Day 18 – Assignment

16-Feb,2022

By Manoj Karnatapu - NBHealthCareTechnologies

nb healthcare technologies

An IT division of NationsBenefits, LLC. USA.

Assignment 1
What is the use of XML
Answer
• Xml is used for Universal data transfer mechanism to send data across different platforms. • User can define, his own tags. Hence it is called as user Defined Tags. • It is not a Platform dependant. It can be used in any platform with ease. • The redundancy in syntax of XML causes higher storage and transportation cost when the volume of data is large.

Assignment 2
Write the points discussed about xml in the class
Answer
• XML stands for extensible Markup Language. • XML has user defined tags. • It is a Case Sensitive. • XML Has only one Root tag as an Entry. • It's not a platform dependant. • XML syntax is verbose and redundant compared to other text-based data transmission formats such as JSON.

Assignment 3

Create a simple xml to illustrate:

a. Tag based xml with 10 products

b. Attribute based xml .

Answer

(a). Tag Based XML

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
▼<Electronics>
  ▼<Mobile>
     <ID>1</ID>
     <Name>RedMi Note 3</Name>
     <Price>13000</Price>
   </Mobile>
  ▼<Mobile>
     <ID>2</ID>
     <Name>RedMi Note 6</Name>
     <Price>16000</Price>
   </Mobile>
  ▼<Mobile>
     <ID>3</ID>
     <Name>RealMe Note 11</Name>
     <Price>14999</Price>
   </Mobile>
  ▼<Mobile>
     <ID>4</ID>
     <Name>Vivo 11 Pro</Name>
     <Price>18000</Price>
   </Mobile>
  ▼<Mobile>
     <ID>5</ID>
     <Name>Oppo Reno</Name>
     <Price>15999</Price>
   </Mobile>
  ▼<Mobile>
     <ID>6</ID>
     <Name>OnePlus 9R</Name>
     <Price>48999</Price>
   </Mobile>
  ▼<Mobile>
     <ID>7</ID>
     <Name>Iphone 13</Name>
     <Price>103000</Price>
   </Mobile>
  ▼<Mobile>
     <ID>8</ID>
     <Name>Samsung Galaxy 9</Name>
     <Price>79989</Price>
   </Mobile>
  ▼<Mobile>
     <ID>9</ID>
     <Name>Nokia Lumia 990</Name>
     <Price>86949</Price>
   </Mobile>
  ▼<Mobile>
     <ID>10</ID>
     <Name>Surface Tab 6</Name>
     <Price>98999</Price>
   </Mobile>
 </Electronics>
```

(b). Attribute Based XML

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<Electronics>
   <Mobile ID="1" Name="RedMi Note 3" Price="13000"/>
   <Mobile ID="2" Name="RedMi Note 6" Price="16000"/>
   <Mobile ID="3" Name="RealMe Note 11" Price="14999"/>
   <Mobile ID="4" Name="Vivo 11 Pro" Price="18000"/>
   <Mobile ID="5" Name="Oppo Reno" Price="15999"/>
   <Mobile ID="6" Name="OnePlus 9R" Price="48999"/>
   <Mobile ID="7" Name="Iphone 13" Price="103000"/>
   <Mobile ID="8" Name="Samsung Galaxy 9" Price="79989"/>
   <Mobile ID="9" Name="Nokia Lumia 990" Price="86949"/>
   <Mobile ID="10" Name="Surface Tab 6" Price="98999"/>
 </Electronics>
```

Assignment 4

Convert the above xml to JSON and display the JSON data

Answer

JSON Format of Assignment-2 XML is :

XML_to_JSON.json

Schema: <No Schema Selected>


```
[
  {
    "ID": "1",
    "Name": "RedMi Note 3",
    "Price": "13000"
  },
  {
    "ID": "2",
    "Name": "RedMi Note 6",
    "Price": "16000"
  },
  {
    "ID": "3",
    "Name": "RealMe Note 11",
    "Price": "14999"
  },
  {
    "ID": "4",
    "Name": "Vivo 11 Pro",
    "Price": "18000"
  },
  {
    "ID": "5",
    "Name": "Oppo Reno",
    "Price": "15999"
  },
  {
    "ID": "6",
    "Name": "OnePlus 9R",
    "Price": "48999"
  },
  {
    "ID": "7",
    "Name": "Iphone 13",
    "Price": "103000"
  },
  {
    "ID": "8",
    "Name": "Samsung Galaxy 9",
    "Price": "79989"
  },
  {
    "ID": "9",
    "Name": "Nokia Lumia 990",
    "Price": "86949"
  },
  {
    "ID": "10",
    "Name": "Surface Tab 6",
    "Price": "98999"
  }
]
```

Assignment 5

Research and write the benefits of JSON over XML. (2 or 3 points)

Answer

- JSON Occupies less file size, comparatively to XML file.
- JSON is faster because it is designed specifically for data interchange.
- JSON encoding is terse, which requires less bytes for transit.
- JSON parsers are less complex, which requires less processing time and memory overhead.
- XML is slower, because it is designed for a lot more than just data interchange.

