

3 – DAY – TASK (21-08-2024)

1. OpenableInterface Program Code:

```
using System;

public class Program
{
    public static void Main()
    {
        Console.Write("Enter the letter found in the paper: ");
        char letter = Convert.ToChar(Console.ReadLine());
        switch(letter)
        {
            case 'T' :
                IOpenable treasure = new TreasureBox();
                Console.WriteLine(treasure.OpenSesame());
                break;
            case 'P' :
                IOpenable parachute = new Parachute();
                Console.WriteLine(parachute.OpenSesame());
                break;
        }
    }
}

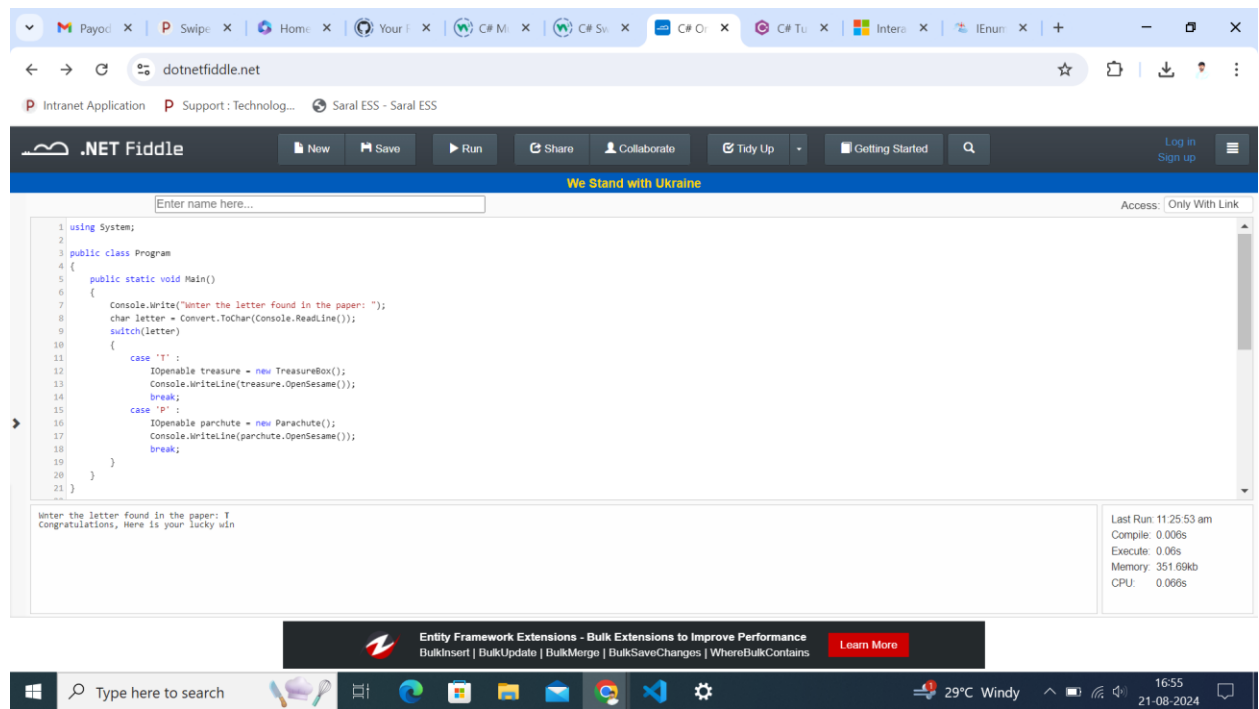
interface IOpenable
{
    string OpenSesame();
}

class TreasureBox : IOpenable
{
    public string OpenSesame()
    {
        return "Congratulations, Here is your lucky win";
    }
}
```

```
}
```

```
class Parachute : IOpenable
{
    public string OpenSesame()
    {
        return "Have a thrilling experience flying in air";
    }
}
```

Output:



2. FlightStatus Program Code:

```
using System;
using System.Collections.Generic;

public class Program
{
    static Dictionary<string, DateTime> flightSchedules = new Dictionary<string,
    DateTime>()
```

```

{
    { "Ar456", new DateTime(DateTime.Now.Year, DateTime.Now.Month,
DateTime.Now.Day, 18, 0, 0) },
    { "Hq457", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 23, 22,
30, 0) },
    { "Tg458", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 22, 20, 15,
0) },
    { "Rs459", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 28, 9, 15,
0) },
    { "Dy460", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 24, 8, 23,
0) },
    { "Zi454", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 23, 5, 40,
0) },
    { "Bm487", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 22, 9, 19,
0) },
};

```

```

public static string FlightStatus(string flightNo)
{
    if (flightSchedules.TryGetValue(flightNo, out DateTime departureTime))
    {
        TimeSpan timeLeft = departureTime - DateTime.Now;
        if (timeLeft.TotalSeconds > 0)
        {
            int days = timeLeft.Days;
            TimeSpan time = timeLeft - TimeSpan.FromDays(days);
            //return $"Time To Flight {days} days {time.ToString(@"hh\:mm\:ss\.ffffff")}";
            //Output
            //Enter the Flight Number: Rs459
            //Time To Flight 6 days 19:58:48.952218
            return $"Time To Flight {timeLeft}";
            //Output
            //Enter the Flight Number: Rs459
            //Time To Flight 6.19:55:58.7112418
        }
        else
        {
            return "Flight Already Left";
        }
    }
}

