Day 4

date 07-08-25

VIJAY M

1.create a two threads to read two separate text files

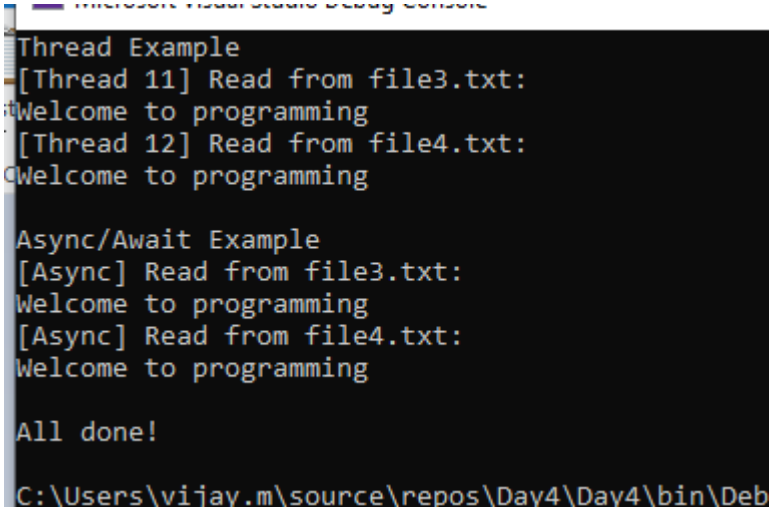
same two filse, try to read using the task async await

```
using System;
using System.IO;
using System.Threading;
using System.Threading.Tasks;

class Program
{
    static void Main(string[] args)
    {
        string filePath1 = "file3.txt";
        string filePath2 = "file4.txt";

        Console.WriteLine("Thread Example");

        Thread t1 = new Thread(() => ReadFileWithThread(filePath1));
        Thread t2 = new Thread(() => ReadFileWithThread(filePath2));

        t1.Start();
        t2.Start();

        t1.Join();
```

```
        t2.Join();

        Console.WriteLine("\nAsync/Await Example");

        Task task1 = ReadFileAsync(filePath1);
        Task task2 = ReadFileAsync(filePath2);

        Task.WaitAll(task1, task2);

        Console.WriteLine("\nAll done!");
    }

    static void ReadFileWithThread(string path)
    {
        try
        {
            if (!File.Exists(path))
            {
                File.Create(path).Dispose();
                Console.WriteLine("File created successfully");
                File.WriteAllText(path, "Welcome to programming");
            }
            string content = File.ReadAllText(path);
            Console.WriteLine($"[Thread {Thread.CurrentThread.ManagedThreadId}] Read from {path}:");
            Console.WriteLine(content);
        }
        catch (Exception ex)
        {
```

```
            Console.WriteLine($"Error reading {path}: {ex.Message}");
        }
    }

    static async Task ReadFileAsync(string path)
    {
        try
        {
            if (!File.Exists(path))
            {
                File.Create(path).Dispose();
                Console.WriteLine("File created successfully");
                File.WriteAllText(path, "Welcome to programming");
            }
            string content = await File.ReadAllTextAsync(path);
            Console.WriteLine($"[Async] Read from {path}:");
            Console.WriteLine(content);
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error reading {path}: {ex.Message}");
        }
    }
}
```

output



2.create delegate use case between teacher class and student class

teacher method should have test_completed() method passed as delegate to student

student class should have a method write_test() which will inturn call the parent delegate

```
using System;

namespace DelegateExample
{

    public delegate void TestDelegate(string message);



    class Teacher
    {
        public void TestCompleted(string message)
        {
            Console.WriteLine($"Teacher received: {message}");
        }
    }
```

```
class Student
{
    private TestDelegate testCallback;

    public Student(TestDelegate testDelegate)
    {
        this.testCallback = testDelegate;
    }

    public void WriteTest()
    {
        Console.WriteLine("Student is writing the test...");

        System.Threading.Thread.Sleep(1000);
        Console.WriteLine("Student completed the test.");

        testCallback("Test is completed by student.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Teacher teacher = new Teacher();
```

```
            Student student = new Student(teacher.TestCompleted);



            student.WriteTest();


            Console.WriteLine("Main program complete.");
        }
    }
}
```

output