916217-Gannarapu Pavan Kumar

**Stage-3 Handson-1** **C# Additional Topics Async Programming, Multithreading**

**Async Await usage - 1**

Two methods that return Asynchronous task

· The first method invokes the second method and awaits till the second method returns simple string. The string data returned by the second method should be stored in a string in the first method to display it

· Use Thread.Sleep in the second method to simulate the time delay

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

namespace threading\_handson1

{

class Program

{

public static async Task Method1()

{

await Task.Run(() => { string simplestring = Method2(); Console.WriteLine(simplestring); });

}

public static string Method2()

{

Console.WriteLine("We are in second method ");

Thread.Sleep(2000);

return "this is the second method printed by first method ";

}

static void Main(string[] args)

{

Method1();

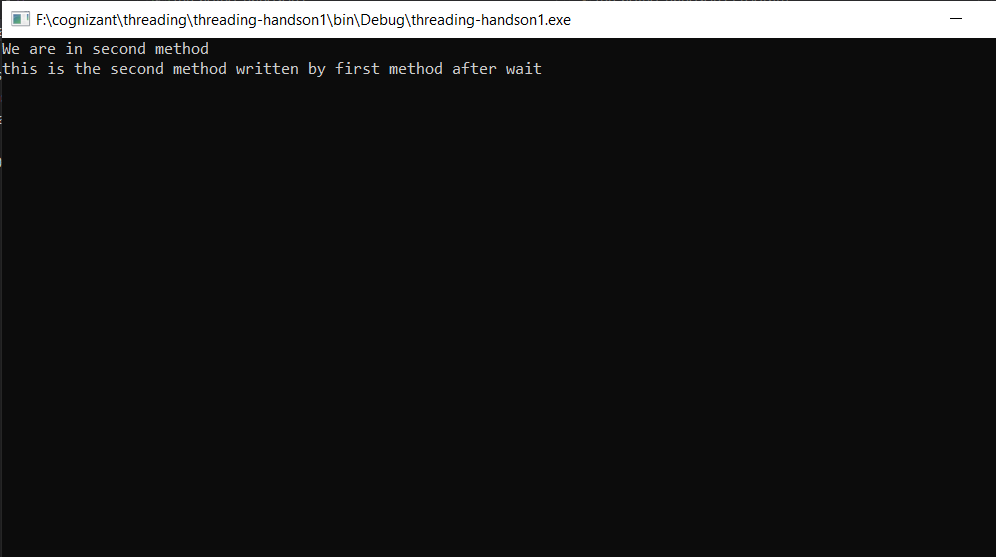
Console.Read();

}

}

}

**Output:**



**Async Await usage – 2**

Use Windows forms application with Async Await usage

Create a Windows Forms application to read the number of characters in a text file(preferably a large file) and print that on a label

Use Async await concept to read the file content as the read operation shouldn't make the Windows form non-responding.

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace threading\_Async\_Await\_usage\_\_\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

public int countlines()

{

int count = 0;

using (StreamReader streamReader = new StreamReader("F:\\cognizant\\threading\\sample.txt"))

{

string content = streamReader.ReadToEnd();

count = content.Length;

Thread.Sleep(500);

}

return count;

}

private async void button1\_Click(object sender, EventArgs e)

{

Task<int> task = new Task<int>(countlines);

task.Start();

textbox1.Text = "File is Loading....";

int count = await task;

textbox1.Text = count.ToString() + " characters";

}

}

}

**Form1.Designer.cs**

namespace threading\_Async\_Await\_usage\_\_\_2

{

partial class Form1

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.button1 = new System.Windows.Forms.Button();

this.textbox1 = new System.Windows.Forms.TextBox();

this.SuspendLayout();

//

// button1

//

this.button1.Location = new System.Drawing.Point(343, 268);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(94, 29);

this.button1.TabIndex = 0;

this.button1.Text = "Submit";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.button1\_Click);

