Software Engineering with Objects and Components

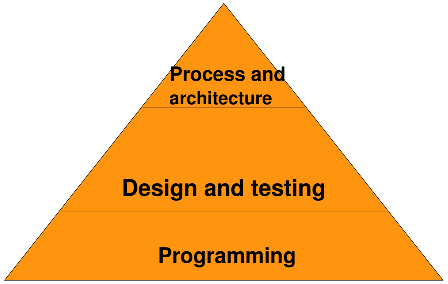
Perdita Stevens

School of Informatics
University of Edinburgh

Plan

- What's this course about?
- How will the course run?
- What are you supposed to know already?

What's this course about?



We assume you can program in Java.

(Ambitious) Aim: after this course you will be able to be *trusted* with a Java compiler – if you understand some requirements you'll be able to build a Java program with a sane design to satisfy them.

If you take Software Testing as well, it may also be correct.

Method

Learning to design well is hard.

Teaching someone to design well is impossible.

But we can teach the skills that will let you learn. E.g.

- the vocabulary of design criteria: what makes a design good?
- how to model designs so that they can be discussed
- how to learn from others' knowledge e.g. recorded as patterns.

How will the course run?

No for-credit coursework but this must not mean no work!

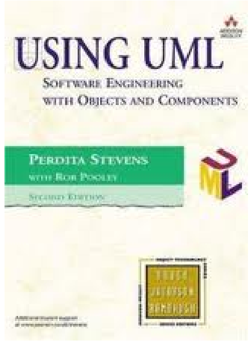
Some “flipping”: some weeks, I will ask you to read/watch videos teaching you basic information outside the timetabled slots, and will then use the timetabled slots to go through examples and let you ask questions.

Tutorials from Week 3. Each tutorial has an associated exercise sheet, available a week in advance.

You will not be admitted to a tutorial without a copy of the tutorial sheet and a genuine attempt at having done it.

Schedule time for doing SEOC work.

Recommended book



Second-hand copies are fine, but make sure they’re second edition (for UML2).

Beyond exam success

80% of success is showing up.

80% of becoming a good software designer is caring and thinking about software design.

From now on, every time you read or write code, ask yourself: why is it designed this way? Could it be improved? How?

What are you supposed to know already?

1. How to program competently in Java (Inf1-OP)
2. What software engineering involves (Inf2C-SE)