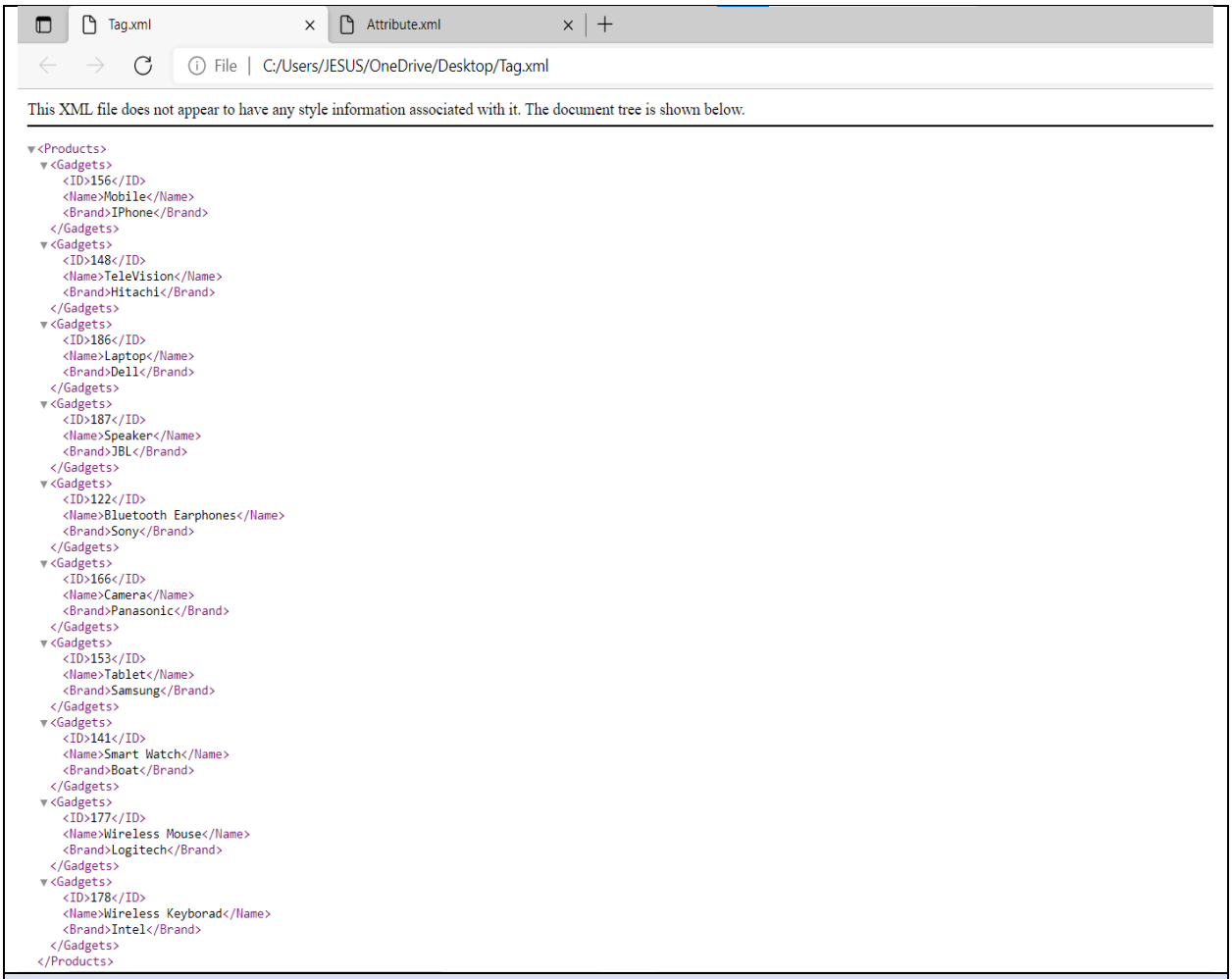
Day 18 Assignment
By M Mary Margarette
On 16-02-2022