Assignment 6

For the below requirement, create a layered architecture project with separate class library for Business logic.

- create console application
- create windows(or desktop) application

Business Requirement:

FIND FACTORIAL OF A NUMBER:

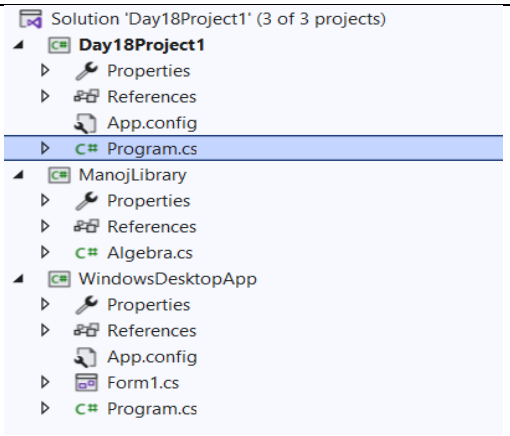
If 0 = 1, positive number (up to 7) = factorial answer, > 7 = -999 (as answer), < 0 = -9999 (as answer)

put the screen shots of the output and project (solution explorer) screen shot

Code

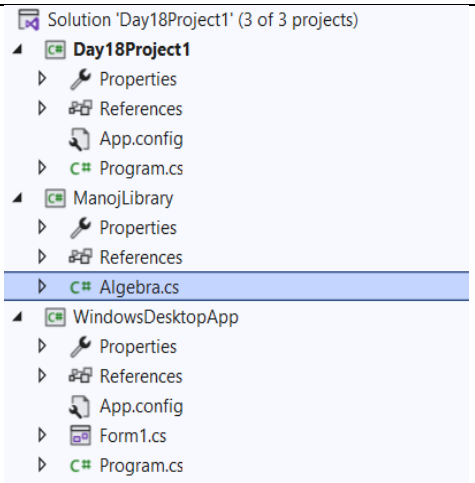
Console App :

```
using System;
using ManojLibrary;

// Author : Manoj.Karnatapu
// Purpose : Creating a Layered Architecture for given Business Requirements In Console
Application.

// For Reference, Check Day18Project1 in the same Repository.
namespace Day18Project1
{
    internal class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter Any Number : ");
            int n = int.Parse(Console.ReadLine());
            Console.WriteLine("\n\n Factorial Result is : {0}",Algebra.Factorial(n));

            Console.ReadLine();
        }
    }
}
```



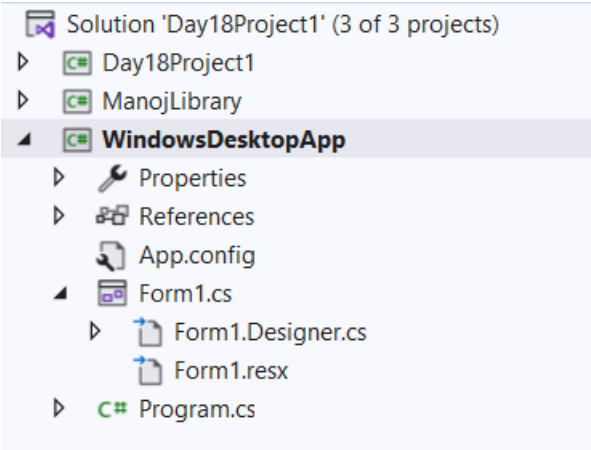
ManojLibrary :

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

// Author : Manoj.Karnatapu
// Purpose : As per Business Requirements, Creating a Factorial Library.

