Date		Unit	4

Application Development on .NET

C# .NET: Building Windows Applications

Example: Windows Form with Timer, PictureBox, ComboBox, and ProgressBar

```
using System;
using System.Drawing;
using System.Windows.Forms;

public class MainForm : Form
{
    private Timer timer;
    private PictureBox pictureBox;
    private ComboBox comboBox;
    private ProgressBar progressBar;

    public MainForm()
    {
        // Form settings
        this.Text = "C#.NET Windows Application";
        this.Size = new Size(400, 300);

        // Timer
        timer = new Timer();
        timer.Interval = 1000; // 1 second
```

```
        timer.Tick += Timer_Tick;
        timer.Start();

        // PictureBox
        pictureBox = new PictureBox();
        pictureBox.Image = Image.FromFile("example.jpg"); // Replace with your image path
        pictureBox.SizeMode = PictureBoxSizeMode.StretchImage;
        pictureBox.Size = new Size(100, 100);
        pictureBox.Location = new Point(10, 10);
        this.Controls.Add(pictureBox);

        // ComboBox
        comboBox = new ComboBox();
        comboBox.Items.AddRange(new string[] { "Option 1", "Option 2", "Option 3" });
        comboBox.Location = new Point(10, 120);
        this.Controls.Add(comboBox);

        // ProgressBar
        progressBar = new ProgressBar();
        progressBar.Location = new Point(10, 150);
        progressBar.Width = 200;
        progressBar.Value = 50; // Initial progress
        this.Controls.Add(progressBar);
    }

    private void Timer_Tick(object sender, EventArgs e)
    {
        // Increment progress bar
        progressBar.Value = (progressBar.Value + 10) % 101;
    }

    [STAThread]
    public static void Main()
    {
        Application.EnableVisualStyles();
        Application.Run(new MainForm());
    }
}
```

2. VB.NET: Windows Form with Controls

Example: Subroutines, Functions, and Multiple Controls

```
Imports System.Windows.Forms

Public Class MainForm
    Private WithEvents Timer1 As New Timer()
    Private PictureBox1 As New PictureBox()
    Private ComboBox1 As New ComboBox()
    Private ProgressBar1 As New ProgressBar()
    Private Button1 As New Button()

    Public Sub New()
        ' Form settings
        Me.Text = "VB.NET Windows Application"
        Me.Size = New Drawing.Size(400, 300)

        ' Timer
        Timer1.Interval = 1000
        Timer1.Start()

        ' PictureBox
        PictureBox1.Image = Image.FromFile("example.jpg") ' Replace with your image path
        PictureBox1.SizeMode = PictureBoxSizeMode.StretchImage
        PictureBox1.Size = New Drawing.Size(100, 100)
        PictureBox1.Location = New Drawing.Point(10, 10)
        Me.Controls.Add(PictureBox1)

        ' ComboBox
        ComboBox1.Items.AddRange(New String() {"Option 1", "Option 2", "Option 3"})
        ComboBox1.Location = New Drawing.Point(10, 120)
        Me.Controls.Add(ComboBox1)

        ' ProgressBar
        ProgressBar1.Location = New Drawing.Point(10, 150)
        ProgressBar1.Width = 200
        ProgressBar1.Value = 50 ' Initial progress
        Me.Controls.Add(ProgressBar1)

        ' Button to call a function
        Button1.Text = "Click Me"
        Button1.Location = New Drawing.Point(10, 180)
        AddHandler Button1.Click, AddressOf Button1_Click
```

```
    Me.Controls.Add(Button1)
  End Sub

  Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    ' Increment progress bar
    ProgressBar1.Value = (ProgressBar1.Value + 10) Mod 101
  End Sub

  Private Sub Button1_Click(sender As Object, e As EventArgs)
    ' Call a function
    MessageBox.Show("Result from function: " & CalculateSum(5, 10))
  End Sub

  Private Function CalculateSum(a As Integer, b As Integer) As Integer
    ' A simple function
    Return a + b
  End Function

  <STAThread>
  Public Shared Sub Main()
    Application.EnableVisualStyles()
    Application.Run(New MainForm())
  End Sub
End Class
```

