

# PRACTICE PRESENTATIONS FEEDBACK

CEWP MOD4B | OBJECT-ORIENTED ANALYSIS AND DESIGN WITH UML

CONCORDIA UNIVERSITY | CENTRE FOR CONTINUING EDUCATION

JARED CHEVALIER

## SUMMARY

### GENERAL

- No need to explain basic theory we've all learned in class
- There should be equal participation from all team members
- Time: 15-20 minutes (15 minutes is the goal, 20 minutes is the maximum)
- Don't forget to present with energy and enthusiasm!

### USE CASE DIAGRAM

- Use cases should be active verb phrases, not simple nouns or verbs
- Use cases should be more high level, from user's point of view, rather than low-level fine-grained actions
- A use case is a way of using the system to achieve one of the user's goals (which may involve a workflow with multiple steps and alternative paths)
- Goal: Know what products are available in store (Use case: Browse products)
- Goal: Decide what products you want to buy (Use case: Shop with cart)
- Goal: Purchase products and arrange delivery (Use case: Place order)
- Goal: Know details about orders you made in the past (Use case: View orders)

### CLASS DIAGRAM

- Include relationships if you actually need to record that a relationship exists between objects in your system, not simply if there is any connection between the real-life entities
- For example, include a "places" relationship between Customer and Order classes, because you will need to record the fact that customer 1 placed orders 1, 2, and 3 (whereas other customers placed orders 4, 5, and 6) so you can access that information later (when a customer views their orders, you should only show them their own orders, not everyone's orders, so you need to record who placed which orders)
- If real-life customer browses products using system, but you don't need to record the fact that customer 1 viewed product 1, 2, or 3, or record information about that fact (when they viewed the product, whether they added to cart, etc.), then don't include the relationship
- Another example: If you want to record the fact that a guest is using system, for business analytics purposes, then include Guest class, but if you don't need to record that information to represent in your system, don't include it in your model
- Make your model as simple as possible, and as complex as required

## SOFTWARE DEVELOPMENT LIFECYCLE

- Remember, there is a distinction between the Planning phase in the software development lifecycle and “planning” as in task/issue tracking with project management software like Trello or Jira
- Planning phase includes tasks such as: Assign project manager, estimate project cost, establish project procedures, define scope of project, define project schedule, cost estimation, budget plan, schedule milestones and reviews, etc.
- Project management with tools like Trello and Jira is involved throughout the entire software development lifecycle (especially in design + implementation + testing), not just in the planning phase

## ENCAPSULATION

- Use private variables
- Remember, in Java the default access modifier is package-private, not private, so you need to explicitly add the `private` keyword
- Don't use public or package-private variables, unless they are immutable constants (static final fields with a strictly immutable data type like a primitive or String) or unless you have an excellent reason for doing so
- Methods should be private unless you need to call them from a different class, in which case they can be public

## DEMO

- Show features from user point of view
- Show success scenarios, of course
- Don't forget to also show error scenarios with error messages
- Show us key parts of your implementation, particularly parts where you solved a problem in an interesting way

## LOGIC

- Currency model: Either use BigDecimal (ideal), or double/float (not ideal)
- If using double/float, then truncate or round to two decimal places whenever updating a currency value – but keep in mind that, because of fundamental incompatibility between base-2 and base-10, you will still have floating point error sometimes, leading to more than 2 non-zero decimal digits, so currency formatting as a final step will be essential

## USER INTERFACE

- Currency formatting: Format currency to make sure exactly 2 decimal places are visible (both \$5.0 and \$5.00000001 are incorrect)