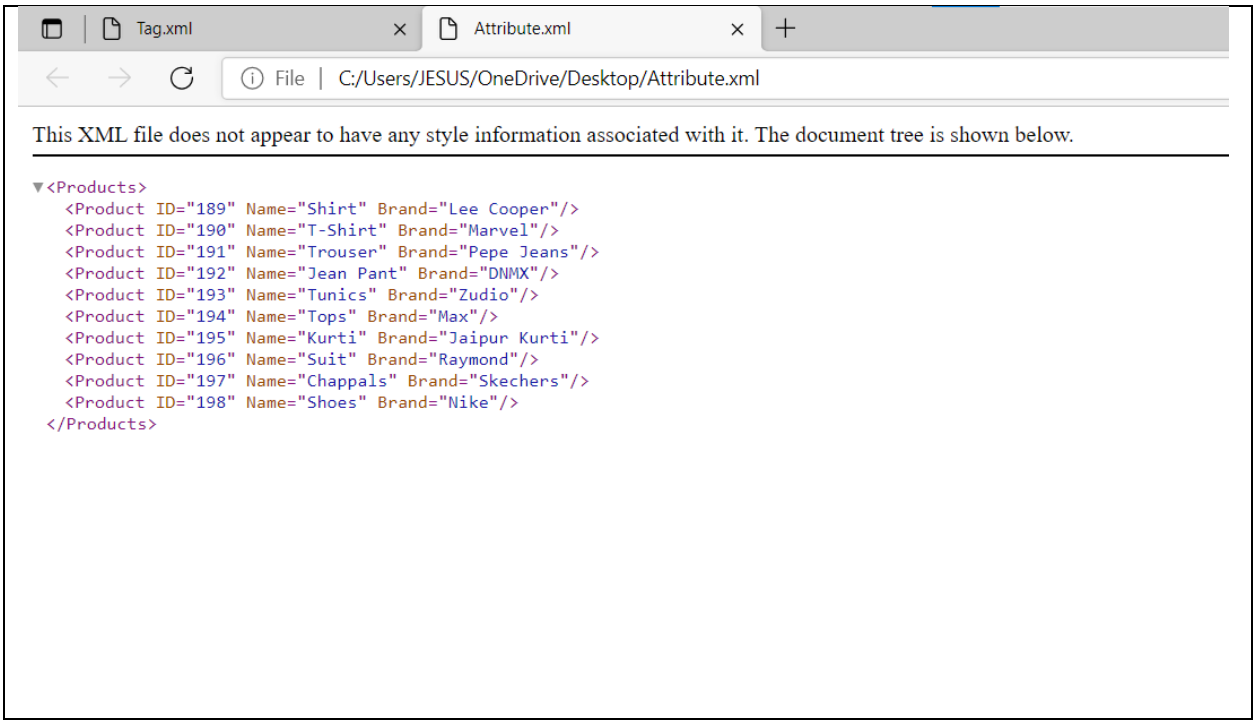
What is the use of XML
• Xml is used to transferred data mechanism to send data across different forms.
• Xml was designed to store and transport data.
• It simplifies data transport.

Write the points discussed about xml in the class
• Xml haves only single root tag.
• It can have any no.of tags in root tag.
• Xml stands for Extensible Markup Language.
• Xml shows error when case sensitive.
• Xml used for universal data transfer mechanism to send data across different platforms.
• Xml will have user defined tags.

Create a simple xml to illustrate: a. Tag based xml with 10 products b. Attribute based xml
Tag Based xml with 10 products



