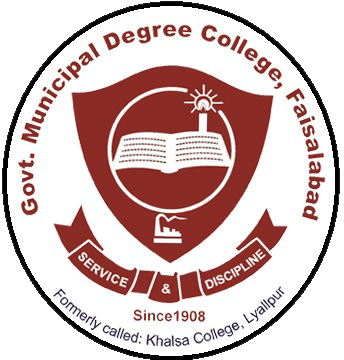
****

**NAME: MUNEEB UR REHMAN**

**ROLL # 120-023**

**COURSE: ADVANCED PROGRAMMING**

**SUBMITTED TO: MAM SANA TARIQ**

**TOPIC: CONTROLS, DLLs, THREADS**

**GOVERNMENT MUNICIPAL GRADUATE COLLEGE**

**TASK # 01:**

**Create a Windows Form application that:**

* **Includes a button to open a file dialog.**
* **Allows the user to select a file and displays the file path in a label.**

**Form.cs:**

using System;

using System.Windows.Forms;

namespace FileDialogApp

{

public class MainForm : Form

{

private Button btnOpenFile;

private Label lblFilePath;

public MainForm()

{

btnOpenFile = new Button { Text = "Select File", Location = new System.Drawing.Point(20, 20) };

lblFilePath = new Label { Location = new System.Drawing.Point(20, 60), AutoSize = true };

btnOpenFile.Click += BtnOpenFile\_Click;

Controls.Add(btnOpenFile);

Controls.Add(lblFilePath);

Text = "File Dialog Example";

Size = new System.Drawing.Size(400, 150);

}

private void BtnOpenFile\_Click(object sender, EventArgs e)

{

using (OpenFileDialog openFileDialog = new OpenFileDialog())

{

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

lblFilePath.Text = "Selected File: " + openFileDialog.FileName;

}

}

}

}

}

**Program.cs:**

using System;

using System.Windows.Forms;

namespace FileDialogApp

{

internal static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

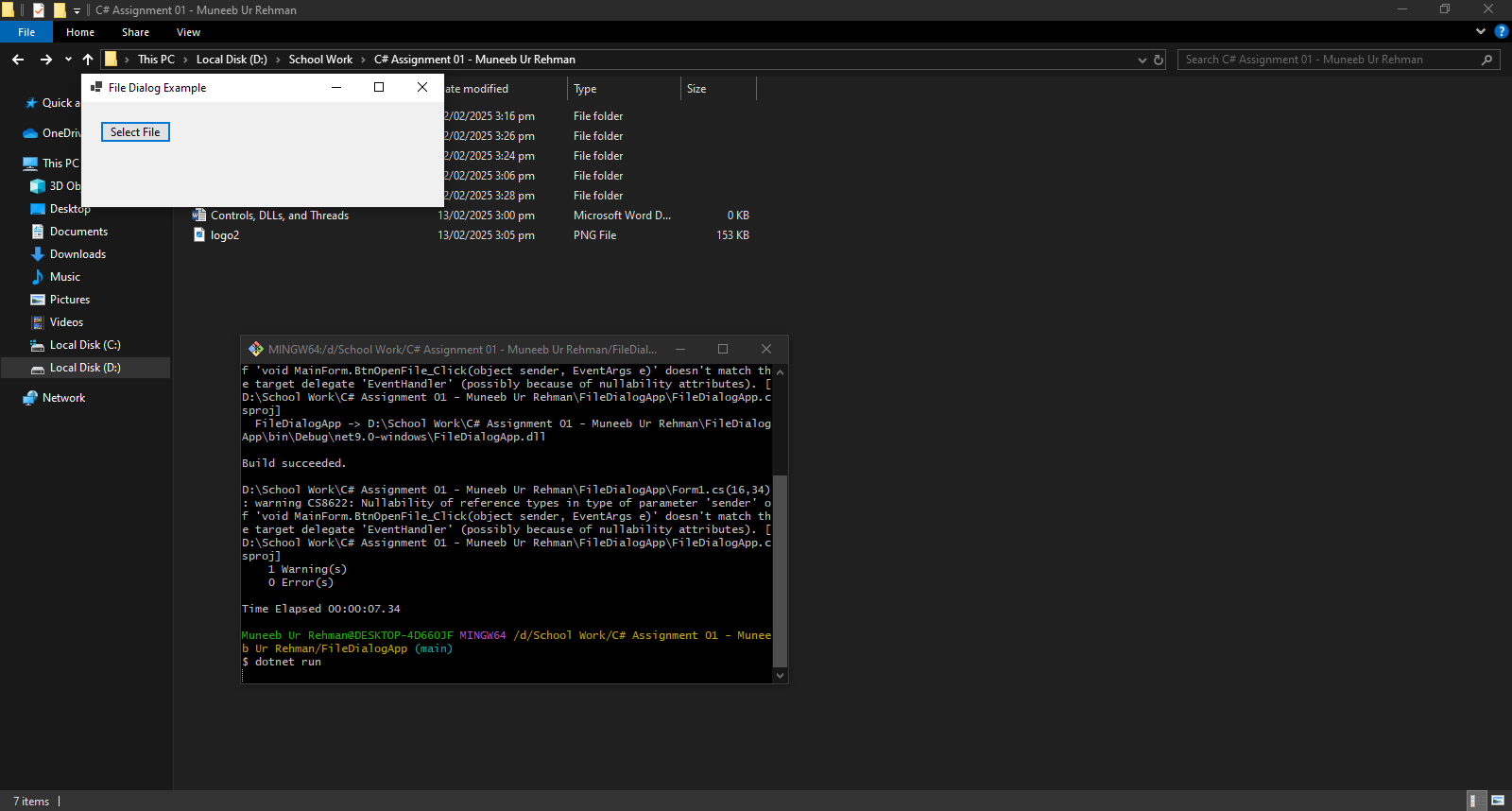
Application.Run(new MainForm()); // Ensure this matches your form name

