School of Science, Computing and Engineering Technologies

Object Oriented Programming

Pass Task 2.4: Case Study — Iteration 1: Identifiable Object

Overview

Object-oriented programming makes best sense with larger programs. The case study will be your opportunity to create a larger program and better understand how the object-oriented approach can make it easier to create complex software solutions.

Purpose:	Practice interpreting UML class diagrams and writing unit tests.
Task:	Understand the case study program and implement iteration 1.
	The task contains personalized requirements.
Deadline:	Due by the end of week three, **Fri, 23 May 2025, 23:59 Hanoi Time (Firmed)**.

Submission Details

All students have access to the Adobe Acrobat tools. Please print your solution to PDF and combine it with the screenshots taken for this task.

- Program source code
- Test source code
- Screenshot of unit tests passing

SWINBURNE UNIVERSITY OF TECHNOLOGY

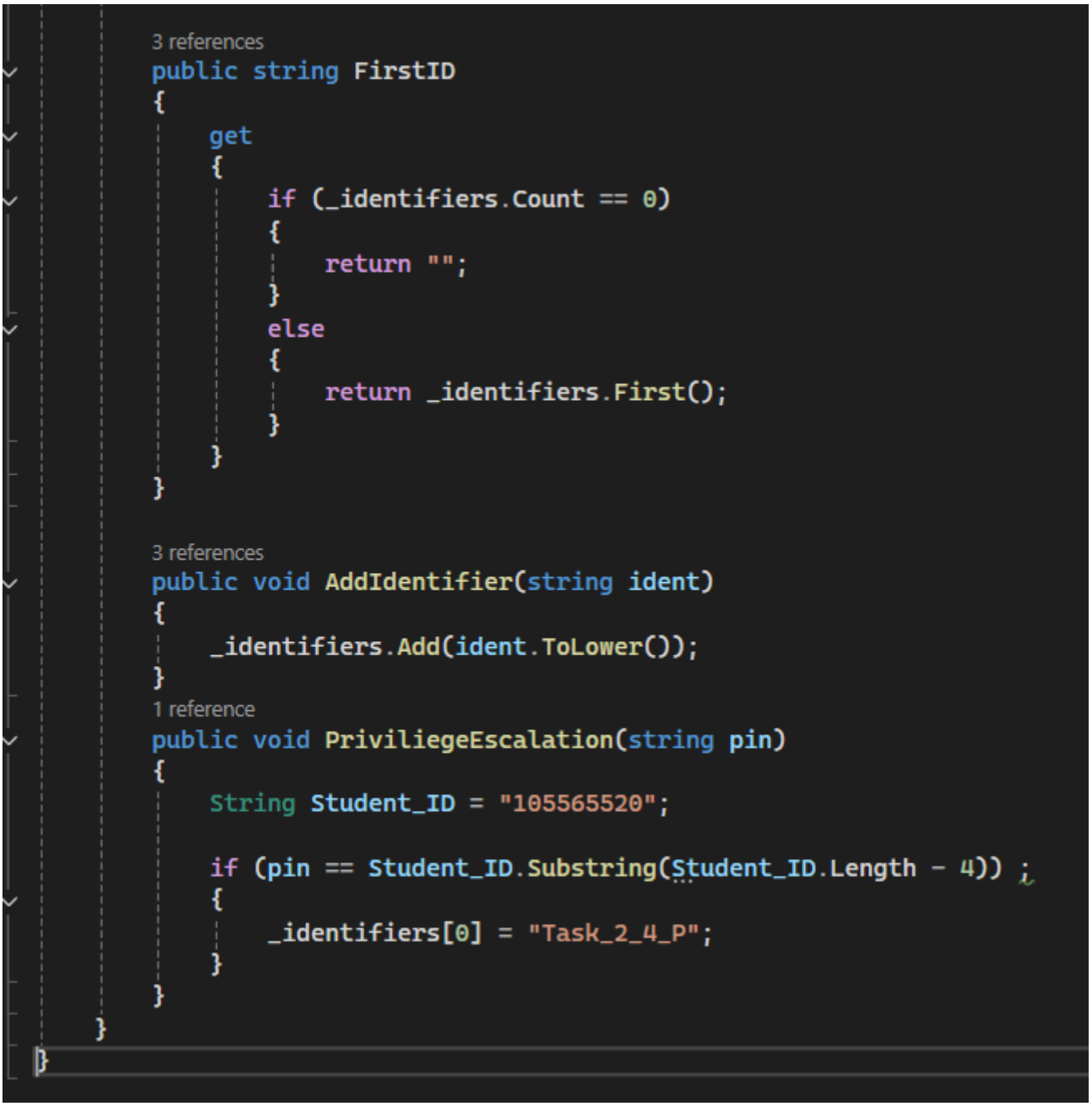
Instructions

1. Review the *Case Study Requirements* document and implementation plan included in the task resources. It outlines what you need to create.
2. For this week aim to complete Iteration 1.

> **Note**: At this point there will not be a "program" as such, just a set of unit tests that help demonstrate that your solution is moving towards completion.

