

COMP 587
Spring 2020

Tests as Specifications

Consider the unit tests below. Write code to get the tests to pass.

```
import static org.junit.Assert.assertEquals;
import static org.junit.Assert.assertTrue;
import static org.junit.Assert.assertFalse;
import org.junit.Test;
public class TestThings {
    @Test
    public void testAddOperation() {
        final Operation<Integer, Integer> op = new AddOp(5);
        assertEquals(8,
                     op.doOperation(Integer.valueOf(3)).intValue());
    }

    @Test
    public void testToStringOperation() {
        final Operation<Integer, String> op =
            new ToStringOp<Integer>();
        assertEquals("7", op.doOperation(Integer.valueOf(7)));
    }

    @Test
    public void testContainerGetItem() {
        final Integer value = Integer.valueOf(8);
        final Container<Integer> c = new Container<Integer>(value);
        assertEquals(value, c.getItem());
    }

    @Test
    public void testContainerToString() {
        final Container<Integer> c =
            new Container<Integer>(Integer.valueOf(42));
        assertEquals("42", c.toString());
    }

    @Test
    public void testContainerHashCode() {
        final Boolean b = Boolean.valueOf(true);
        final Container<Boolean> c = new Container<Boolean>(b);
        assertEquals(b.hashCode(), c.hashCode());
    }
}
```

```

@Test
public void testContainerEqualsTrue() {
    final Container<Integer> c1 =
        new Container<Integer>(Integer.valueOf(77));
    final Container<Integer> c2 =
        new Container<Integer>(Integer.valueOf(77));
    assertTrue(c1.equals(c2));
    assertTrue(c2.equals(c1));
}

@Test
public void testContainerEqualsFalse() {
    final Container<String> c1 = new Container<String>("hi");
    final Container<String> c2 = new Container<String>("bye");
    assertFalse(c1.equals(c2));
    assertFalse(c2.equals(c1));
}

@Test
public void testConvertIntegerString() {
    final Operation<Integer, String> op =
        new ToStringOp<Integer>();
    final Container<Integer> c =
        new Container<Integer>(Integer.valueOf(777));
    assertEquals(new Container<String>("777"),
        c.convert(op));
}

@Test
public void testConvertIntegerInteger() {
    final Operation<Integer, Integer> op = new AddOp(3);
    final Container<Integer> c =
        new Container<Integer>(Integer.valueOf(123));
    assertEquals(new Container<Integer>(Integer.valueOf(126)),
        c.convert(op));
}

@Test
public void testConvertStringString() {
    final Operation<String, String> op =
        new ToStringOp<String>();
    final Container<String> c = new Container<String>("foobar");
    assertEquals(c, c.convert(op));
}
}

```

