**Exercise 1: Implementing the Singleton Pattern**

using System;

public class Singleton

{

    // Private static variable to hold the single instance of the class

    private static Singleton \_instance;

    // Private constructor to prevent instantiation from outside

    private Singleton()

    {

        // Initialization code here

    }

    // Public static method to provide access to the instance

    public static Singleton Instance

    {

        get

        {

            // Lazy initialization

            if (\_instance == null)

            {

                \_instance = new Singleton();

            }

            return \_instance;

        }

    }

    // Example method to demonstrate functionality

    public void ShowMessage()

    {

        Console.WriteLine("Hello from the Singleton instance!");

    }

}

class Program

{

    static void Main(string[] args)

    {

        // Access the Singleton instance and call a method

        Singleton singletonInstance = Singleton.Instance;

        singletonInstance.ShowMessage();

        // Verify that the same instance is returned

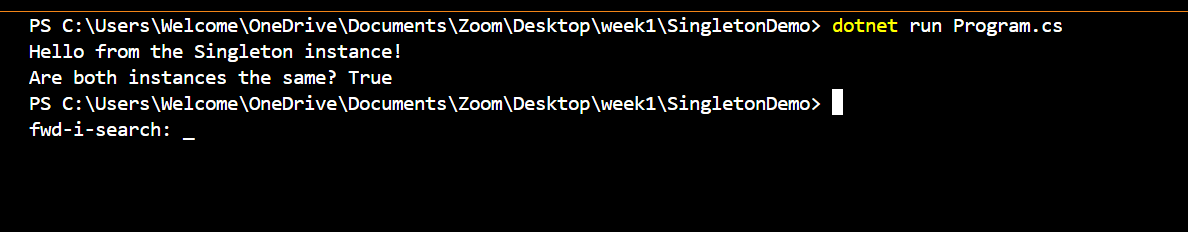
        Singleton anotherInstance = Singleton.Instance;

        Console.WriteLine($"Are both instances the same? {singletonInstance == anotherInstance}");

    }

}

**Output:**

****