// for Reference, check Day18Project1 in the same Repository.
namespace ManojLibrary
{
    public class Algebra
    {
        public static int Factorial(int n)
        {
            if (n == 0)
                return 1;
            else if (n < 0)
                return -9999;
            else if (n > 7)
                return -999;
            else
            {
                int fact = 1;
                for (int i = 1;i <= n; i++)
                {
                    fact *= i;
                }
                return fact;
            }
        }
    }
}
```



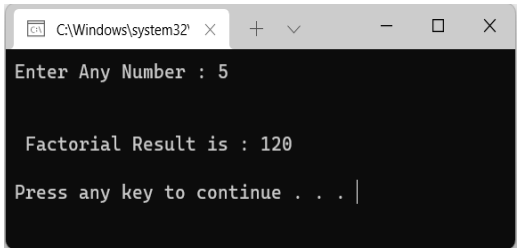
Windows Desktop App :

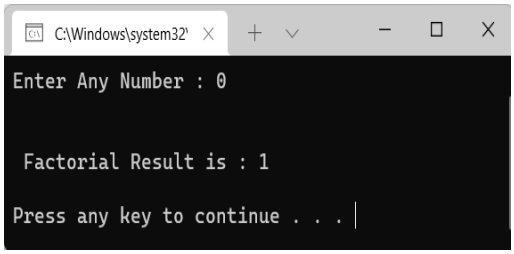
```
using System;
using System.Windows.Forms;
using ManojLibrary;

// Author : Manoj.Karnatapu
// Purpose : Creating a Layered Architecture for given
//           Business Requirements In Windows Desktop Application.

// For Reference, Check Day18Project1 in the same Repository.

namespace WindowsDesktopApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            int n = int.Parse(textBox1.Text);
            int result = Algebra.Factorial(n);
            textBox2.Text = result.ToString();