C# and **VB.NET** Windows Forms applications.
Usage of:

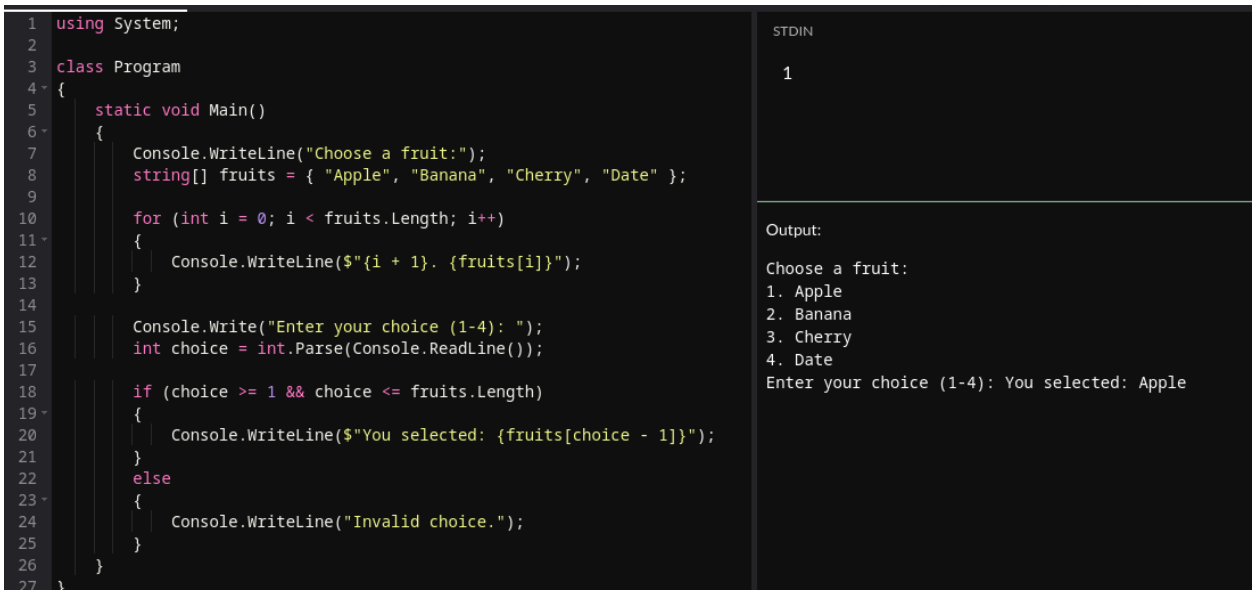
- **Timer**: Demonstrates periodic updates (e.g., updating a progress bar).
- **PictureBox**: Displays an image.
- **ComboBox**: Drop-down menu for user selection.
- **ProgressBar**: Visualizes progress.

Subroutines and Functions: Shows function usage for dynamic behaviors.

Examples to test on Online compiler

1. Using Controls Like Combo-box and Group-box

Since console applications don't support GUI controls directly, we can mimic their behavior with console-based menus.



```
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("Choose a fruit:");
        string[] fruits = { "Apple", "Banana", "Cherry", "Date" };

        for (int i = 0; i < fruits.Length; i++)
        {
            Console.WriteLine($"{i + 1}. {fruits[i]}");
        }

        Console.Write("Enter your choice (1-4): ");
        int choice = int.Parse(Console.ReadLine());

        if (choice >= 1 && choice <= fruits.Length)
        {
            Console.WriteLine($"You selected: {fruits[choice - 1]}");
        }
        else
        {
            Console.WriteLine("Invalid choice.");
        }
    }
}
```

