

**Object Oriented Design Patterns**  
**SE-OODP Continuous Assessment**

# **VMCS with a new perspective**

© 2007-2016 NUS. The contents contained in this document may not be reproduced in any form or by any means, without the written permission of ISS, NUS other than for the purpose for which it has been supplied.

**Please read carefully this document  
and make sure you comply to the instructions herewith.  
If you have any questions, ask them in the course forum.**

## ASSIGNMENT

In this assignment, you are required to redesign the Vending Machine Control System (VMCS) Computer Simulation system that was provided to you in Unit 3.

The new design should look into various aspects of a good design such as maintainability, extensibility, reusability, etc. and make your system resilient to change as much as possible. In order to achieve that, you will need to make use of the various design patterns taught in the class including the Gang of Four and other design patterns.

## OVERVIEW OF SYSTEM REQUIREMENTS

The **Vimto Soft Drinks Company** operates and maintains a large number of automatic soft drinks dispensers at various sites throughout Singapore. Previously these machines have been controlled by analog devices, which are now out of production and also difficult to maintain. It has been decided to invest in a computerized control system that will be installed in a new range of soft drinks vending machines - the **Vimto 45** drinks dispensing machine. These machines will sell several different brands of drinks.

The following gives an overview of the system functions:

1. Acceptance/rejection and cumulative totaling of coins entered by customers.
2. Dispensing of a drinks can of a selected brand to the customer.
3. Indication of whether the brands are available (or whether they are *out-of stock*).
4. Dispensing of change to customers.
5. Displaying of requested information to the vending machine maintainer.
6. Setting of new prices for drinks brands by the vending machine maintainer.
7. Issue of all the coins held within machine to the maintainer.
8. Simulated user interfaces for the customer and maintainer, and an additional user interface to allow the state of the vending machine to be monitored and for the stocks levels of coins and drinks cans held to be changed.

Refer to the User Requirements Specification (URS) document [ISS/VMCS/TR.1/V2.0] for detailed functional requirements.

Also refer to the Use-Case-Model Survey for VMCS, Issue 2.0 [ISS/VMCS/TR.2/1] for the description of the requirements in terms of Use Cases. This document provides an overview of these use cases and their Actors (users).

## General Requirements

### Design of the system

Refer to the VMCS CASE STUDY – Design Model Report document, Issue 4.0 [ISS/VMCS/TR.4/1] for an initial design model of the system. This includes the sequence diagrams and the object specifications for the VMCS system.

### Teams

You will be working in teams of 5~6 members. Use the IVLE and assign yourselves to teams before 24 Sep 2016.

### Assignment

You are required to do the following:

- Study the design presented in the above document
- Identify design problems (at least one per team member) from the point of view of a good design
  - The design problems could be to improve the current design for extensibility, reusability, maintainability or simply elegance
  - After considering problems in the current design, if required, you may consider design problems due to incorporation of any additional relevant extensions to the current set of requirements.
- For each design problem,
  - Identify candidate design patterns that you think would solve the design problems. Please provide a discussion about why you think they could solve the problem.
  - Select appropriate design pattern(s). Please provide elaborate justifications for your choice.
  - Redesign the use case(s) to illustrate the workings and benefit of incorporating the design pattern(s)
    - Provide the modified use case(s) design including sequence diagram(s) and class diagram(s) as appropriate
    - Discuss how this pattern solves the problem
    - Discuss and justify the relevant implementation issues and your choices [See Implementation Section of the design pattern]
  - Implement a use case(s) that involves the above mentioned design pattern(s)
    - You will need to provide sufficient code for the use case involved in illustrating this design pattern(s)
    - Sufficient and useful data to illustrate the workings of the use case must also be provided

**Note: You do not need to provide the code for the whole system.**

## PRILIMINARY PRESENTATION

The first presentations will be on **3 Oct 2016, Mon (Full-time), 1 Oct 2016, Sat (Part-time)**.

Every team will be given a total of **20 minutes to present including Q&A** (15 minutes of presentation and 5 minutes for Q&A). You can present using PowerPoint slides.

It is **compulsory** that you be present for all the presentations of your class and not just your team presentation. This will further facilitate your learning.

The presentation will normally be according to the ascending order of the team numbers. Please ensure that you are ready to present when your team is called. The exact schedule will be provided nearer to the date of presentation.

