

School of Science, Computing and Engineering Technologies  
Swinburne University of Technology  
**SWE30003 Software Architectures and Design**  
**Tutorial Week 6 (OO Design)**

Please carefully read the description of the well-known problem of the *Towers of Hanoi*:

*"In the jungle outside Hanoi there is a community of monks whose duty is to move rings stacked on poles according to certain rules. There are 64 rings, all of different sizes. Originally they were all stacked neatly on top of each other on a single pole: the largest ring at the bottom, and then the remaining rings in decreasing size, with the smallest ring at the top. There are three poles in the jungle and the rings are to be moved from the first pole onto the third pole. Only one ring may be moved at a time. The rings may be placed on any of the three poles, but a ring may not be placed on top of a smaller ring. It is said that when the monks are finished moving the rings so that they are all stacked in the correct order on the third pole, the world will end."*

You are given the task to come up with an *object-oriented design* of an application that takes a number of rings as argument, say  $n$ , and produces a list of moves (e.g., “move ring on top of pole 1 to pole 3”) that solves the Towers of Hanoi for  $n$  rings. To facilitate the analysis, you can initially assume that you have an *even* number of rings.

In order to do so, form groups of three to four students and address the following questions:

1. Carefully analyze the problem description and identify *candidate classes*.
2. For each of the identified candidate classes, create a CRC (Class Responsibility Collaboration) Card and briefly describe the purpose of each class at the back of the card.
3. Given the problem description, identify responsibilities and carefully assign them to the classes identified in Step 1. Discard any unnecessary classes.

*Note:* you may have to add a new class(es) responsible for setting up the poles, creating the required number of rings, and starting the move of rings. Also, the ring at the top of the initial pole may have different responsibilities than the other rings. Further, a given ring will always move in the *same* direction, that is, either left or right.

4. Given your current “solution”, perform a walk-through where each member of your small group plays the role of one (or possibly more) objects. What do you learn from this walk-through?
5. Based on the observations from Step 4, reconsider your “solution”: do you need different classes? Do you have *evenly distributed system intelligence*? Do you need to re-assign responsibilities? Can you reduce the amount of communication between various objects? You may need to iterate steps 1 to 4 a few times to get your solution “right”.

.....

6. Any questions from weekly submissions?