Attribute based xml



Convert the above xml to JSON and display the JSON data

Tag Based xml with 10 products

```
{
  "Products": {
    "Gadgets": [
      {
        "ID": "156",
        "Name": "Mobile",
        "Brand": "iPhone"
      },
      {
        "ID": "148",
        "Name": "Television",
        "Brand": "Hitachi"
      },
      {
        "ID": "186",
        "Name": "Laptop",
        "Brand": "Dell"
      },
      {
        "ID": "187",
        "Name": "Speaker",
        "Brand": "JBL"
      },
      {
```

```
        "ID": "122",
        "Name": "Bluetooth Earphones",
        "Brand": "Sony"
      },
      {
        "ID": "166",
        "Name": "Camera",
        "Brand": "Panasonic"
      },
      {
        "ID": "153",
        "Name": "Tablet",
        "Brand": "Samsung"
      },
      {
        "ID": "141",
        "Name": "Smart Watch",
        "Brand": "Boat"
      },
      {
        "ID": "177",
        "Name": "Wireless Mouse",
        "Brand": "Logitech"
      },
      {
        "ID": "178",
        "Name": "Wireless Keyboard",
        "Brand": "Intel"
      }
    ]
  },
  "#omit-xml-declaration": "yes"
}
```

Attribute based xml

```
{
  "Products": {
    "Product": [
      {
        "-ID": "189",
        "-Name": "Shirt",
        "-Brand": "Lee Cooper"
      },
      {
        "-ID": "190",
        "-Name": "T-Shirt",
        "-Brand": "Marvel"
      },
      {
        "-ID": "191",
        "-Name": "Trouser",
```

```
        "-Brand": "Pepe Jeans"
      },
      {
        "-ID": "192",
        "-Name": "Jean Pant",
        "-Brand": "DNMX"
      },
      {
        "-ID": "193",
        "-Name": "Tunics",
        "-Brand": "Zudio"
      },
      {
        "-ID": "194",
        "-Name": "Tops",
        "-Brand": "Max"
      },
      {
        "-ID": "195",
        "-Name": "Kurti",
        "-Brand": "Jaipur Kurti"
      },
      {
        "-ID": "196",
        "-Name": "Suit",
        "-Brand": "Raymond"
      },
      {
        "-ID": "197",
        "-Name": "Chappals",
        "-Brand": "Skechers"
      },
      {
        "-ID": "198",
        "-Name": "Shoes",
        "-Brand": "Nike"
      }
    ]
  },
  "#omit-xml-declaration": "yes"
}
```


Research and write the benefits of JSON over XML (2 or 3 points)
▪ JSON is faster because it is designed for data interchange only. XML is slower because it is designed for a lot more than just data interchange.
▪ JSON occupies less memory than XML.
▪ JSON is a light weight data exchange format Where XML parsing always consumes lot of browser resources.
▪ JSON is easy and faster to parse when compared to XML.

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: FINDING FACTORIAL OF A NUMBER: 0 = 1 positive number (up to 7) = factorial > 7 = -999 (as answer) < 0 = -9999 (as answer) put the screen shots of the output and project (solution explorer) screen shot
Console Application

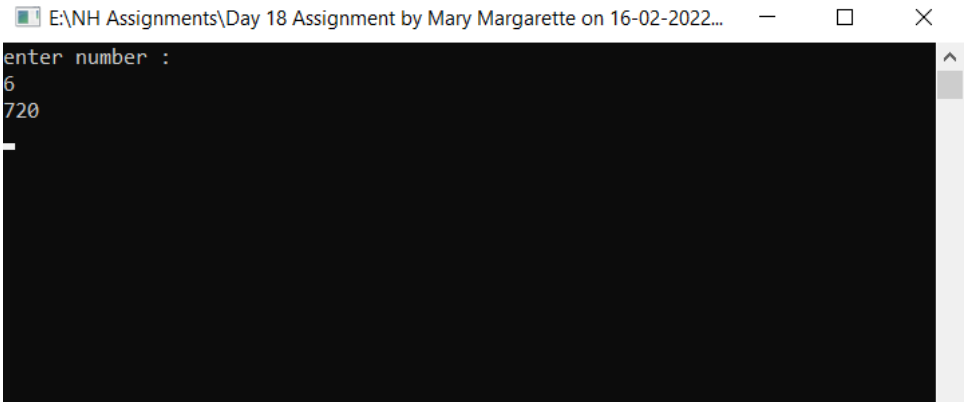
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Mathematics
{
    1 reference
    public class Algebra
    {
        1 reference
        public static int Factorial(int n)
        {
            int fact = 1;