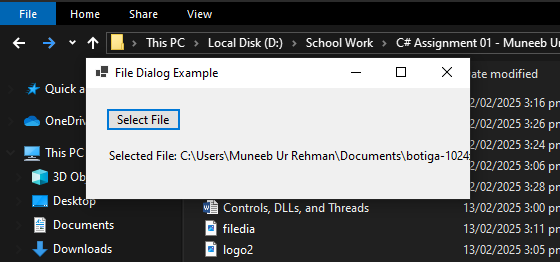
}

}

}

**Application running:**

****

****

**TASK # 02:**

**Develop a Windows Form application that includes:**

* **A ComboBox to select a color.**
* **A ProgressBar that updates dynamically when a button is clicked.**

**Form.cs:**

using System;

using System.Drawing;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ColorProgressApp

{

public class MainForm : Form

{

private ComboBox colorComboBox;

private ProgressBar progressBar;

private Button startButton;

public MainForm()

{

// Set Form Properties

Text = "Color & ProgressBar Demo";

Size = new Size(400, 200);

StartPosition = FormStartPosition.CenterScreen;

// Create ComboBox

colorComboBox = new ComboBox

{

Location = new Point(20, 20),

Width = 150,

};

colorComboBox.Items.AddRange(new string[] { "Red", "Green", "Blue", "Yellow" });

colorComboBox.SelectedIndexChanged += ColorComboBox\_SelectedIndexChanged;

// Create ProgressBar

progressBar = new ProgressBar

{

Location = new Point(20, 60),

Width = 300,

Height = 20,

Minimum = 0,

Maximum = 100,

Value = 0

};

// Create Start Button

startButton = new Button

{

Text = "Start Progress",

Location = new Point(20, 100),

Width = 150

};

startButton.Click += StartButton\_Click;

// Add Controls to Form

Controls.Add(colorComboBox);

Controls.Add(progressBar);

Controls.Add(startButton);

}

private async void StartButton\_Click(object sender, EventArgs e)

{

// Simulate Progress

progressBar.Value = 0;

for (int i = 0; i <= 100; i += 10)

{

await Task.Delay(200); // Simulate work

progressBar.Value = i;

}

}

private void ColorComboBox\_SelectedIndexChanged(object sender, EventArgs e)

{

string selectedColor = colorComboBox.SelectedItem.ToString();

switch (selectedColor)

{

case "Red":

BackColor = Color.Red;

break;

case "Green":

BackColor = Color.Green;

break;

case "Blue":

BackColor = Color.Blue;

break;

case "Yellow":

BackColor = Color.Yellow;

break;

}

}

}

}

**Program.cs:**

using System;

using System.Windows.Forms;

