

OOP Examples Sheet 3

Dr Andrew Rice
Michaelmas 2019
(Material by Dr Robert Harle)

Practical exercises are listed on the course materials page on the department webpages.

Practical exercises associated with this supervision

- Alice in Wonderland

Lecture 10: Covariance and functional features (badly named Copying Objects in the notes!)

- 10.1. (W) Give an example of how covariant arrays in Java can create runtime errors.
- 10.2. (W) Compare Inner-classes, method-local classes, anonymous inner classes and lambda functions. What general advice would you give someone who is trying to choose which one to use?
- 10.3. (W) Complete the Alice in Wonderland practical task

Lecture 11: Design Patterns

- 11.1. (W) Explain the difference between the State pattern and the Strategy pattern.
- 11.2. (W) In lectures the examples for the State pattern used academic rank. Explain the problems with the first solution of using direct inheritance of `Lecturer` and `Professor` from `Academic` rather than the State pattern.
- 11.3. A drawing program has an abstract `Shape` class. Each `Shape` object supports a `draw()` method that draws the relevant shape on the screen (as per the example in lectures). There are a series of concrete subclasses of `Shape`, including `Circle` and `Rectangle`. The drawing program keeps a list of all shapes in a `List<Shape>` object.
 - (a) Should `draw()` be an abstract method?
 - (b) Write Java code for the function in the main application that draws all the shapes on each screen refresh.
 - (c) Show how to use the Composite pattern to allow sets of shapes to be grouped together and treated as a single entity.
 - (d) Which design pattern would you use if you wanted to extend the program to draw frames around some of the shapes? Show how this would work.
- 11.4. (*) Explain how Java uses the Decorator pattern with `Reader` (yes, research will be required).