STDIN

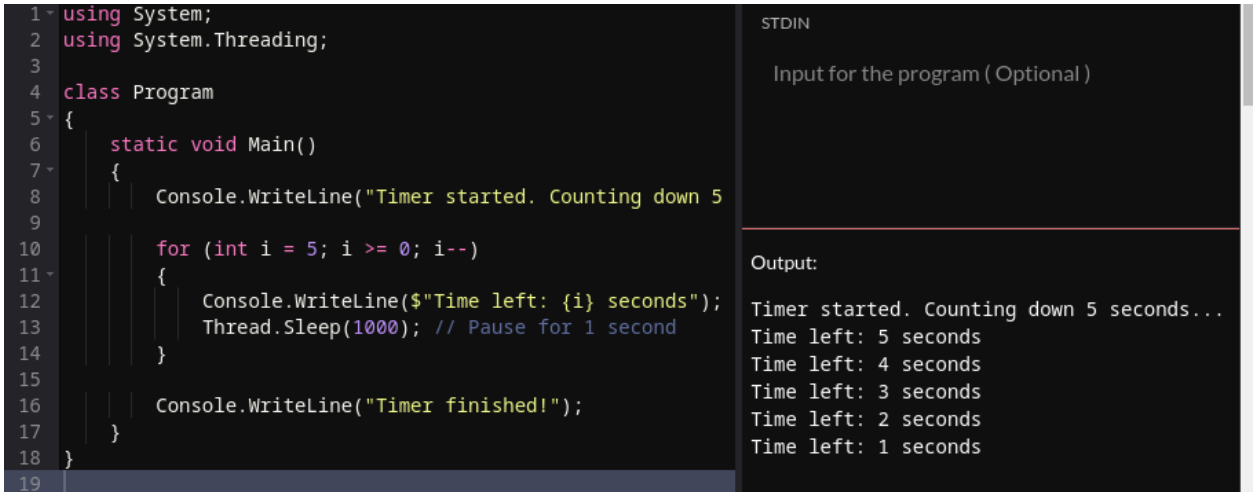
```
1
```

Output:

```
Choose a fruit:
1. Apple
2. Banana
3. Cherry
4. Date
Enter your choice (1-4): You selected: Apple
```

2. Using Timer

In console applications, a timer can be simulated using a loop and `Thread.Sleep`.



3. Track-bar Simulation

A track-bar can be simulated with input values between a specified range.

```
using System;

class Program
{
   static void Main()
   {
      int min = 0, max = 10;

      Console.WriteLine($"Move the track-bar (Enter a value between {min} and {max}): ");
      int value = int.Parse(Console.ReadLine());

      if (value >= min && value <= max)
      {
         Console.WriteLine($"Track-bar value: {value}");
      }
      else
      {
         Console.WriteLine("Invalid value.");
      }
   }
}
```

STDIN
4

Output:

Move the track-bar (Enter a value between 0 and 10):
Track-bar value: 4

5. Subroutines and Functions in VB.NET (C# Equivalent)

```
using System;

class Program
{
    static void Main()
    {
        PrintMessage();
    }

    static void PrintMessage()
    {
        Console.WriteLine("This is a subroutine example.");
    }
}
```

STDIN

4

Output:

This is a subroutine example.

Function example

```
using System;

class Program
{
    static void Main()
    {
        int a = 10, b = 20;
        int sum = Add(a, b);
        Console.WriteLine($"The sum of {a} and {b} is {sum
    }

    static int Add(int x, int y)
    {
        return x + y;
    }
}
```

