Assertions: Same and True

Assertions

- Test for Same and NotSame

Method name	Description
`assertSame`	Assert that items refer to same object
`assertNotSame`	Assert that items do not refer to same object

Code to Test

DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

  private String academy = "Luv2Code Academy";
  private String academyDuplicate = academy;

  public String getAcademy() {
      return academy;
  }

  public String getAcademyDuplicate() {
      return academyDuplicate;
  }
}
```

Check if these refer to the same object

DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

   private String academy = "Luv2Code Academy";
   private String academyDuplicate = academy;


   public String getAcademy() {
       return academy;
   }

   public String getAcademyDuplicate() {
       return academyDuplicate;
   }
}
```

DemoUtilsTest.java

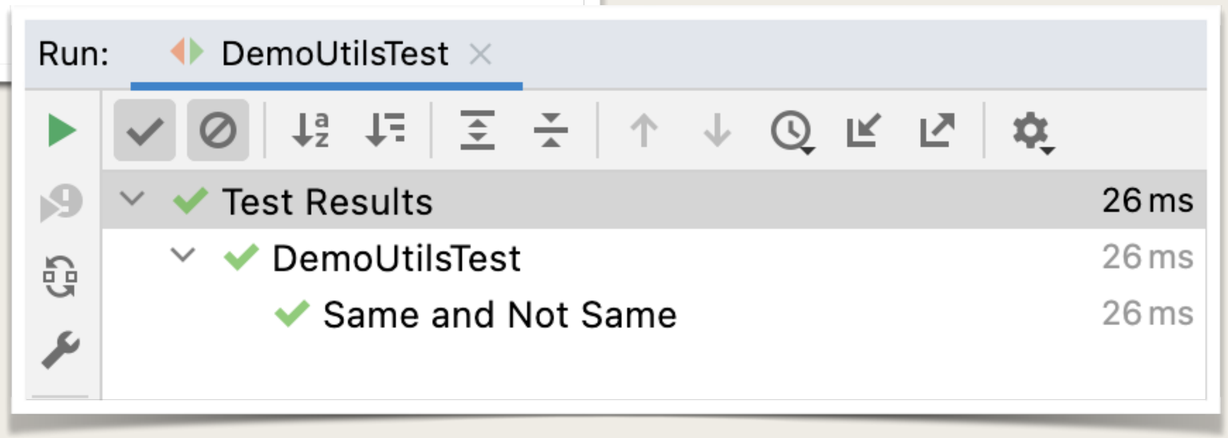
```
package com.luv2code.junitdemo;

import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;

class DemoUtilsTest {

    DemoUtils demoUtils;

    ...

    @DisplayName("Same and Not Same")
    @Test
    void testSameAndNotSame() {

        String str = "luv2code";

        assertSame(demoUtils.getAcademy(), demoUtils.getAcademyDuplicate(), "Objects should refer to same object");
        assertNotSame(str, demoUtils.getAcademy(), "Objects should not refer to same object");
    }

}
```

Object1

Object2

Object1

Object2



Assertions

- Test for True and False

Method name	Description
`assertTrue`	Assert that condition is true
`assertFalse`	Assert that condition is false

Code to Test

DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

  public Boolean isGreater(int n1, int n2) {
      if (n1 > n2) {
          return true;
      }
      return false;
  }
}
```

For academic purposes only :-)
We can refactor accordingly

Code to Test

DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

  public Boolean isGreater(int n1, int n2) {
      return n1 > n2;
  }
}
```

Refactored version

DemoUtilsTest.java

```
package com.luv2code.junitdemo;

import org.junit.jupiter.api.*;
import static org.junit.jupiter.api.Assertions.*;

class DemoUtilsTest {

    DemoUtils demoUtils;

    ...

    @DisplayName("True and False")
    @Test
    void testTrueFalse() {
        int gradeOne = 10;
        int gradeTwo = 5;

        assertTrue(demoUtils.isGreater(gradeOne, gradeTwo), "This should return true");
        assertFalse(demoUtils.isGreater(gradeTwo, gradeOne), "This should return true");
    }
}
```

Boolean condition

Boolean condition

DemoUtils.java

```
package com.luv2code.junitdemo;

public class DemoUtils {

  public Boolean isGreater(int n1, int n2) {
      return n1 > n2;
  }
}
```