

Date : 22-08-2024

1.

Code :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
namespace OpenableInterface
{
```

```
    interface IOpenable
    {
        string OpenSesame();
    }
```

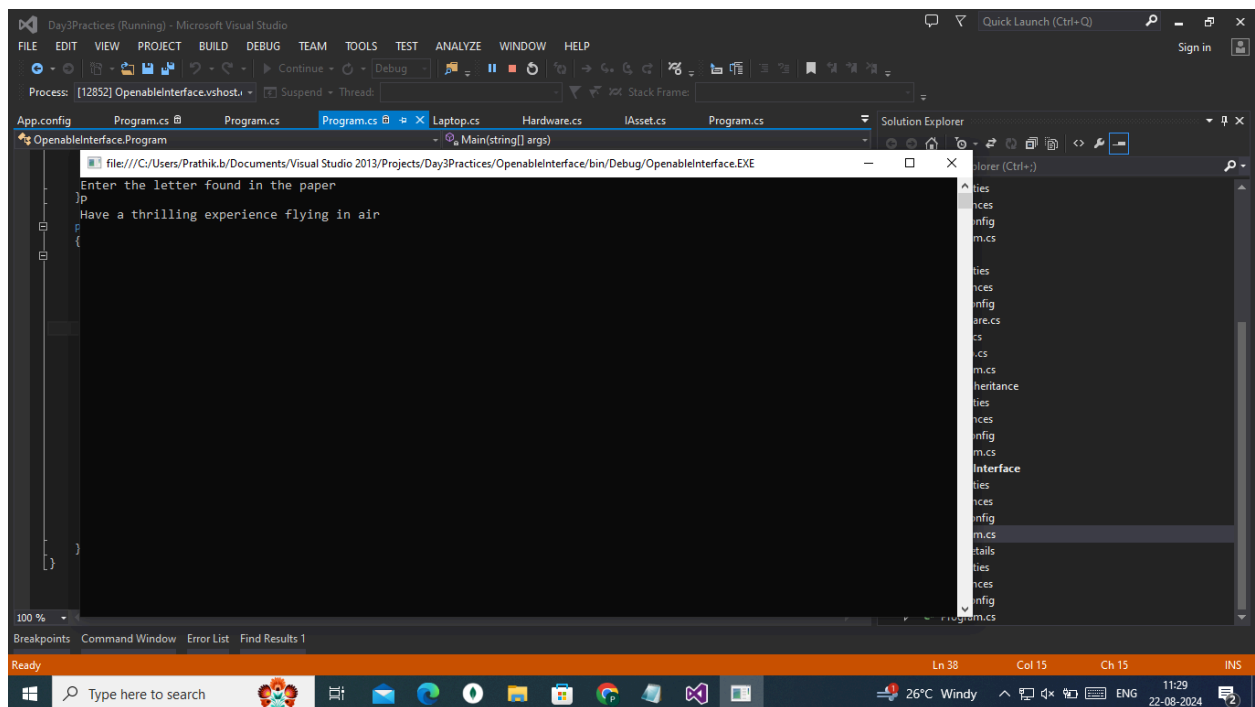
