

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: CDS20007 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             }
20         }
21
22         public string FirstID
23         {
24             get
25             {
26                 if (_identifiers.Count == 0)
27                 {
28                     return "";
29                 }
30                 else
31                 {
32                     return _identifiers[0];
33                 }
34             }
35         }
36
37         public bool AreYou(string id)
38         {
39             return _identifiers.Contains(id.ToLower()); //ToLower due to test
40         }
41
42         public void AddIdentifier(string id)
43         {
44             _identifiers.Add(id.ToLower());
45         }
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
   ↳ name).
11
12 namespace NUnitTests //This should match your NUnit test project name.
13 {
14     [TestFixture]
15     public class TestIdentifiableObject
16     {
17         private IdentifiableObject _testObject;
18
19         [SetUp]
20         public void SetUp()
21         {
22             _testObject = new IdentifiableObject(new string[] { "fred", "bob" });
23         }
24
25         [Test]
26         public void TestAreYou()
27         {
28             Assert.IsTrue(_testObject.AreYou("fred"));
29             Assert.IsTrue(_testObject.AreYou("bob"));
30         }
31
32         [Test]
33         public void TestNotAreYou()
34         {
35             Assert.IsFalse(_testObject.AreYou("wilma"));
36             Assert.IsFalse(_testObject.AreYou("boby"));
37         }
38
39         [Test]
40         public void TestCaseSensitive()
41         {
42             Assert.IsTrue(_testObject.AreYou("fred"));
43             Assert.IsTrue(_testObject.AreYou("bob"));
44         }
45
46         [Test]
47         public void TestFirstID()
48         {
49             Assert.That(_testObject.FirstID, Is.SameAs("fred"));
50         }
51
52         [Test]
```

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

