

SWINBURNE UNIVERSITY OF TECHNOLOGY

COS20007 OBJECT ORIENTED PROGRAMMING

Case Study Iteration 1 - Identifiable Object

PDF generated at 21:38 on Tuesday 3rd October, 2023

```
1  using System;
2  /*
3   * File: NunitTemplate.cs
4   * Unit: COS20007 Object Oriented Programming
5   * Institution: Swinburne University of Technology
6   */
7
8  namespace TestIdentifiableObject
9  {
10    public class IdentifiableObject
11    {
12      private List<string> _identifiers = new List<string>();
13
14      public IdentifiableObject(string[] idents) //constructor
15      {
16        foreach (string s in idents)
17        {
18          _identifiers.Add(s.ToLower()); //ToLower to match AddIdentifier
19          definition
20        }
21      }
22
23      public string FirstID
24      {
25        get
26        {
27          if (_identifiers.Count == 0)
28          {
29            return "";
30          }
31          else
32          {
33            return _identifiers[0];
34          }
35        }
36
37      public bool AreYou(string id)
38      {
39        return _identifiers.Contains(id.ToLower()); //ToLower due to test
40        definition
41      }
42
43      public void AddIdentifier(string id)
44      {
45        _identifiers.Add(id.ToLower());
46      }
47    }
48 }
```

```
1  /*
2   * File: NunitTemplate.cs
3   * Unit: COS20007 Object Oriented Programming
4   * Institution: Swinburne University of Technology
5   */
6
7  using System;
8  using System.Collections.Generic;
9  using NUnit.Framework; //Don't forget this.
10 using TestIdentifiableObject; //Rename this to the namespace of your project (project
11    ↳ name).
12
13 namespace NUnitTests //This should match your NUnit test project name.
14 {
15     [TestFixture]
16     public class TestIdentifiableObject
17     {
18         private IdentifiableObject _testObject;
19
20         [SetUp]
21         public void SetUp()
22         {
23             _testObject = new IdentifiableObject(new string[] { "fred", "bob" });
24         }
25
26         [Test]
27         public void TestAreYou()
28         {
29             Assert.IsTrue(_testObject.AreYou("fred"));
30             Assert.IsTrue(_testObject.AreYou("bob"));
31         }
32
33         [Test]
34         public void TestNotAreYou()
35         {
36             Assert.IsFalse(_testObject.AreYou("wilma"));
37             Assert.IsFalse(_testObject.AreYou("boby"));
38         }
39
40         [Test]
41         public void TestCaseSensitive()
42         {
43             Assert.IsTrue(_testObject.AreYou("fred"));
44             Assert.IsTrue(_testObject.AreYou("bob"));
45         }
46
47         [Test]
48         public void TestFirstID()
49         {
50             Assert.That(_testObject.FirstID, Is.SameAs("fred"));
51         }
52
53         [Test]
```

```
53     public void TestFirstIDWithNoIDs()
54     {
55         IdentifiableObject emptyObject = new IdentifiableObject(new string[] {}
56         ); // Creating an object with no identifiers
57         Assert.That(emptyObject.FirstID, Is.SameAs(""));
58     }
59
60     [Test]
61     public void TestAddID()
62     {
63         _testObject.AddIdentifier("wilma");
64         Assert.IsTrue(_testObject.AreYou("fred"));
65         Assert.IsTrue(_testObject.AreYou("bob"));
66         Assert.IsTrue(_testObject.AreYou("wilma"));
67     }
68 }
```