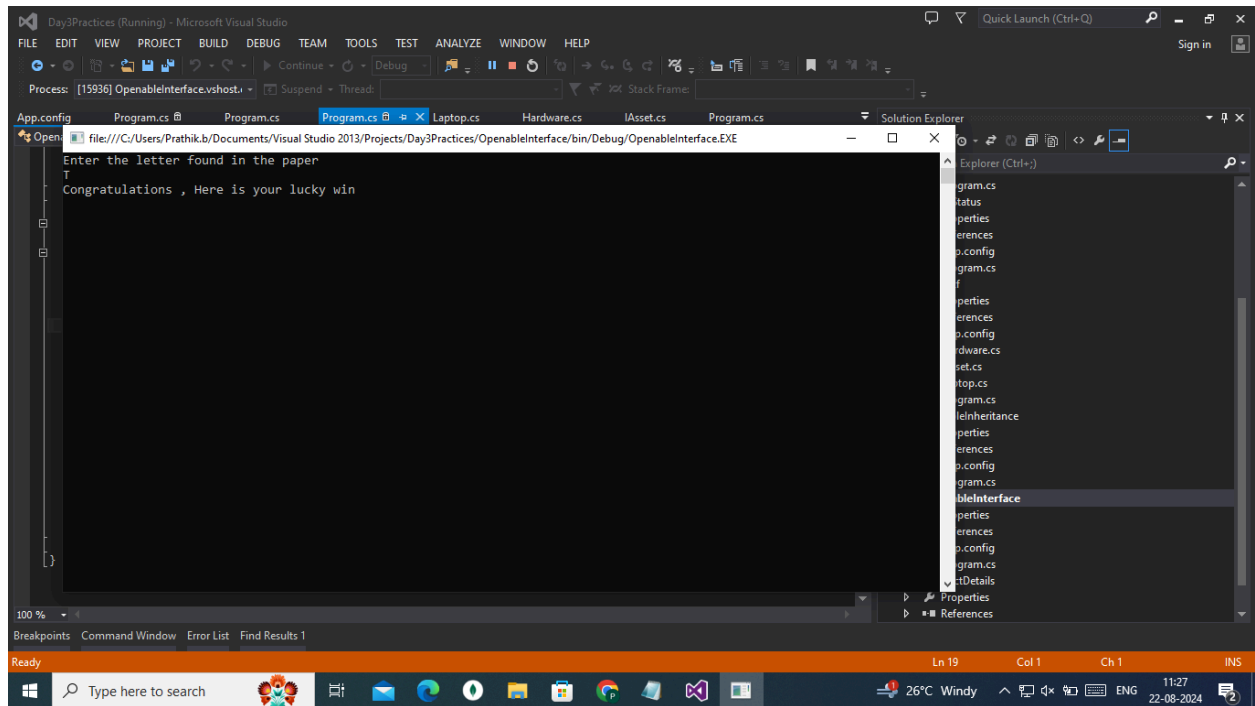
```
    class TreasureBox : IOpenable
    {
        public string OpenSesame()
        {
            return "Congratulations , Here is your lucky win";
        }
    }
```

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