            Console.ReadLine();
        }
    }
}
```

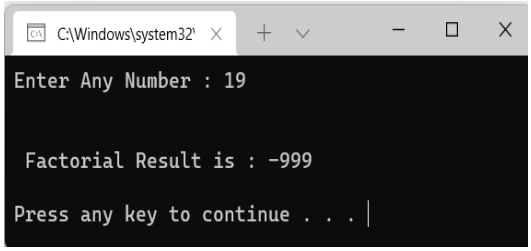


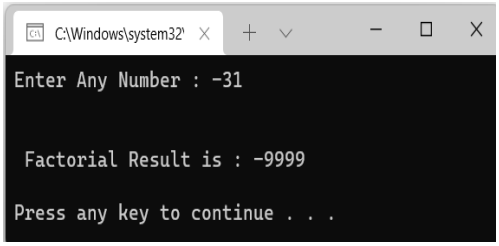
Output

Console App Outputs:

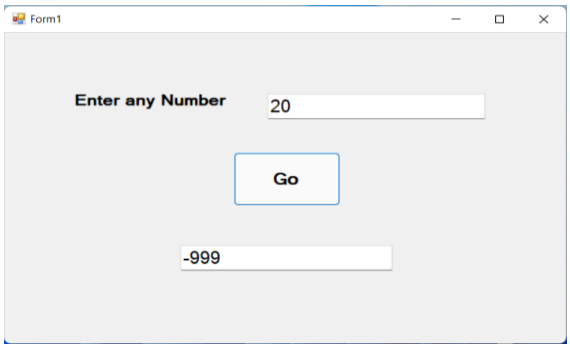


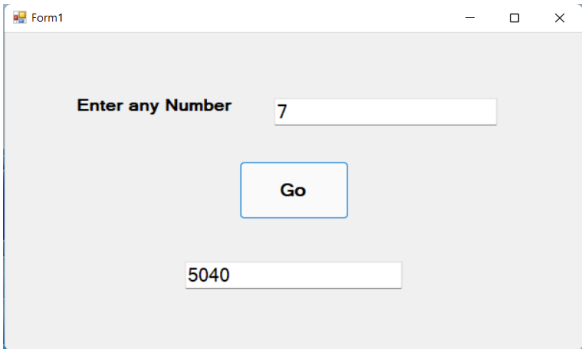


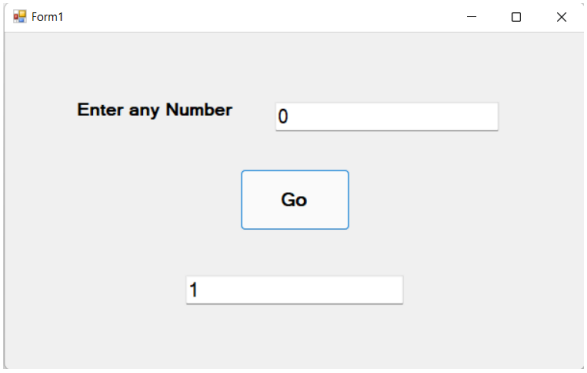


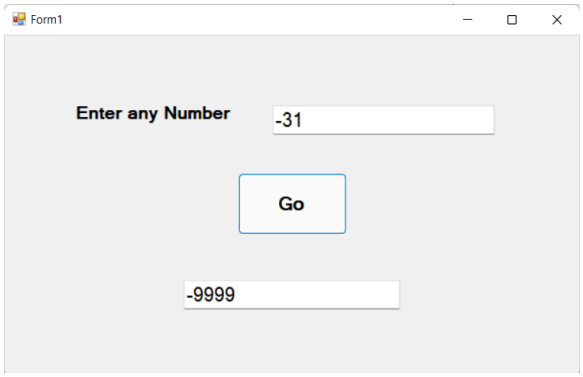


Windows Desktop Outputs:









Assignment 7

For the above method, Implement TDD
and write 4 test cases and put the code in word document.
put the screen shot of all test cases failing. Make the test cases pass. **Put** the screen shot

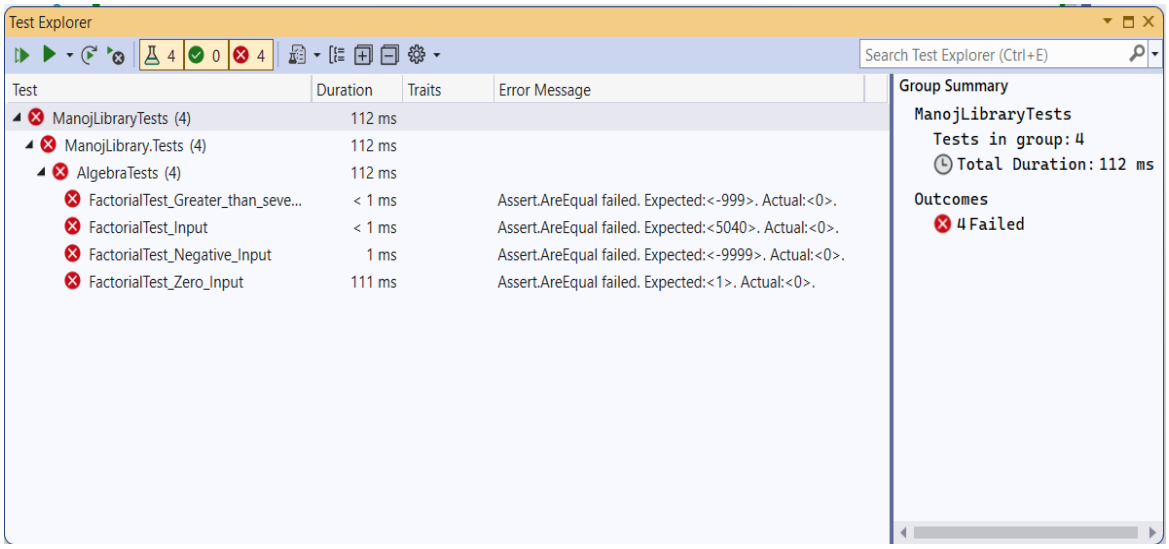
Code

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using ManojLibrary;

// Author : Manoj.Karnatapu
// Purpose : Creating Test Case, For Algebra Library, For Factorial Method.

namespace ManojLibrary.Tests
{
    [TestClass()]
    public class AlgebraTests
    {
        [TestMethod()]
        public void FactorialTest_Zero_Input()
        {
            // Arrange
            int n = 0;
            int expected = 1;
            // Act
            int actual = Algebra.Factorial(n);

            // Assert
            Assert.AreEqual(expected, actual);
        }

        [TestMethod()]
        public void FactorialTest_Negative_Input()
        {
            // Arrange
            int n = -31;
            int expected = -9999;
            // Act
            int actual = Algebra.Factorial(n);

            // Assert
            Assert.AreEqual(expected, actual);
        }

        [TestMethod()]
        public void FactorialTest_Greater_than_seven_Input()
        {
            // Arrange
            int n = 20;
            int expected = -999;
            // Act
            int actual = Algebra.Factorial(n);

            // Assert
            Assert.AreEqual(expected, actual);
        }

        [TestMethod()]
        public void FactorialTest_Input()
        {
            // Arrange
            int n = 7;
            int expected = 5040;
            // Act
            int actual = Algebra.Factorial(n);

            // Assert
```