```

```

    }
}
else
{
    return "Flight Not Found";
}
}

public static void Main()
{
    Console.WriteLine("Enter the Flight Number: ");
    string flightNo = Console.ReadLine();
    Console.WriteLine(FlightStatus(flightNo));
}
}

```

Output:

The screenshot shows the .NET Fiddle web application interface. The browser address bar displays 'dotnetfiddle.net'. The application has a dark theme and a top navigation bar with buttons for 'Now', 'Save', 'Run', 'Share', 'Collaborate', 'Tidy Up', and 'Getting Started'. A banner for 'We Stand with Ukraine' is visible. On the left, there are settings for 'Language' (C#), 'Project Type' (Console), 'Compiler' (NET 8), and 'NuGet Packages'. The main editor area contains the following C# code:

```

1 using System;
2 using System.Collections.Generic;
3
4 public class Program
5 {
6     static Dictionary<string, DateTime> flightSchedules = new Dictionary<string, DateTime>()
7     {
8         { "A456", new DateTime(DateTime.Now.Year, DateTime.Now.Month, DateTime.Now.Day, 18, 0, 0) },
9         { "H457", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 23, 22, 30, 0) },
10        { "G458", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 22, 20, 15, 0) },
11        { "R459", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 28, 0, 15, 0) },
12        { "D460", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 24, 0, 23, 0) },
13        { "Z464", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 23, 5, 40, 0) },
14        { "B487", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 22, 9, 19, 0) },
15    };
16
17     public static string FlightStatus(string flightNo)
18     {
19
20     }
21 }

```

Below the code editor, the input 'Enter the flight Number: R459' is shown, followed by the output 'Time To Flight 6:19:55:58.7112418'. On the right, performance metrics are displayed: 'Last Run: 1:19:01 pm', 'Compile: 0.007s', 'Execute: 0.07s', 'Memory: 1.22Mb', and 'CPU: 0.077s'. At the bottom, there is a blue banner that says 'Looking for the best .NET Developers?' and a Windows taskbar with the date '21-08-2024' and time '18:57'.

3. ProductDetails Program Code:

```

using System;
using System.Collections.Generic;

```

```

public class Program
{
    static Dictionary<string, DateTime> flightSchedules = new Dictionary<string,
DateTime>()
    {
        { "Ar456", new DateTime(DateTime.Now.Year, DateTime.Now.Month,
DateTime.Now.Day, 18, 0, 0) },
        { "Hq457", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 23, 22,
30, 0) },
        { "Tg458", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 22, 20, 15,
0) },
        { "Rs459", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 28, 9, 15,
0) },
        { "Dy460", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 24, 8, 23,
0) },
        { "Zi454", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 23, 5, 40,
0) },
        { "Bm487", new DateTime(DateTime.Now.Year, DateTime.Now.Month, 22, 9, 19,
0) },
    };

    public static string FlightStatus(string flightNo)
    {
        if (flightSchedules.TryGetValue(flightNo, out DateTime departureTime))
        {
            TimeSpan timeLeft = departureTime - DateTime.Now;
            if (timeLeft.TotalSeconds > 0)
            {
                int days = timeLeft.Days;
                TimeSpan time = timeLeft - TimeSpan.FromDays(days);
                //return $"Time To Flight {days} days {time.ToString(@"hh\:mm\:ss\.ffffff")}";
                //Output
                //Enter the Flight Number: Rs459
                //Time To Flight 6 days 19:58:48.952218
                return $"Time To Flight {timeLeft}";
                //Output
                //Enter the Flight Number: Rs459
                //Time To Flight 6.19:55:58.7112418
            }
        }
    }
}

```

```

    }
    else
    {
        return "Flight Already Left";
    }
}
else
{
    return "Flight Not Found";
}
}

public static void Main()
{
    Console.WriteLine("Enter the Flight Number: ");
    string flightNo = Console.ReadLine();
    Console.WriteLine(FlightStatus(flightNo));
}
}

```

Output:

The screenshot shows the .NET Fiddle web application interface. The main editor displays a C# program that defines a list of products and a Main method that iterates through them, printing their details. The output window shows the results of the program execution, displaying a table of product information.

Product Data:

Product Name	Serial Number	Purchase Date	Purchase Cost
Apple	Prod1234	2024:28:21	200
Hair Trimmer	Prod1276	2024:28:21	1200
Steel Box	Prod1289	2024:28:21	400
Rope	Prod1213	2024:28:21	99
Chair	Prod1342	2024:28:21	309

Performance Metrics:

- Last Run: 1:28:00 pm
- Compile: 0.007s
- Execute: 0.08s
- Memory: 1.47Mb
- CPU: 0.087s