```
    public class Program
    {
```

```
static void Main(string[] args)
{
    Console.WriteLine("Enter the letter found in the paper");
    char c = Convert.ToChar(Console.ReadLine());
    if (c == 'T')
    {
        TreasureBox tb = new TreasureBox();
        Console.WriteLine(tb.OpenSesame());
    }
    else if (c == 'P')
    {
        Parachute p = new Parachute();
        Console.WriteLine(p.OpenSesame());
    }
    else
    {
        Console.WriteLine("Better Luck Next Time");
    }
    Console.ReadKey();
}
}
```

Output :



2.

Code :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace FlightStatus
{
    class Program
    {

        static Dictionary<string, DateTime> flights = new Dictionary<string,DateTime>();

        static string FlightStatus(string Fno)
        {
            DateTime time = flights[Fno];
            if (DateTime.Now < time)
            {
                TimeSpan res = time.Subtract(DateTime.Now);
                return "Time To Flight " + res;
            }
            return "Flight Already Left";
        }

        static void Main(string[] args)
        {

            flights.Add("ZW346", Convert.ToDateTime("15:30:30"));
            flights.Add("BR267", Convert.ToDateTime("07:30:30"));
            flights.Add("IN147", Convert.ToDateTime("20:30:30"));

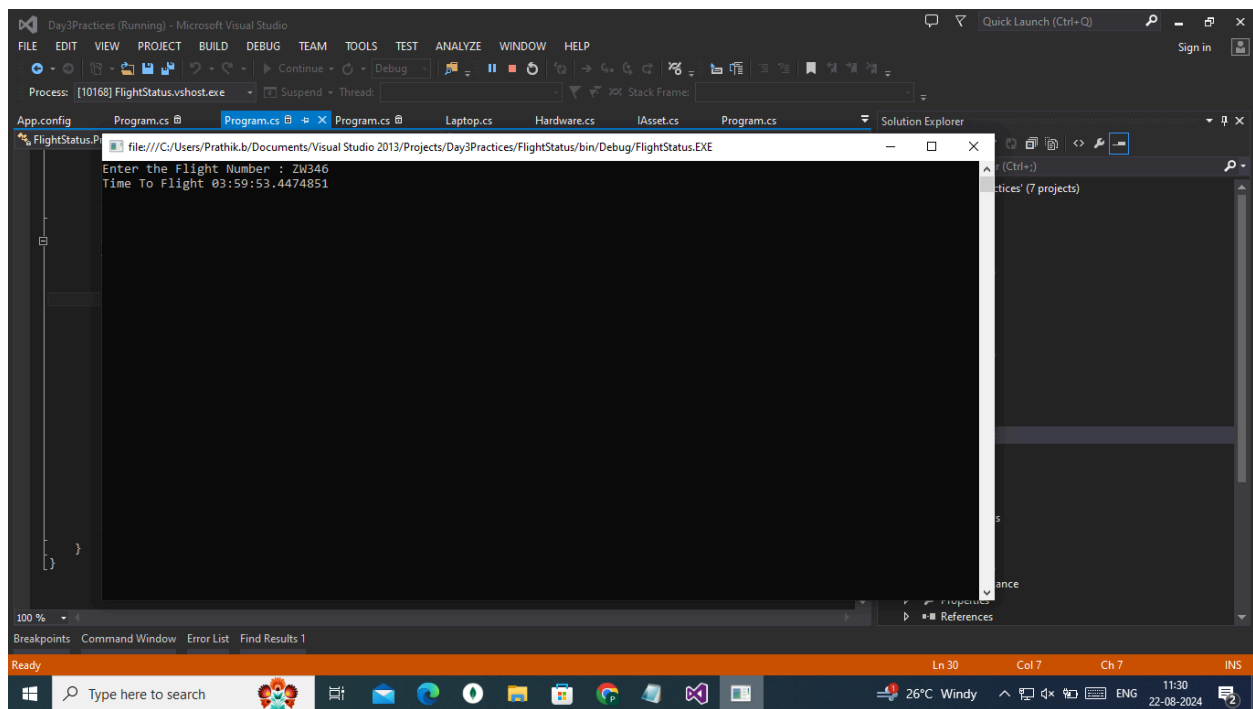
            Console.Write("Enter the Flight Number : ");
            string Fno = Console.ReadLine();
```

