Day 4

Dharunya S

Create two threads to read two separate text files

```
namespace DailyTaskThread;
class program
{

    static void Main()
    {
        string filepath = @"D:\Training\Phase 2\new.txt";
        string filepath2 = @"D:\\Training\\Phase 2\\new2.txt";


        Thread file1 = new Thread(() => ReadFile(filepath));
        Thread file2 = new Thread(() => ReadFile(filepath2));

        file1.Start();
        file2.Start();

        file1.Join();
        file2.Join();

        Console.WriteLine("Finished reading files using Threads.\n");
        static void ReadFile(string fileName)
        {
            string content = File.ReadAllText(fileName);
            Console.WriteLine($"Thread reading {fileName}:");
            Console.WriteLine(content);
        }
    }
}
```

Output:

```
Thread reading D:\Training\Phase 2\new.txt:
 Hello world
Thread reading D:\\Training\\Phase 2\\new2.txt:
 Hello world
```

2.Same two files, try to read using the Task Async await.

```
namespace DailyTaskAsync
{

    class Program
```

```
    {
        static async Task Main()
        {
            string filepath1 = @"D:\Training\Phase 2\new.txt";
            string filepath2 = @"D:\Training\Phase 2\new2.txt";

            Task task1 = ReadFileAsync(filepath1);
            Task task2 = ReadFileAsync(filepath2);

            await Task.WhenAll(task1, task2);

            Console.WriteLine("Finished reading files using async/await.\n");
        }

        static async Task ReadFileAsync(string fileName)
        {
            string content = await File.ReadAllTextAsync(fileName);
            Console.WriteLine($"Async reading {fileName}:");
            Console.WriteLine(content);
        }
    }
}
```

Output:

```
Async reading D:\Training\Phase 2\new2.txt:
 Hello world
Async reading D:\Training\Phase 2\new.txt:
 Hello world
Finished reading files using async/await.
```

3. 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.

```
namespace DailytaskDele;

    class Program
{
    public delegate void TestCompletedDelegate();

    static void Main(string[] args)
    {

        Teacher teacher = new Teacher();
```

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

class Teacher
{
    public void TestCompleted()
    {
        Console.WriteLine("Student has completed test notified to teacher.");
    }
}

class Student
{
    public void WriteTest(Program.TestCompletedDelegate notifyTeacher)
    {
        Console.WriteLine("Student writing the test");
        Console.WriteLine("Test completed by student ");
        notifyTeacher();
    }
}
```

OutPut:

```
Student writing the test
Test completed by student
Student has completed test notified to teacher.
```