            if (n == 0)
            {
                return 0;
            }
            else if (n > 7)
            {
                return -999;
            }
            else if (n < 0)
            {
                return -9999;
            }
            else
            {
                for(int i = 1; i <= n; i++)
                {

                    fact = fact * i;
                }
                return fact;
            }
        }
    }
}
```

Algebra.cs | Program.cs

Day 18 Project 1 | Day_18_Project_1.Program

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6   using Mathematics;
7
8   namespace Day_18_Project_1
9   {
        0 references
10      internal class Program
11      {
            0 references
12          static void Main(string[] args)
13          {
14              int n;
15              Console.WriteLine("enter number :");
16              n=Convert.ToInt32(Console.ReadLine());
17              Console.WriteLine(Algebra.Factorial(6));
18              Console.ReadLine();
19          }
20      }
21  }
```

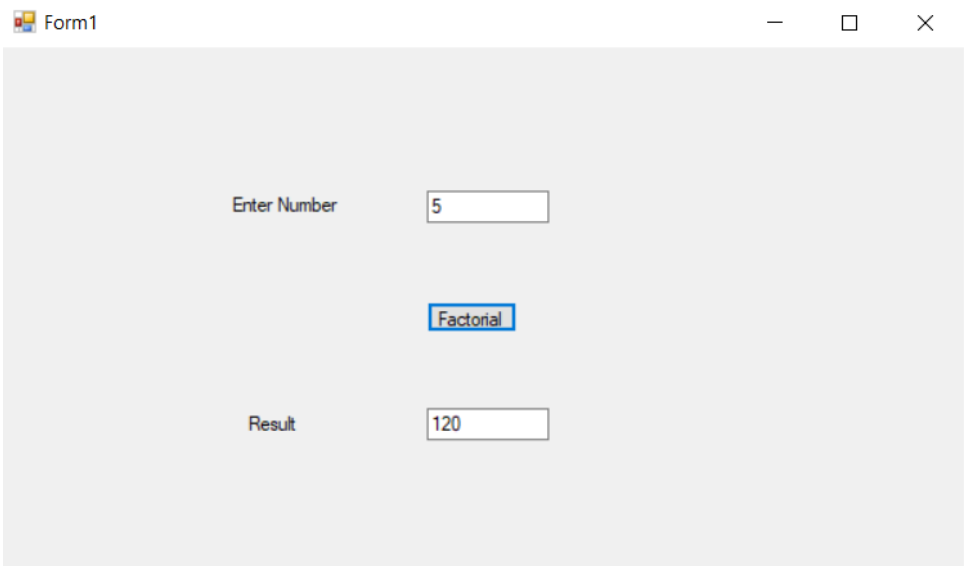
Console Application Output:



Windows Form Application:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Mathematics;