You will need to pick any **one problem** and discuss the following:

- Brief description of the design problem
- Candidate design patterns considered
- Motivation to choose a pattern that would solve the problem
- Structure of the pattern (you should map the participants to your applications classes/objects),
- Collaborations among the participants (specific to your application objects),
- Implementation decisions that you have taken.

## END OF PROJECT PRESENTATION AND DEMO

The final presentations will be on **26 Oct 2016, Wed (Full-time, day-time) and 25 & 26 Oct 2016, Tue & Wed (Part-time, evening)**. Every team will be given a total of **35 minutes to present and show a demo of your work, including Q&A** (15 minutes of presentation, 10 minutes of demo and 10 minutes for Q&A).

It is **compulsory** that you be present for all the presentations of your class and not just your team presentation. This will further facilitate your learning.

The presentation will normally be according to the ascending order of the team numbers. Please ensure that you are ready to present when your team is called. The exact schedule will be provided nearer to the date of presentation.

Your **presentation** should include the following:

- For any **one problem** that has not been presented in the preliminary presentation and discuss the following:
  - Brief description of the design problem
  - Candidate design patterns considered
  - Motivation to choose a pattern that would solve the problem
  - Structure of the pattern (you should map the participants to your applications classes/objects),
  - Collaborations among the participants (specific to your application objects),
  - Implementation decisions that you have taken.

Show a **demo** of one use case that is involved in the design problem presented.

## END OF PROJECT CROSS EVALUATION

This evaluation component is introduced to assess the ability of students in evaluating other's work. This will be conducted during the final presentation; every team will be cross-evaluated by another team on the **one** design problem presented. For that matter, additional 5 minutes are catered for each presentation.

One “evaluating” team is assigned to evaluate a “presenting” team. The evaluating team critiques the work of the presenting team. The aim is to assess the evaluating team's ability in understanding other's work and providing critiques objectively.

The evaluating team should evaluate the following aspects of the **one** design problem of the producer team via **Q&A during the presentation**:

- Motivation to choose a pattern that would solve the problem,
- Structure of the pattern (participants are mapped to classes/objects in the context of problem),
- Collaborations among the participants (specific to the context of problem),
- Implementation decisions that were taken.

The evaluating team shall be assessed by the lecturers based on the following criteria:

- Ability to evaluate the validity of the design problem,
- Ability to evaluate the appropriateness of a pattern for the design problem,
- Ability to evaluate the compliance in applying the pattern in the design solution,
- Ability to evaluate the rationale in the implementation decisions taken.

## END OF PROJECT DELIVERABLES

You will upload the following deliverables to IVLE by the final presentation day (see above):

- source code
- data files
- executable program and the necessary instructions to run the program
- design document
  - For each design problem, you are required to discuss at least the following:
    - Description of the design problem
    - Candidate design patterns considered
    - Motivation to choose a pattern that would solve the problem including support for new requirements or changes to existing requirements
    - Structure of the pattern (you should map the participants to your applications classes/objects),
    - Collaborations among the participants (specific to your application objects),
    - Implementation decisions that you have taken.
  - Modified design document including modified sequence diagrams, class diagrams, object specifications, state charts etc. for the relevant use cases.
  - Individual team member contributions (including design and implementation)

Note:

- Please ensure that your application can be compiled without any errors and that it can be executed successfully. Furthermore, you must provide sufficient documentation for compiling your application, configuring your environment and executing your application including any batch files where relevant. There will be a penalty if we are unable to compile or execute your application successfully.
- Continuous assessment is an integral part of this elective. You will disqualify this elective if you fail to submit the project or if you fail to make a reasonable attempt at the project.
- Each member of the team is required to do the design and implementation (coding, testing) of *at least one problem*.

The final deliverables must be packed into a ZIP file named `OODP_CA_Bxx.zip` (where B is your batch: PT – Part-time, FT – Full-time; xx is your team number, e.g. 01, 02, etc...). The ZIP file must be uploaded onto IVLE in the Course (SG 5208 Object Oriented Design Patterns) Workbin in the “StudentSubmission” folder by the deadline.

**The application should be developed using JDK1.6.x. *No other language or platform is allowed.***

## PEER ASSESSMENT

Each team member is expected to rate his/her peer's contribution to the project. In order to ensure that equal marks are awarded to all team members, each team member is expected to contribute equally to the CA.

The instructions to submit Peer Assessment form shall be provided in due time via IVLE.

The deadline for submission of the Peer Assessment form is on the final presentation day (see above).