namespace ColorProgressApp

{

static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

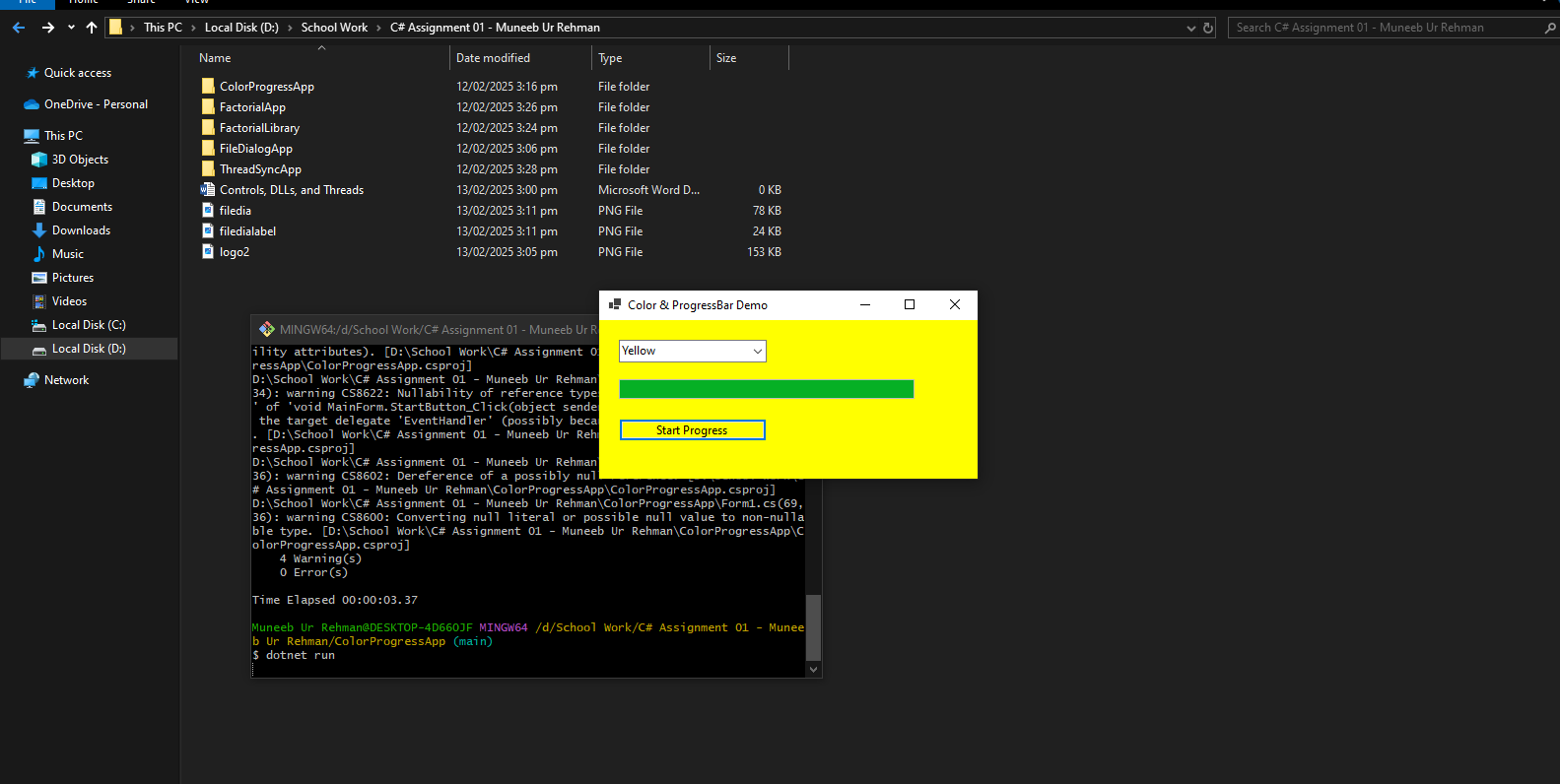
Application.Run(new MainForm()); // Ensure this matches the class name

}

}

}

**Application running:**

****

**TASK # 03:**

**Write a C# program that::**

* **Creates a DLL containing a method to calculate the factorial of a number**
* **Imports and uses this DLL in a separate C# application.**

**Sol:**

**Factorial.cs:**

using System;

namespace FactorialLibrary

{

public class FactorialCalculator

{

public static long CalculateFactorial(int number)

{

if (number < 0)

throw new ArgumentException("Number must be non-negative.");

long result = 1;

for (int i = 1; i <= number; i++)

{

result \*= i;

}

return result;

}

}

}

**Program.cs in Factorial App:**

using System;

using FactorialLibrary;

namespace FactorialApp

{

class Program

{

static void Main()

{

Console.Write("Enter a number: ");

int num = Convert.ToInt32(Console.ReadLine());

long factorial = FactorialCalculator.CalculateFactorial(num);