//

// textbox1

//

this.textbox1.Location = new System.Drawing.Point(326, 128);

this.textbox1.Name = "textbox1";

this.textbox1.Size = new System.Drawing.Size(214, 27);

this.textbox1.TabIndex = 1;

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 20F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(800, 450);

this.Controls.Add(this.textbox1);

this.Controls.Add(this.button1);

this.Name = "Form1";

this.Text = "Form1";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Button button1;

private System.Windows.Forms.TextBox textbox1;

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace threading\_Async\_Await\_usage\_\_\_2

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.SetHighDpiMode(HighDpiMode.SystemAware);

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

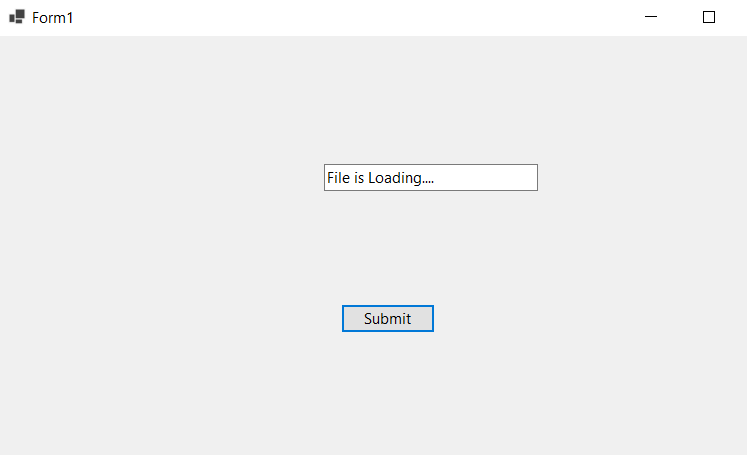
Application.Run(new Form1());

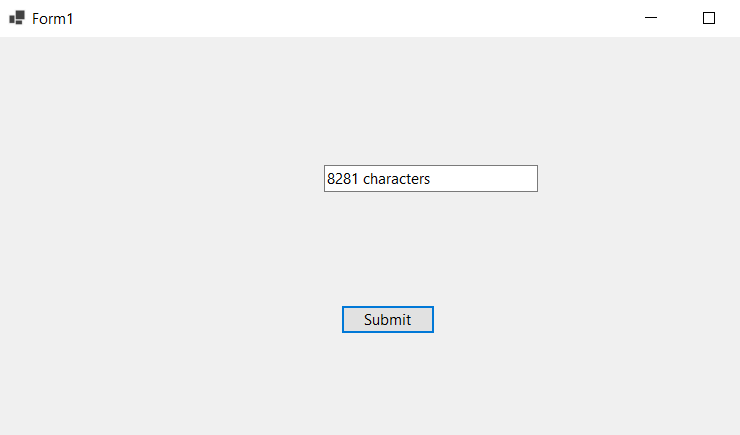
}

}

}

**Output**





**Named parameters – Order of arguments as per the function and modify**

**Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace named\_arguments

{

class Program

{

public static void GetCohortDetails(string Cohortname,int GenCcount,string Mode,string Track,string CurrentModule)

{

Console.WriteLine("It is {0} with {1} GenCs undergoing training for {2} thru {3}. The current module of training is {4}", Cohortname, GenCcount, Track, Mode, CurrentModule);

}

public static void OrderDetails(string Productname, string Sellername, int Orderquantity = 1, bool returnable = true)

{

Console.WriteLine("Here is the order detail – {0} number of {1} by {2} is ordered. It’s returnable status is {3}", Orderquantity, Productname, Sellername, returnable);

}

static void Main(string[] args)

{

Console.WriteLine("GetCohortDetails");

GetCohortDetails(Cohortname: "INTCDE21008", GenCcount: 18, Track: ".NET", Mode: "OBL", CurrentModule: "Stage-3");

Console.WriteLine("OrderDetails");

OrderDetails(Sellername: "sunil", Productname: "apples", Orderquantity: 10, returnable: false);

Console.Read();

}

}

}

**Output:**

