReadKey(True)
    End Sub
End Module
```

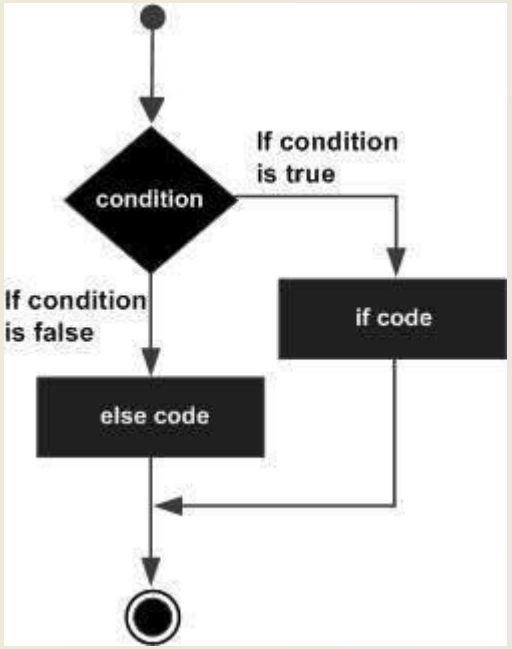
If...Then...Else Statement (Visual Basic)

- An **If** statement can be followed by an optional **Else** statement, which executes when the Boolean expression is false.
- **Syntax**
  - The syntax of an If...Then... Else statement in VB.Net is as follows –

```
If(boolean_expression)Then
   'statement(s) will execute if the Boolean expression is true
Else
  'statement(s) will execute if the Boolean expression is false
End If
```

If...Then...Else Statement (Visual Basic)

- If the Boolean expression evaluates to **true**, then the if block of code will be executed, otherwise else block of code will be executed.



Example : Light Switch Checker (uses Boolean)

- **Scenario:**
  - You’re simulating a light switch.
    If the switch is True → the light is **ON**.
    If it’s False → the light is **OFF**.

```
Module Program
    Sub Main()
        Dim lightOn As Boolean

        Console.Write("Is the light on? (true/false): ")
        lightOn = Console.ReadLine()

        If lightOn = True Then
            Console.WriteLine("The light is ON. The room is bright.")
        Else
            Console.WriteLine("The light is OFF. The room is dark.")
        End If

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

The If...Else If...Else Statement

- An **If** statement can be followed by an optional **Else if...Else** statement, which is very useful to test various conditions using single If...Else If statement.
- When using If... Else If... Else statements, there are few points to keep in mind.
  - An If can have zero or one Else's and it must come after an Else If's.
  - An If can have zero to many Else If's and they must come before the Else.
  - Once an Else if succeeds, none of the remaining Else If's or Else's will be tested.

The If...Else If...Else Statement

- Syntax
- The syntax of an if...else if...else statement in VB.Net is as follows –

```
If(boolean_expression 1)Then
   ' Executes when the boolean expression 1 is true
ElseIf( boolean_expression 2)Then
   ' Executes when the boolean expression 2 is true
ElseIf( boolean_expression 3)Then
   ' Executes when the boolean expression 3 is true
Else
   ' executes when the none of the above condition is true
End If
```

Example: Internet Speed Evaluator

- Scenario:

  You work as a communication engineer checking the internet quality at different locations.

  The program asks for the internet speed (Mbps) and tells the user whether the connection is slow, average, or fast.

```
Module Program
    Sub Main()
        Dim speed As Double

        Console.Write("Enter your internet speed (in Mbps): ")
        speed = Console.ReadLine()

        If speed < 5 Then
            Console.WriteLine("Connection is SLOW. Consider upgrading your plan.")
        ElseIf speed >= 5 And speed <= 20 Then
            Console.WriteLine("Connection is AVERAGE. Suitable for basic use.")
        Else
            Console.WriteLine("Connection is FAST. Great for streaming and online meetings!")
        End If

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

Age Checker Example

- Simple Age Checker
- Goal:
  - Write a program that asks the user for their age and then prints a message depending on the number they enter.

```
Module Program
    Sub Main()
        Console.Write("Enter your age: ")
        Dim age As Integer = Console.ReadLine()

        If age < 13 Then
            Console.WriteLine("You are a child.")
        ElseIf age >= 13 And age <= 19 Then
            Console.WriteLine("You are a teenager.")
        Else
            Console.WriteLine("You are an adult.")
        End If

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

Question:

Question:

Ali is hosting a game show **“Ali’s Big Giveaway!”**. Write a Visual Basic program that asks the user to choose door **1, 2, or 3** and displays:

- 1 → “You win a new car!”
- 2 → “You win a new boat!”
- 3 → “You win a new house!”

- Otherwise → “Bad input!”

```
Module Program
    Sub Main()
        Console.WriteLine("Ali's big giveaway!")
        Console.Write("Would you prefer what is behind door number 1, 2, or 3: ")
        Dim userValue As String = Console.ReadLine()
        Dim Message1 As String = ""

        If userValue = "1" Then
            Message1 = "You win a new car!"
        ElseIf userValue = "2" Then
            Message1 = "You win a new boat!"
        ElseIf userValue = "3" Then
            Message1 = "You win a new house!"
        Else
            Message1 = "Bad input!"
        End If

        Console.WriteLine(Message1)
        Console.ReadLine()
    End Sub
End Module
```

Thank You ☺