OO Project Intro

TDA367/DIT212

Joachim von Hacht

Course Intention

Exercise what you learned so far, i.e. creating a non-trivial modest sized application from scratch

- So far you have got (a lot) of starting help ...
- Now, you are on you own!

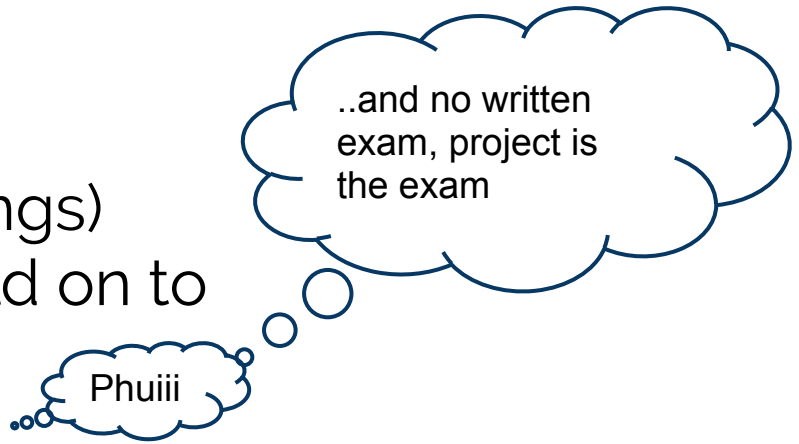
As you notice this course is spoken in Swedish and mostly written in English

Sorry..

Correction ...

Not completely on your own

- Have a group (4 stud/group)
- Have an assistant (weekly meetings)
- Will have a process model to hold on to
- ...so this will be great fun!



Last year was a big success

- Overall student impression ca 4,5 ... no major changes...
- Course is rather intense in the final weeks...

Target audiences

Course has 2 major target audiences

- IT programme year 1
- GU/CS year 2
- Others, year ... ?

Will handle this as a year 1 course!

