# SEOC: summary and revision suggestions

Perdita Stevens

School of Informatics
University of Edinburgh

# Learning objectives

1. Design simple object-oriented systems, making appropriate use of available components;
2. Design simple software components, making sensible API decisions;
3. Evaluate and evolve object-oriented software designs, making use of common design patterns if appropriate;
4. Create, read and modify UML diagrams documenting designs;
5. Discuss the use of modelling in software development, e.g. why and how models of software can have varying degrees of formality.

# Using SEOC

- Please remember what you learned here when you develop software in future – e.g., in your project. This doesn't mean "draw UML diagrams for everything" (think about whether they're useful in the circumstances!) but it does mean "make your designs SOLID"!
- Please do well in your exams...

# SEOC exams

Same format as in past years: three questions, of which you choose two.

Aim to have a balance between:

- bookwork
- straightforward application of knowledge
- more challenging problem-solving.

Visiting undergraduates examined this semester, everyone else in the summer.

# Example exam question types

Be prepared to answer exam qustions *such as*

- You are to design a system to do .... Draw a ... diagram to illustrate ....
- Explain what ... means in this ... diagram.
- Draft an API for ...
- Write an OCL expression to ... / What does the following OCL mean... ?
- Imagine you are ... Suggest how you should use modelling to help you work.
- To make the following design decision, what information would you need and why?
- What is the ... design principle/pattern? Apply it/explain whether it is useful in this situation ...

# Notes on doing SEOC exams

Write clearly and concisely.

Many questions lend themselves to addendums like "Comment briefly on [missing or unclear requirements, design alternatives, problems in a design]" – this is your chance to show that you can think like a designer! Don't woffle. Do show understanding. (The mark scheme will typically say something like "1 mark each for any two reasonable points, e.g ...")

*Assume your papers are being read by a reasonable human being.*

- If you are confused about what a question wants you to assume, briefly say why, say what you are assuming, and do what seems best to you.
- If you find in the last minute that you've made the same mistake 6 times in a diagram, don't try to redraw it – write a note saying e.g. "All the states should have rounded corners!".

# Examinable material

Basically, everything covered in

- lectures
- required readings
- tutorial sheets
- videos, including both the ones I made and the Bloch API one

unless otherwise stated.

# Revision resources

The final tutorial is for revision, based around past exam questions.

The Forum remains open – general revision questions should go there (drop me an email too if you want input from me).

Several of the tutorial sheets had Extension sections labelled "do these if you have time or later for revision": recommended.

Past exam papers (on university past paper site): useful but NB syllabus shift. E.g. many past papers ask you to draw use case diagrams; I wouldn't.

# Next year's SEOC

As you know, this year's SEOC was a complete rewrite.

Would very much appreciate feedback on what should be changed for next year: please fill in the questionnaire.

Some questions I'd particularly like input on:

- How was the overall pace and depth of the course?
- To what extent were the tutorials useful? Were the tutorial sheets at the right level/the right length, or should they be harder/easier/longer/shorter? Would it have been better to hand in work and get it marked (but still not have it count towards the course mark?)
- To what extent were the videos and MCQs useful?
- I used less required reading than in previous years, and didn't (as I intended) also suggest further non-examinable reading. If I had, would you have read it? Have you read all the required reading?

# Coursera

I'm contemplating writing a short course on using UML for Coursera; if I did so, there would be more, and more professional, videos with accompanying questions, covering the "mechanics" of UML (but probably not the OO design principle material).

A possibility is to use this material in SEOC as well as for worldwide students, e.g. have the first few weeks of SEOC be this course, then go on with extra material just for SEOC.

- On the spectrum from "great, extra flexibility to do the work when I wanted" to "swiz, no proper in-person lectures", how would you feel about that?
- What kind of extra input (just for Edinburgh students) would you like? Example classes, tutorials, office hours? Should this be optional or compulsory?
- Any other comments?