CE00527-5 Further Object Oriented Programming
Tutorial Week 5

In this tutorial you will

- Practice using Exceptions and Assertions
- Learn to use JUnit testing

Part 1 – Exceptions and Assertions –Basic Exercises

1. Write a class containing two static methods with signatures

```
public static int inputInt()
public static double inputDouble()
```

which ask the user for input of a given type and return the data entered. The methods should not crash if the user inputs data of the wrong type – instead, they should catch the exception thrown, output an appropriate message, and prompt the user to try again. Add a main method to the class which tests these methods.

Hint: You can clear invalid input from a Scanner object using the Scanner next() method.

Put in your portfolio the code listing for your class, and examples of output from testing the methods.

2 Experiment with using assertions to test your code is valid. Find a suitable method in any of the classes you have made already (or make a new one) and put in an assertion. Test the method under conditions when the assertion evaluates to true, and others where it is false.

Put in your portfolio the code listing of your method which includes an assertion, and an example of the output generated when the assertion is false.

Part 2 – Using JUnit for testing –Basic Exercise

4 Create a new Netbeans project and put into it a Person class with only two attributes – firstName and lastName of type String. Person should have an appropriate constructor, get- and set- methods for each attribute, and a toString() method which returns the Person's full name (first name and last name with a space in between).

Create a PersonTest class with JUnit tests for each method and use them to test your class.

Put the code listing for the Person class, the PersonTest class, and a screen shot of the resulting JUnit Test Results window (with all the tests green!) in your portfolio

5 In the same project as question 4, develop a DVD class which stores information about DVD. This class should have:

- a String attribute title
- an attribute leadActor of type Person
- an int attribute noOfStars (number of stars)
- a constructor with appropriate parameters to initialise all the attributes
- getter methods for each attribute

- A toString method, which returns the DVD title, lead actor's name, and number of stars as a suitably formatted String.

Use JUnit to test your DVD class. You will need to write a class DVDTest. Declare a DVD object, d1, as an attribute of DVDTest. In the setup() method, construct d1 with the following attributes:

- Title: Inception
- Lead Actor: A Person with first name "Leonardo", last name "DiCaprio"
- Number of stars: 5

Write methods to test each method of d1. Test the DVD getLeadActor() method by checking that the lead actor's name is as expected, using the following test method

```
@Test
public void testGetLeadActor() {
    Person leadActor = d1.getLeadActor();
    assertEquals("Leonardo DiCaprio", leadActor.toString());
}
```

Ensure that all your tests pass.

Put the code listing for the DVD class, the DVDTest class, and a screen shot of the resulting JUnit output (with all the tests green!) in your portfolio.