- GU/CS have heard some of this, possible will find tempo a bit slow

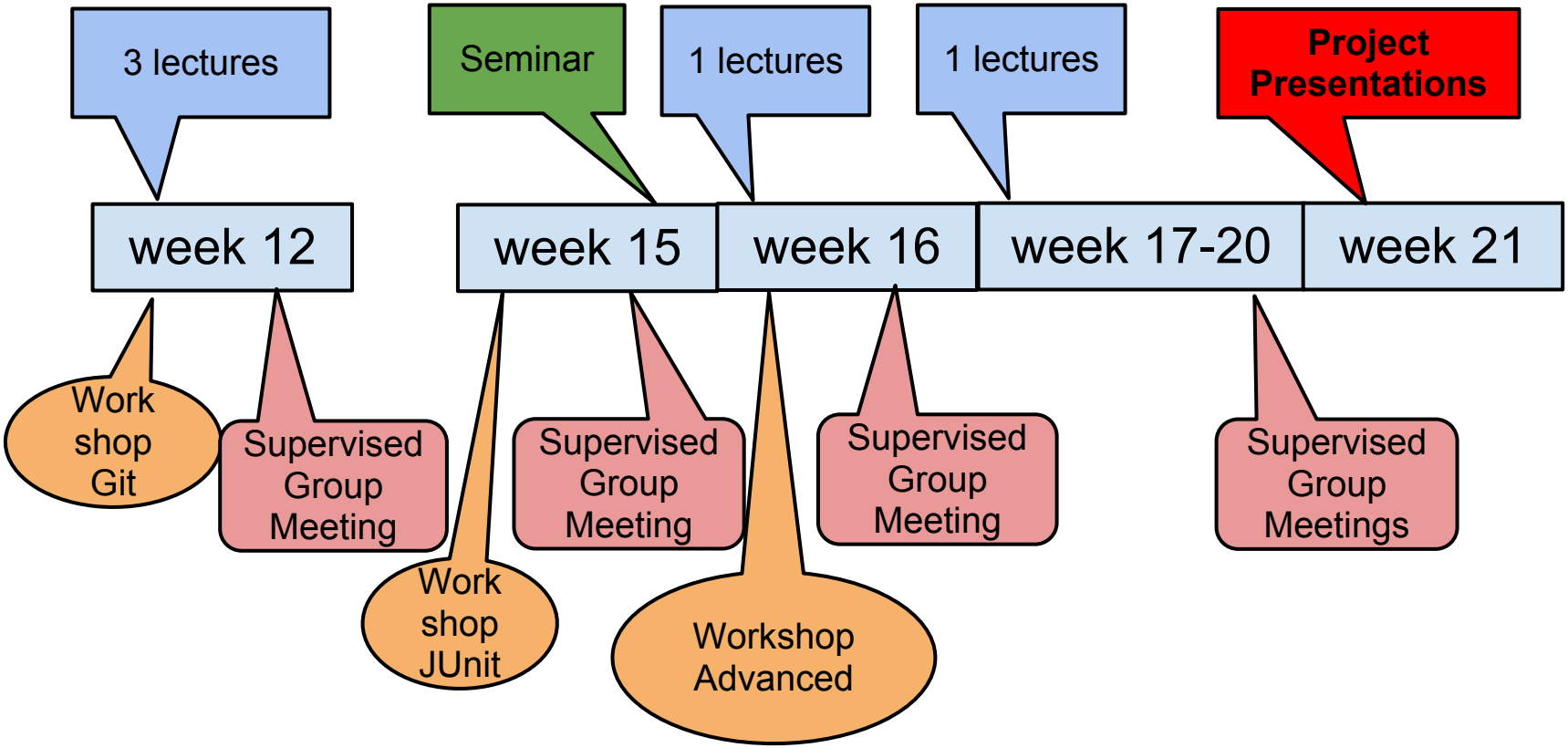
Course Position

Course is part of the OO-trail

In real life ordering is reversed

- First OOA ...
- then OOD...
- and finally OOP (we'll do it that way)

Roadmap and Organization



Detailed roadmap on course page

Lectures

Mostly as usual

- Only 5 lectures
- Will try to be somewhat interactive (because I'll try to explain a process)
- There are slides, I'll try to talk freely from the slides, possible will not show each and every, you look (course page)
- Slides mostly will come after lecture (because of interactivity)

Workshops

Designed to quickly get you going

- Basic Git, basic JUnit.
- Using additional software (libraries), advanced
- Workshops are self instructing, but assistant present just in case...

Our standard development environment is: Eclipse

- Knowledge assumed
- Many tutorials on the Web

Group Meetings

You are supposed to organize 2 weekly documented meetings on your own

- I.e. written agendas/outcomes
- Template for agenda on course page
- Agendas/outcomes in English or Swedish

The supervised meeting 1/week are mandatory

- If absent have to do compensational assignment

Supervised Meetings

Role of assistant is to help on an overall level (process, design, ...)

- All assistants have instructions what to do, if problems contact me
- Not a bugfixer
- If adding extra libraries you are on your own (assistants don't know all graphics libraries/physics engines...)
  - You are encourage to try (will give edge to project)

You are supposed to push ... collect questions

Seminar

End of study week 2 you will present a (very) preliminary model for your application, more to come...

- Short 10 min.
- What are we going to do...
- Overview of the model (a class diagram, pick up your UML-skills...)

Presentation of Project

Presentation is a part of the learning (and the grading)

- About 15 min
- Running demo of project
- Technical walkthrough
- Opposing 1 other project i.e. questions to the group regarding the project (technical solutions/alternatives etc)
- Must be present for one day, you're encourage to question (after opponents)
- For IT presentation part of cooperation with "LSP310 - Kommunikation och ingenjörskompetens"

The project

You almost certainly will not be able to finish

- When is an application finished?
- We expect a prototype (but of course much functionality will impress)

Selected project normally not crucial for the grade

- Almost any project can be "complexified"
- Discuss with assitent!

Project Type

Expected application type is a standalone, end user application with a GUI

- Highest grade can be achieved by this

Of course you may create more advanced projects

- But it's not a prerequisite for highest grade

Course Grading

Step 1: The project

- The project will get an overall grade

Step 2: Individual

- Each individual will get a grade
- Mostly project and individual grade will be the same but if we see big differences they can vary

See course and project PM on course page for more details

Project Grading

Criteria

- Functionality, how much is working?
- Complexity (problem and/or technical) and size
- Quality; design, adhering to design principles, technical solutions, code organization, coupling/cohesion, tests, ...,
- Traceability, is it possible to follow the process?
- Documentation, is it possible to gain some understanding?
- See Course PM and Project PM for details (course page)

Individual Grading

Actively contribute to the process, attending meetings etc

- Ok, with different ambitions, group decide, speak out!

Contribute at least 600 rows of good quality/meaningful code (excl. comments)

- Not auto generated (GUI code or similar..).
- We will be able to trace this!
- Take turns when committing to code version handling system
- Annotate classes with **@author** and use "revised by..."
- Document in agendas who is responsible for ...?

Who should Participate?

If failed both preceding OO course, this is not a good course to take

- Take any programming course instead!
- IT-program has been informed

Questions

???