SYSC 2004 C and D: Object-Oriented Software Development **Programming Assignment No 4** [3 Marks] (Due date: **Mar. 13, 2020**)

The main objective of this programming assignment No 4 is to learn about test automation and how to use effectively the unit testing, regression tests, and to learn and experiment with Automated testing using BlueJ integrated JUnit (designing test cases, creating test classes, defining test methods, specifying assertions, creating fixtures, recording a test, running tests, and analyzing test results). To this end we will be reusing the *online-shop-junit* project from Chapter 9. Source code for this project is available in cuLearn in the general block (Chapter 9).

Q1: Create a test that performs *negative testing* on the boundaries of the rating range. That is, test the values 0 and 6 as a rating (the values just outside the legal range from 1 to 5). We expect these to return **false**, so assert **false** in the result dialog. You will notice that one of these actually (incorrectly) returns **true**. This is a bug. Make sure that you assert **false** anyway. The assertion states the *expected* result, not the *actual* result.

Q2: Run all tests again. Explore how the Test Result dialog displays the failed test. Select the failed test in the list. What options do you have available to explore the details of the failed test?

Q3: Create a test class that has **Comment** as its reference class. Create a test that checks whether the author and rating details are stored correctly after creation. Record separate tests that check whether the **upvote** and **downvote** methods work as expected.

Q4: Create tests for **SalesItem** that test whether the **findMostHelpfulComment** method works as expected. Note that this method returns a **Comment** object. During your testing, you can use the Get button in the method result dialog to get the result object onto the object bench, which then allows you to make further method calls and add assertions for this object. This allows you to identify the **Comment** object returned (e.g., by checking its author). You can also assert that the result is *null* or *not null,* depending on what you expect.

Q5: Add further automated tests to your project until you reach a point where you are reasonably confident of the correct operation of the classes. Use both positive and negative tests. If you discover any errors, be sure to record tests that guard against recurrence of these errors in later versions.

Important Notes and Submission Guidelines:

You start by reading the source code *online-shop-junit* project and identifying the classes and their objects and methods involved for every test of functionality, then for every test you need to establish a *test procedure* and follow its steps. You might decide whether you need to use and reuse *fixtures* or not.

For this programming assignment No4, you have to submit a ZIP file that contains your BlueJ project source code. In addition, you also have to book and pass a demo using BleueJ with your teaching assistant (TA) in the lab room or during her/his office hours. Your TA will lead you and conduct and run the tests of your programs and evaluate your work. You are also allowed to do your demo using your own laptop.