STDIN

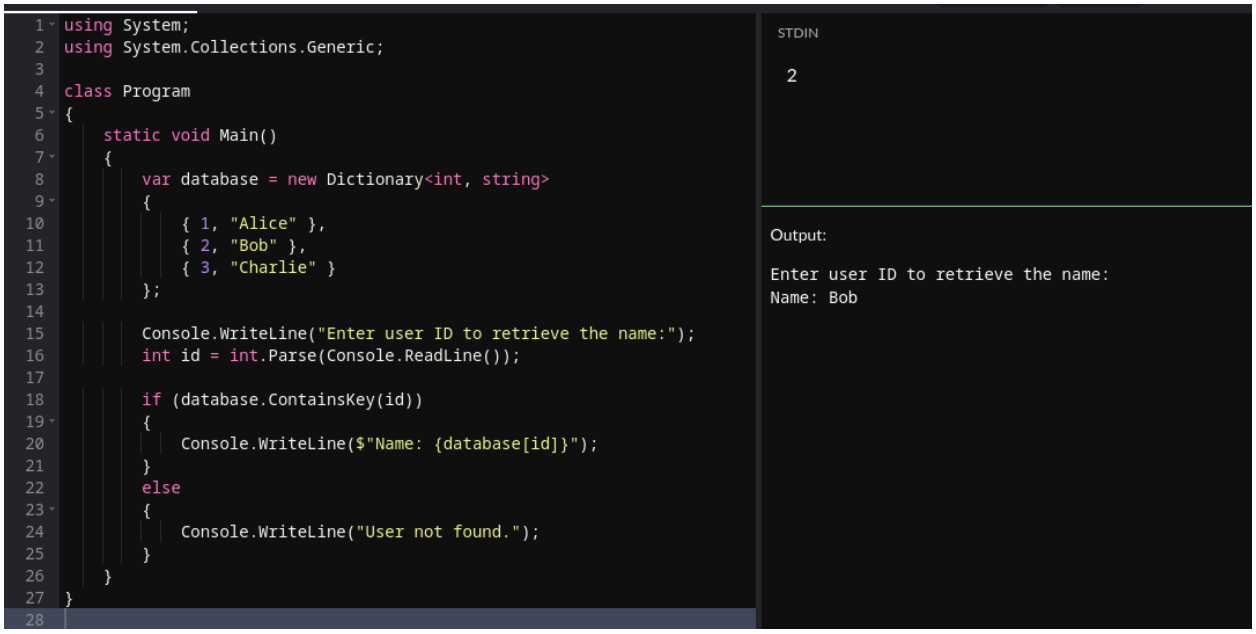
4

Output:

The sum of 10 and 20 is 30.

6. Database Applications

Since online compilers typically don't support database connections, we can simulate a database using a dictionary.



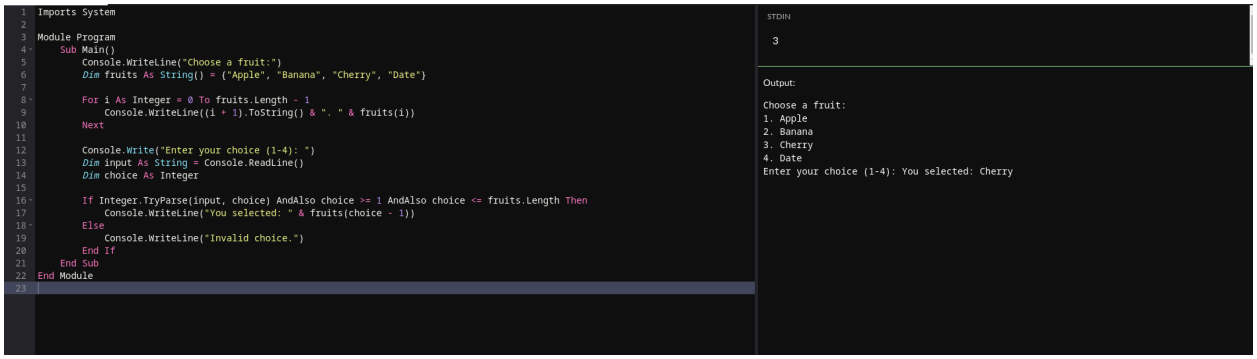
```
using System;
using System.Collections.Generic;

class Program
{
    static void Main()
    {
        var database = new Dictionary<int, string>
        {
            { 1, "Alice" },
            { 2, "Bob" },
            { 3, "Charlie" }
        };

        Console.WriteLine("Enter user ID to retrieve the name:");
        int id = int.Parse(Console.ReadLine());

        if (database.ContainsKey(id))
        {
            Console.WriteLine($"Name: {database[id]}");
        }
        else
        {
            Console.WriteLine("User not found.");
        }
    }
}
```

```
STDIN
2

Output:

Enter user ID to retrieve the name:
Name: Bob
```

Note - For full GUI implementations with controls like `PictureBox` and `GroupBox`, you would need to use Windows Forms or WPF, which aren't compatible with online compilers.