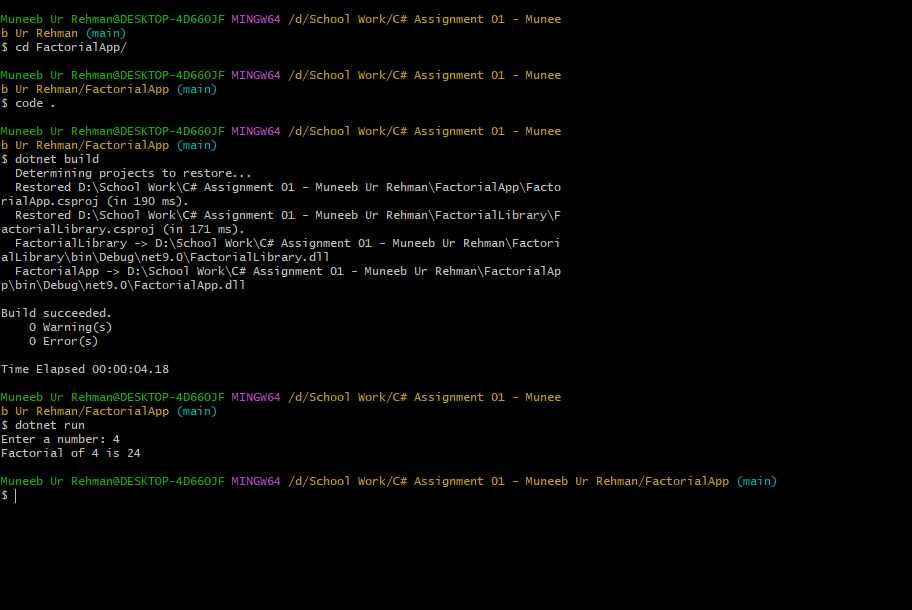
Console.WriteLine($"Factorial of {num} is {factorial}");

}

}

}

**Application running:**

****

**TASK # 04:**

**Write a C# application that:**

* **Creates two threads: one to display even numbers and another to display odd numbers up to 20.**
* **Uses synchronization to ensure the output does not overlap or conflict.**

**Sol:**

**Program.cs:**

using System;

using System.Threading;

namespace ThreadSyncApp

{

class Program

{

private static readonly object lockObject = new object();

private static int number = 1; // Shared counter

static void Main()

{

Thread evenThread = new Thread(PrintEvenNumbers);

Thread oddThread = new Thread(PrintOddNumbers);

evenThread.Start();

oddThread.Start();

evenThread.Join(); // Wait for evenThread to finish

oddThread.Join(); // Wait for oddThread to finish

Console.WriteLine("Both threads have completed execution.");

}

static void PrintEvenNumbers()

{

while (number <= 20)

{

lock (lockObject)

{

if (number % 2 == 0)

{

Console.WriteLine($"Even: {number}");

number++;

}

}

Thread.Sleep(100); // Simulate work

}

}

static void PrintOddNumbers()

{

while (number <= 20)

{

lock (lockObject)

{

if (number % 2 != 0)

{

Console.WriteLine($"Odd: {number}");

number++;

}

}

Thread.Sleep(100); // Simulate work

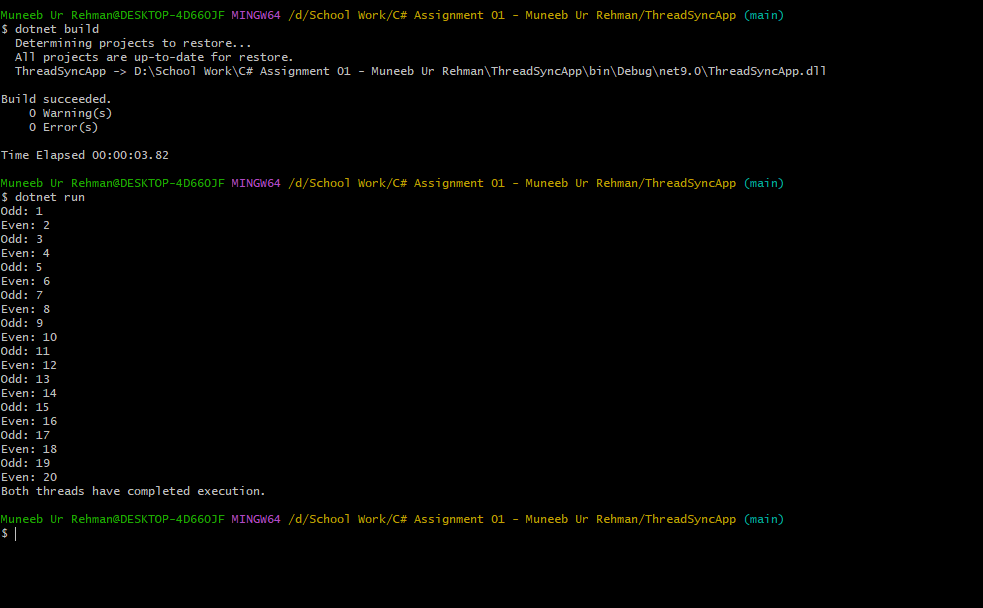
}

}

}

}

**Application Running:**

****