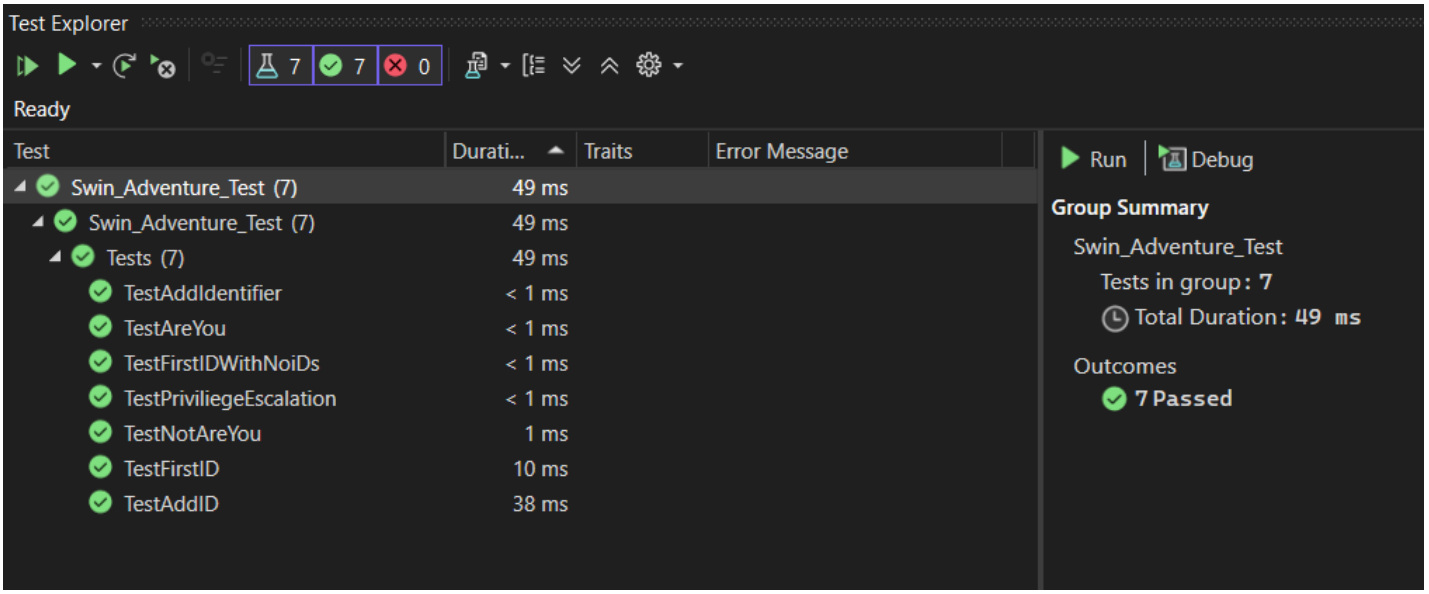
Once your tests are working correctly get a screenshot of the tests passing and submit them along with the code.

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

namespace Swin_Adveture_Task
{
    5 references
    public class IdentifiableObject
    {
        public List<string> _identifiers = new List<string>();

        2 references
        public IdentifiableObject(string[] idents)
        {
            foreach (string ident in idents)
            {
                AddIdentifier(ident);
            }
        }

        5 references
        public bool AreYou(string ident)
        {
            foreach (string identifier in _identifiers)
            {
                if (ident.ToLower() == identifier)
                {
                    return true;
                }
            }
            return false;
        }
```


3 references
public string FirstID
{
get
{
if (_identifiers.Count == 0)
{
return "";
}
else
{
return _identifiers.First();
}
}
}
3 references
public void AddIdentifier(string ident)
{
_identifiers.Add(ident.ToLower());
}
1 reference
public void PriviliegeEscalation(string pin)
{
String Student_ID = "105565520";
if (pin == Student_ID.Substring(Student_ID.Length - 4)) ;
{
_identifiers[0] = "Task_2_4_P";
}
}
}
}

```
using Microsoft.VisualStudio.TestPlatform.ObjectModel;
using Swin_Adveture_Task;

namespace Swin_Adventure_Test
{
    [TestFixture]
    0 references
    public class Tests
    {
        IdentifiableObject ident;
        [SetUp]
        0 references
        public void Setup()
        {
            ident = new IdentifiableObject(new string[] { "105565520", "Gia", "Hieu" });
        }
        [Test]
        0 references
        public void TestAreYou()
        {
            Assert.That(ident.AreYou("105565520"), Is.True);
            Assert.That(ident.AreYou("Gia"), Is.True);
        }
        [Test]
        0 references
        public void TestNotAreYou()
        {
            Assert.That(ident.AreYou("unknown_id"), Is.False);
        }
        [Test]
        0 references
        public void TestFirstID()
        {
            Assert.That(ident.FirstID, Is.EqualTo("105565520"));
        }
        [Test]
        0 references
        public void TestAddIdentifier()
        {
            ident.AddIdentifier("NewID");
            Assert.That(ident.AreYou("NewID"), Is.True);
        }
```

```
        public void TestAddIdentifier()
        {
            ident.AddIdentifier("NewID");
            Assert.That(ident.AreYou("NewID"), Is.True);
        }
        [Test]
        0 references
        public void TestFirstIDWithNoiDs()
        {
            IdentifiableObject emptyIdent = new IdentifiableObject(new string[] { });
            Assert.That(emptyIdent.FirstID, Is.EqualTo(""));
        }
        [Test]
        0 references
        public void TestAddID()
        {
            ident.AddIdentifier("James");
            Assert.That(ident.AreYou("James"), Is.True);
        }
        [Test]
        0 references
        public void TestPriviliegeEscalation()
        {
            ident.PriviliegeEscalation("6520");
            Assert.That(ident.FirstID, Is.EqualTo("Task_2_4_P"));
        }
    }
}
```



Assessment Criteria

Make sure that your task has the following in your submission:

- The “Universal Task Requirements” (see Canvas) have been met.