VB.NET

Combo-box Simulation



```
Imports System

Module Program
    Sub Main()
        Console.WriteLine("Choose a fruit:")
        Dim fruits As String() = {"Apple", "Banana", "Cherry", "Date"}

        For i As Integer = 0 To fruits.Length - 1
            Console.WriteLine((i + 1).ToString() & ". " & fruits(i))
        Next

        Console.Write("Enter your choice (1-4): ")
        Dim input As String = Console.ReadLine()
        Dim choice As Integer

        If Integer.TryParse(input, choice) AndAlso choice >= 1 AndAlso choice <= fruits.Length Then
            Console.WriteLine("You selected: " & fruits(choice - 1))
        Else
            Console.WriteLine("Invalid choice.")
        End If
    End Sub
End Module
```

```
STDIN
 3

Output:

Choose a fruit:
1. Apple
2. Banana
3. Cherry
4. Date
Enter your choice (1-4): You selected: Cherry
```

Timer example

```vb
Imports System
Imports System.Threading

Module Program
    Sub Main()
        Console.WriteLine("Timer started. Counting down 5 seconds...")

        For i As Integer = 5 To 0 Step -1
            Console.WriteLine("Time left: " & i & " seconds")
            Thread.Sleep(1000)
        Next

        Console.WriteLine("Timer finished!")
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
Timer started. Counting down 5 seconds...
Time left: 5 seconds
Time left: 4 seconds
Time left: 3 seconds
Time left: 2 seconds
Time left: 1 seconds
Time left: 0 seconds
Timer finished!
```

Track-bar Simulation

```vb
Imports System

Module Program
    Sub Main()
        Dim min As Integer = 0
        Dim max As Integer = 10

        Console.WriteLine("Move the track-bar (Enter a value between " & min & " and " & max & "): ")
        Dim input As String = Console.ReadLine()
        Dim value As Integer

        If Integer.TryParse(input, value) AndAlso value >= min AndAlso value <= max Then
            Console.WriteLine("Track-bar value: " & value)
        Else
            Console.WriteLine("Invalid value.")
        End If
    End Sub
End Module
```