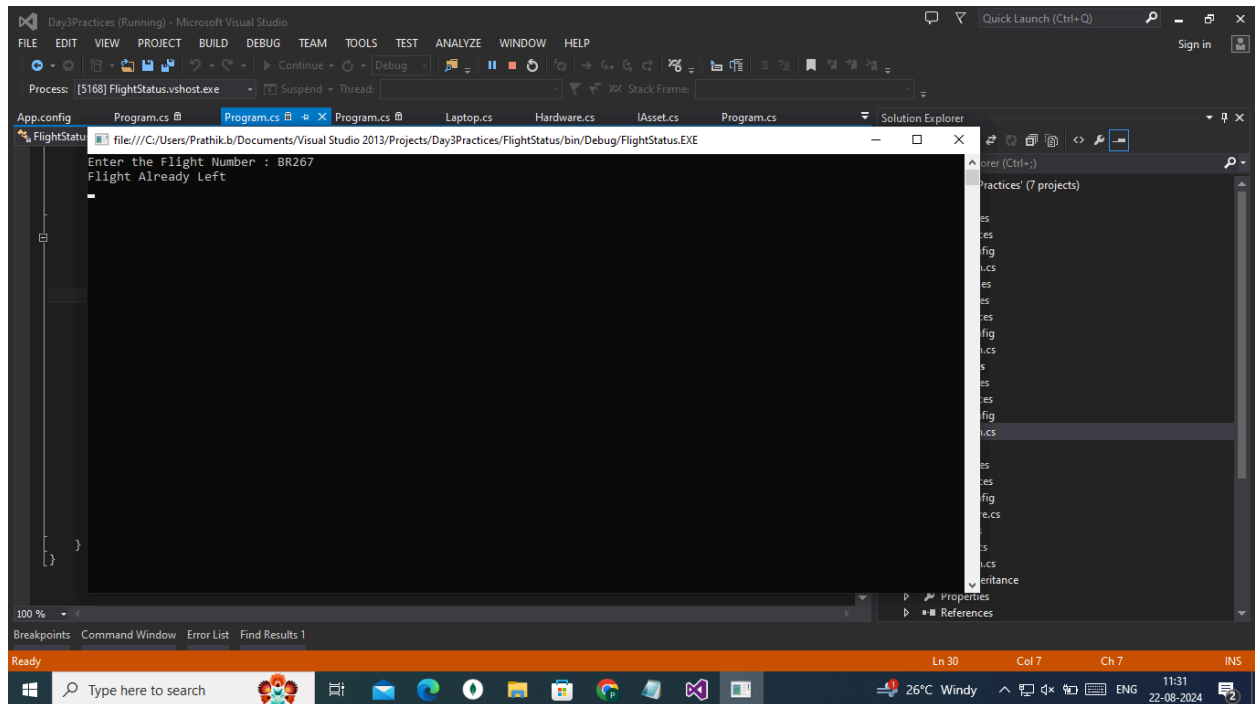
```

        if (flights.ContainsKey(Fno))
        {
            Console.WriteLine(FlightStatus(Fno));
        }
        else
        {
            Console.WriteLine("Invalid Flight Number");
        }
        Console.ReadKey();
    }
}
}

```

Output :





3.

Code :

```
using System;
```

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

```
using System.Linq;
```

```
using System.Text;
```

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

```
namespace ProductDetails
```

```
{
```

```
    class Product
```

```
    {
```

```
        public string _productName { get; set; }
```

```
        public string _serialNumber { get; set; }
```

```
        public DateTime _purchaseDate { get; set; }
```

```
        public double _cost { get; set; }
```

```
        public Product(string PName, string SNumber, DateTime PDate, double cost)
```

```
        {
```

```
_productName = PName;  
_serialNumber = SNumber;  
_purchaseDate = PDate;  
_cost = cost;  
}
```

```
public override string ToString()  
{  
    return String.Format("{0,-15}{1,-15}{2,-15}{3,-15}",  
        _productName,  
        _serialNumber,  
        _purchaseDate.ToString("dd-MM-yyyy"),  
        _cost);  
}  
}
```

```
class Program  
{  
    static void Main(string[] args)  
    {  
        List<Product> list = new List<Product>();  
  
        list.Add(new  
Product("Bottle","SN7543456",Convert.ToDateTime("22-08-2024"),200.00));  
        list.Add(new Product("Lunch Box", "SN9874821",  
Convert.ToDateTime("22-08-2024"), 150.00));  
        list.Add(new Product("Towel", "SN7253178", Convert.ToDateTime("22-08-2024"),  
75.00));  
  
        Console.WriteLine(String.Format("{0,-15}{1,-15}{2,-15}{3,-15}", "Product Name",  
"Serial Number", "Purchase Date", "Purchase Cost"));  
  
        foreach (var li in list)  
        {  
            Console.WriteLine(li.ToString());  
        }  
    }  
}
```

Console.ReadKey();

```
}  
}  
}
```

Output :