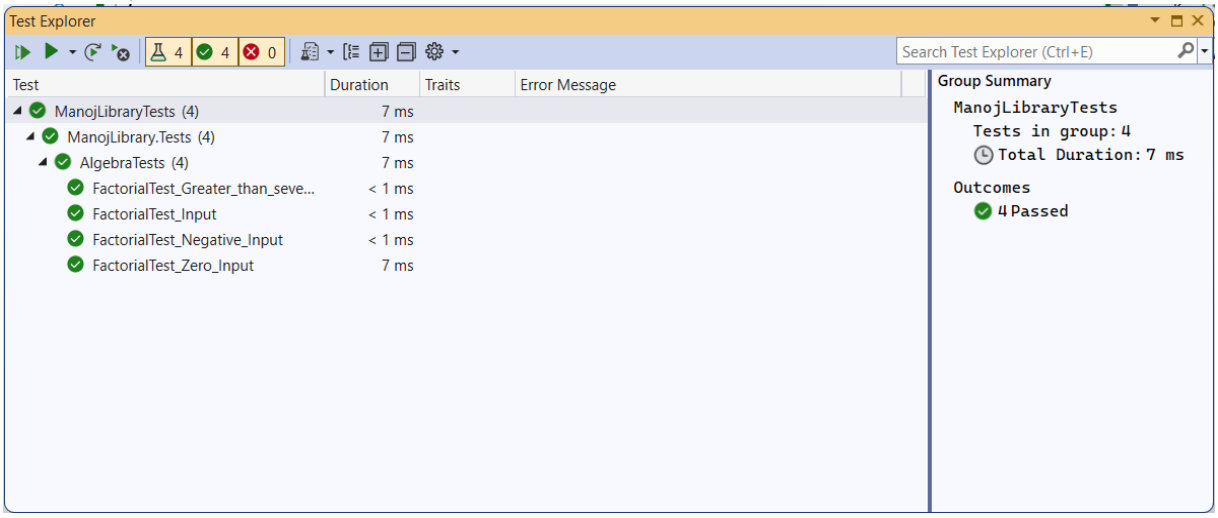
```
            Assert.AreEqual(expected, actual);
        }
    }
}
```

Output

All Test Cases Failed.



All Test Cases, Succussed.



Assignment 8

Add one more method to check if the number is palindrome or not in the above Algebra class and write test case for the same.

Code

Algebra Library :

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

// Author : Manoj.Karnatapu
// Purpose : As per Business Requirements, Creating a Factorial Library.
```

```
// for Reference, check Day18Project1 in the same Repository.
namespace ManojLibrary
{
    public class Algebra
    {
        public static int Factorial(int n)
        {
            if (n == 0)
                return 1;
            else if (n < 0)
                return -9999;
            else if (n > 7)
                return -999;
            else
            {
                int fact = 1;
                for (int i = 1;i <= n; i++)
                {
                    fact *= i;
                }
                return fact;
            }
        }

        public static bool IsPalindrome(int n)
        {
            int rev = 0, rem, m;
            m = n;
            while (m > 0)
            {
                rem = m % 10;
                m = m / 10;
                rev = rev * 10 + rem;
            }
            if (n == rev)
                return true;
            else
                return false;
        }
    }
}
```

ManojLibrary. Tests :

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using ManojLibrary;

// Author : Manoj.Karnatapu
// Purpose : Creating Test Case, For Algebra Library, For Factorial Method & IsPalindrome.

namespace ManojLibrary.Tests
{
    [TestClass()]
    public class AlgebraTests
    {
        [TestMethod()]
        public void FactorialTest_Zero_Input()
        {
            // Arrange
            int n = 0;
            int expected = 1;
            // Act
            int actual = Algebra.Factorial(n);
```

```
            // Assert
            Assert.AreEqual(expected, actual);
        }

        [TestMethod()]
        public void FactorialTest_Negative_Input()
        {
            // Arrange
            int n = -31;
            int expected = -9999;
            // Act
            int actual = Algebra.Factorial(n);

            // Assert
            Assert.AreEqual(expected, actual);
        }

        [TestMethod()]
        public void FactorialTest_Greater_than_seven_Input()
        {
            // Arrange
            int n = 20;
            int expected = -999;
            // Act
            int actual = Algebra.Factorial(n);

            // Assert
            Assert.AreEqual(expected, actual);
        }

        [TestMethod()]
        public void FactorialTest_Input()
        {
            // Arrange
            int n = 7;
            int expected = 5040;
            // Act
            int actual = Algebra.Factorial(n);

            // Assert
            Assert.AreEqual(expected, actual);
        }

        [TestMethod()]
        public void PalindromeTest_Input()
        {
            // Arrange
            int n = 131;
            bool expected = true;

            // Act
            bool actual = Algebra.IsPalindrome(n);

            // Assert
            Assert.AreEqual(expected, actual);
        }
    }
}
```

Output

All Test Cases, succeeded including Palindrome Test.