Software Engineering with Objects and Components
Practical Work: Tutorial 6

Note that you are required to do some preparation for the tutorial. Before the tutorial you need to *prepare* a draft that:

1. **Presents the Class Diagram or the Java definition of the main classes that you are developing Java code for. *Note that in order to realise sensible and non-trivial use cases you may need to implement other classes that collaborate with the main ones.***
2. **Presents the main use cases that involve those classes.**
3. **Presents an outline of the sequence (or collaboration) diagrams corresponding to those use cases.**

The aim of this tutorial is to identify implementation issues (e.g., conflicting definitions, missing information, etc.) that inhibit the collaboration between teams in order to realise high-level and complex use cases. Ideally:

1. This should be a small subset that would allow the rapid implementation of a basic system.
2. There should be agreement across the teams on the collection of selected use cases that will provide the basis for deliverable 2.

Tutorial Outcomes

By the end of this tutorial your group should have:

1. Agreed on a set of use cases from each team that will be used as the basis for the preliminary implementation of the system.
2. Agreed with each team on the classes that they should provide tests and implementations for.
3. Ensured that each team has agreed on the Java class definition they are implementing for deliverable 2 (with their tutor).
4. Begun to refine what test cases are appropriate for their classes.

After the tutorial:

1. You should review the use cases chosen to represent the preliminary implementation and attempt to identify inconsistencies.
2. You should finalise the definition of the main classes that will provide the basis for deliverable 2.
3. You should finalise the tests you need for your chosen classes.

Team Instructions

Each team brings a copy of the Class Diagram with the main Java class definitions they are implementing for their deliverable 2.

Tutorial Activities

Inspection of Diagrams and Java Classes [15 mins]

☞ *Collectively* Each team will peer-review one draft with the main Java class definitions implemented by another team for deliverable 2. Each team will have the opportunity to ask questions about the class definitions. The rest of the group will comment on the suitability of the proposed implementation. The aim of the comments should be to fix the inconsistencies in the class definitions and improve the quality of the testing of the team presenting their proposals.

✎ *Take notes of any required comments on the classes (e.g., different attributes, methods, relationships, etc.).*

♯	*identified issues, changes or comments*

A Statechart Diagram [15 mins]

☞ *Individually* ✎ Each member of the team should draw a statechart diagram for one of the classes involved in the realisation of the selected use cases.

Draw Statechart Diagram

Test Planning [20 mins]

☞ *Collectively* ✎ Each team will start to identify test cases for the identified classes. The test cases will be derived following a use case driven approach. That is, the test cases take into account different scenarios drawn from the use cases.

Test Case ID	**Scenario**	**Description**	**Data Values**	**Expected Result**	**Actual Result**
♯ 1					
♯ 2					
♯ 3					
♯ 4					
♯ 5					
♯ 6					
⋮					