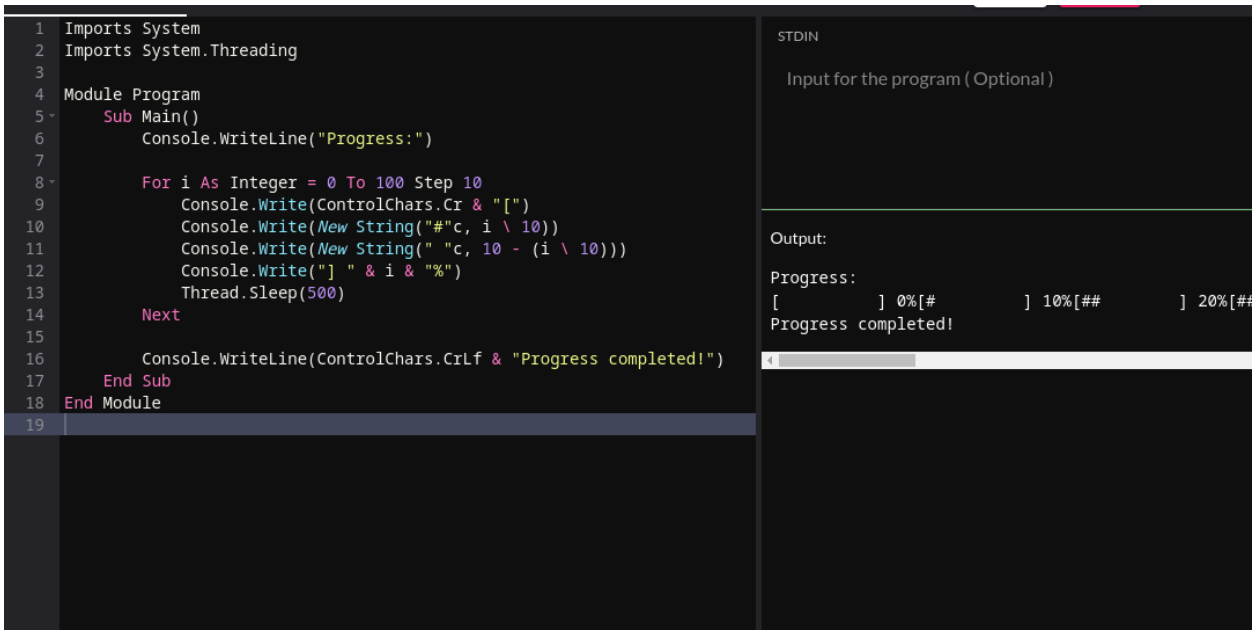
STDIN

3

Output:

```
Move the track-bar (Enter a value between 0 and 10):
Track-bar value: 3
```

Progress-bar Simulation



```
Imports System
Imports System.Threading

Module Program
    Sub Main()
        Console.WriteLine("Progress:")

        For i As Integer = 0 To 100 Step 10
            Console.Write(ControlChars.Cr & "[")
            Console.Write(New String("#"c, i \ 10))
            Console.Write(New String(" "c, 10 - (i \ 10)))
            Console.Write("] " & i & "%")
            Thread.Sleep(500)
        Next

        Console.WriteLine(ControlChars.CrLf & "Progress completed!")
    End Sub
End Module
```

```
STDIN

Input for the program ( Optional )

Output:

Progress:
[          ] 0%[#         ] 10%[##        ] 20%[##
Progress completed!
```

Output:

Progress:

[ ] 0%
[#] 10%
[##] 20%
[###] 30%
[####] 40%
[#####] 50%
[######] 60%
[#######] 70%
[########] 80%
[#########] 90%
[##########] 100%
Progress completed!

Subroutines and Functions

Subroutine

```vb
Imports System

Module Program
    Sub Main()
        PrintMessage()
    End Sub

    Sub PrintMessage()
        Console.WriteLine("This is a subroutine example.")
    End Sub
End Module
```

STDIN

Input for the program (Optional)

Output:

```
This is a subroutine example.
```

Functions

```vb
Imports System

Module Program
    Sub Main()
        Dim a As Integer = 10
        Dim b As Integer = 20
        Dim sum As Integer = Add(a, b)
        Console.WriteLine("The sum of " & a & " and " & b & " is " & sum & ".")
    End Sub

    Function Add(x As Integer, y As Integer) As Integer
        Return x + y
    End Function
End Module
```

STDIN

Input for the program (Optional)

Output:

```
The sum of 10 and 20 is 30.
```

Database Simulation

```vb
Imports System
Imports System.Collections

Module Program
    Sub Main()
        ' Initialize dictionary using older syntax
        Dim database As New Hashtable()
        database(1) = "Alice"
        database(2) = "Bob"
        database(3) = "Charlie"

        ' Prompt user for input
        Console.WriteLine("Enter user ID to retrieve the name:")
        Dim input As String = Console.ReadLine()
        Dim id As Integer

        ' Validate input and retrieve name
        If Integer.TryParse(input, id) AndAlso database.ContainsKey(id) Then
            Console.WriteLine("Name: " & database(id).ToString())
        Else
            Console.WriteLine("User not found.")
        End If
    End Sub
End Module
```

STDIN

3

Output:

```
Enter user ID to retrieve the name:
Name: Charlie
```

`Hashtable`: `Hashtable` is an older collection type that is more widely supported in older VB.NET environments.
Initialization: Used explicit key-value assignments instead of inline initialization.
Accessing Values: Used `.ToString()` for `database(id)` to ensure compatibility

END