namespace WindowsForm
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            int n = Convert.ToInt32(textBox1.Text);
            int result = Algebra.Factorial(n);
            textBox2.Text=result.ToString();
        }
    }
}
```

Windows Form Application Output:

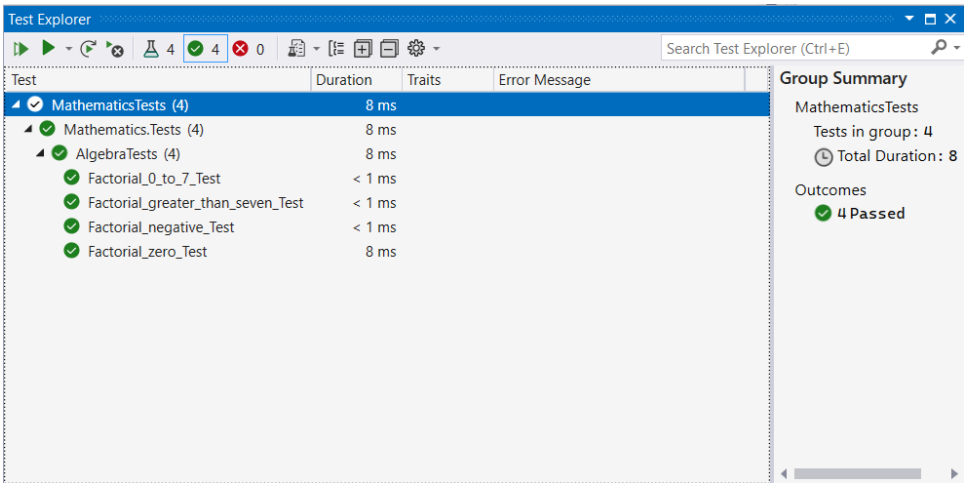


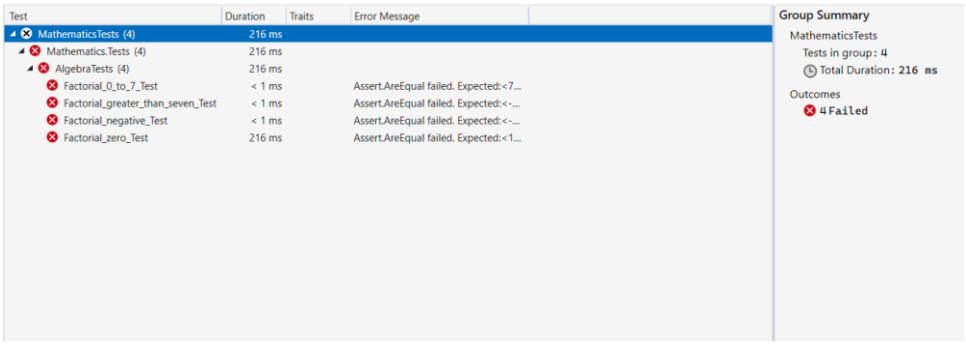
For the above method, Implement TDD and write 4 test cases and put the code in a word document.
put the screen shot of all test cases failing.
make the test cases pass.
put the screen shot.

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using Mathematics;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Mathematics.Tests
{
   [TestClass()]
   public class AlgebraTests
   {
      [TestMethod()]
      public void Factorial_zero_Test()
      {
         //Arrange
         int n = 0;
         int expected = 1;

         //Act
         int actual = Algebra.Factorial(n);

         //Assert
         Assert.AreEqual(expected, actual);
      }
      [TestMethod()]
      public void Factorial_0_to_7_Test()
      {
         //Arrange
         int n = 6;
```

```
        int expected = 720;

        //Act
        int actual = Algebra.Factorial(n);

        //Assert
        Assert.AreEqual(expected, actual);
    }
    [TestMethod()]
    public void Factorial_negative_Test()
    {
        //Arrange
        int n = -5;
        int expected =-9999 ;

        //Act
        int actual = Algebra.Factorial(n);

        //Assert
        Assert.AreEqual(expected, actual);
    }
    [TestMethod()]
    public void Factorial_greater_than_seven_Test()
    {
        //Arrange
        int n = 9;
        int expected = -999;

        //Act
        int actual = Algebra.Factorial(n);

        //Assert
        Assert.AreEqual(expected, actual);
    }
    [TestMethod()]
    public void AddTest()
    {
```

```
        Assert.Fail();
      }
   }
}
```





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.

