IFN501 Assignment 2

Rules

Assignment 2 is a programming exercise where you are required to develop and submit a C# program, covering one of the two following alternative scenarios:

- Scenario 1 is an extended version of the "product inventory" scenario from assignment 1's exercise 2 and 3.
- Scenario 2 is a self-proposed use-case, where you can define and propose the scope of the program yourself.

Working in pairs

Assignment 2 can be worked on individually or in pairs. If you decide to work on the assignment as a pair, you are yourself responsible for finding a fellow co-student to partner with. You will then also need to manage the "work relationship" with your partner yourself. If a pair of two students first starts to work on the assignment together and later parts ways, both students are required to continue to work on the assignment as if they would have been working individually from the onset (and also submit two separate assignments on Blackboard).

Submission

Assignment 2 is due before the end of week 13 (May 29, 2015) and must be submitted on QUT Blackboard. The program you have produced must be submitted in the form of a zipped (compressed) Visual Studio solution folder, alongside with instructions how to run the program and optionally further notes which you would like to have your tutor consider when marking the assignment.

Marking

Assignment 2 contributes 40% of the overall mark for IFN501. Different to assignment 1, graded marks will be assigned, based on the following criteria:

- Functional correctness (the program fully and correctly covers the prescribed functionality)
- Code quality (the program correctly uses the prescribed programming concepts)

Both in scenario 1 and 2, implementing one or all of the stated optional features may compensate for missed marks from the aforementioned assessment of the mandatory features. However, marks earned from implementing from optional features of assignment 2 cannot be used to "top up" missed marks from assignment 1.

Variations

You can apply for variations to both, the pre-defined scenario 1 or the self-proposed scenario 2 (e.g., use Web forms instead of Windows forms). These variations must be approved in writing by your tutor or lecturer. If you have successfully applied for

a variation, you need to attach the confirmation (e.g., an email from your tutor) to the materials that are finally submitted on Blackboard. When applied for scenario 1, a variation is meant to offer some flexibility for individual creativity, but not to reduce the workload. You can also propose additional optional tasks. With scenario 2, a variation can also be sought if some elements of your self-proposed (and approved) scenario later turn out to be unrealistic.

Plagiarism

QUT's general rules concerning plagiarism apply: the program code that is submitted must be your own. In particular, you must not share any code with other IFN501 students. We will check for duplications among submissions.

Scenario 1

In scenario 1, we ask you to program an extended version of the "product inventory" use-case from assignment 1. Specifically, you need to implement a C# program that manages a number of product catalogues, where each catalogue contains products from a single category, e.g., electronics, apparel, food, beverages, etc.

Mandatory Features

The mandatory features that are to be implemented are:

- Adding a new product catalogue, which is initially empty
- Removing a product catalogue
- Adding a product to a product catalogue
- Removing a product from a product catalogue
- Listing the products in a product catalogue
- Calculating the median price for all products in a single product catalogue
- Calculating the median price for all products across all product catalogues

Similarly to assignment 1, you are supposed to use classes for products and product catalogues (and other concepts your choice, where applicable). Enhancing the assignment 1 scenario, a number of changes apply:

- Use a C# enumeration for the available product categories (you can hard-code your custom set of product categories)
- Use suitable collection classes (instead of arrays) to store the product catalogues and the product catalogues
- Use a GUI (Windows forms) for all user interactions (no console input/output).

Optional Features

In addition to the mandatory features, a number of optional features can be implemented to compensate for missed marks from the mandatory features:

- Sort products (in a product catalogue) by name or price
- Store the product inventory (i.e., all product catalogues and the contained products) in a file and load (recover) the product inventory from a file

Scenario 2

Alternatively to scenario 1, scenario 2 gives you the means to propose your own use-case. The use-case should be comparable to scenario 1 in that it covers the same underlying programming concepts and requires similar effort to complete. Similarly to scenario 1, it is highly recommended that you devise a number of optional features, which are taken into account when there are any shortcomings with the mandatory features.

Approval

If you choose to propose your own scenario, you will need to gain written approval from your tutor. He/she will help you making sure that the aforementioned quality criteria are satisfied while the scope remains manageable in the remaining time frame. Submit your idea as soon as possible and allow for a few days to discuss and refine your use-case with your tutor. You need to have gained approval for your scenario by May 13th. Only once your tutor has approved your idea, you can rely on it being accepted for marking.