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

namespace Palindrome
{
```

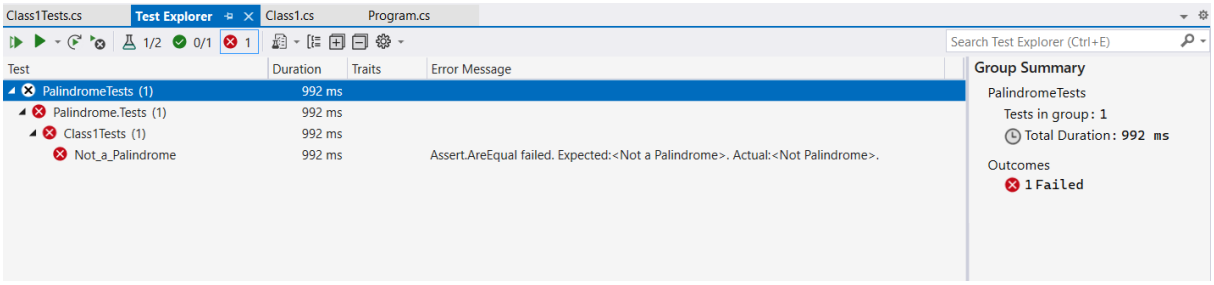
```
   public class Class1
   {
      public static string Palin(int n)
      {
         int sum = 0, rem;
         int temp = n;
         while (n > 0)
         {
            rem = n % 10;
            sum = sum* 10 + rem;
            n = n / 10;
         }
         if (temp == sum)
            return "Palindrome";
         else
            return "Not Palindrome";
      }
   }
}
```

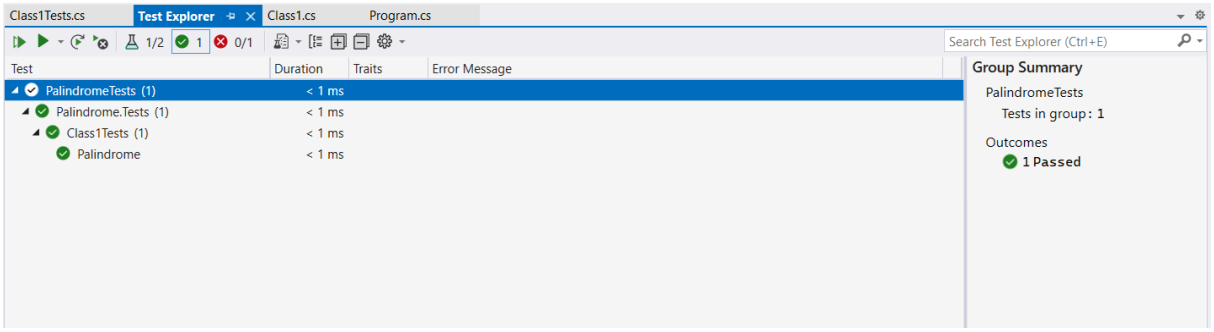
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Palindrome;

namespace Day_18_Project_2
{
   internal class Program
   {
      static void Main(string[] args)
      {
         int n;
         Console.WriteLine(Class1.Palin(12321));
         Console.ReadLine();
      }
   }
}
```

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

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


namespace Palindrome.Tests
{
   [TestClass()]
   public class Class1Tests
   {
      [TestMethod()]
      public void Palindrome()
      {
         //Arrange
         int n = 1551;
         String expected = "Palindrome";

         //Act
         string actual = Class1.Palin(n);

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

      }
      [TestMethod()]
      public void Not_a_Palindrome()

      {
         //Arrange
         int n = 5558;
         String expected = "Not a Palindrome";

         //Act
         string actual = Class1.Palin(n);

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

      }
```

```
    }
}
```


Class1Tests.cs
Test Explorer
Class1.cs
Program.cs
1/2
0/1
1
Search Test Explorer (Ctrl+E)
Test
Duration
Traits
Error Message
PalindromeTests (1) 992 ms
Palindrome.Tests (1) 992 ms
Class1Tests (1) 992 ms
Not_a_Palindrome 992 ms
Assert.AreEqual failed. Expected:<Not a Palindrome>. Actual:<Not Palindrome>.
Group Summary
PalindromeTests
Tests in group: 1
Total Duration: 992 ms
Outcomes
1 Failed


Class1Tests.cs
Test Explorer
Class1.cs
Program.cs
1/2
1
0/1
Search Test Explorer (Ctrl+E)
Test
Duration
Traits
Error Message
PalindromeTests (1) < 1 ms
Palindrome.Tests (1) < 1 ms
Class1Tests (1) < 1 ms
Palindrome < 1 ms
Group Summary
PalindromeTests
Tests in group: 1
Outcomes
1 Passed