



Aalto University  
School of Science

# CS-C2120

## Ohjelmointistudio 2: projekti

Alkaa 14.15

Lauri Malmi, Otto Seppälä

13.1.2021

# Lecture practices

- I recommend using laptop / desktop when following lectures. Mobile screen is small and it may be hard to read slides / other content.
  - Keep your microphone closed in Zoom to avoid distracting voices.
  - If you want to ask something
    - Send the question in chat to all (public for all) OR
    - Send a *private message* to Otto Seppälä (hidden, maintains anonymity)
    - Otto follows the chat and responds in writing or notifies me.
  - Occasionally, I set up a poll (interactive question), where you can (anonymously) select answer(s) on multiple choice questions on your own computer.
  - Breakout rooms not used in this lecture.
-

# General

- The course is a direct continuation of the course Programming 1.
- Targeted to CS major students
  - Many other students also take the course
  - 5 ECTS
- Continues the Programming MOOC with 2 credit extension.

# Two parallel courses

- Programming Studio 2 and CS-C2105, Programming Studio A, are parallel courses.
    - Studio 2 given in Finnish
    - Studio A given in English
    - Joint organization and schedule
    - The course contents and requirements have very little differences in Spring 2021.
-

# Goals 1

- Learn some approaches in program design and implementation methods
    - Focuses on OO design
    - Basics of UML
    - Some design patterns
    - Testing
    - Version controlling
  - These will be applied in a personal project work.
-

# Goals 2

- Learn some new important features of Scala
  - File management
  - Exceptions
  - Types and type management
  - Basics of building graphical user interfaces
  - Threads

# Prerequisites

- Programming 1
  - Strong recommendation that you completed at least most of B level assignments
  - If not, this course could be difficult.
  - If you completed Programming 1 earlier than last autumn, use some time to recap your Scala programming skills.

# Requirements 1

1. Weekly exercises during period 3
  - UML design task
  - Scala programming exercises
  - Version control exercises
2. Chapter feedback
  - The same practice as in Programming 1
  - Weekly summaries will be short



# Requirements 2

3. Personal programming project
  - Includes project plan, technical plan, implementation and demo
  - *This is the most important part of the course.*
4. Course feedback survey at the end

# If you started earlier...

- If you completed weekly exercises in spring 2020, but did not complete the project.
    - You need to do only the project
    - You must follow the instructions on this spring course.
  - Otherwise, you need to redo the whole course
-

# Grading

- Weekly exercises (30%)
- Project work (70%)
  - Project grading covers many aspects, e.g., program functionality, user interface features, code quality, data storage, testing and documentation.
- Note: MOOC students do only the exercises, not the project

# Exercises 1

- Includes
    - Some multiply choice question to check understanding of new concepts
    - Programming exercises
    - UML design exercise
    - Version control exercises
  - Can be solved using pair programming
    - Register as a group in A+, if you want this.
-

# Exercises 2

- Automatic assessment used in most assignments
    - Allows several resubmissions after getting feedback.
    - The best result is recorded.
  - The deadlines are strict
    - After DL you can submit your work, but gain no points.
    - In case of system problems we postpone deadline and announce this in MyCourses / A+, if possible.
  - **Start early. Do NOT start working the previous day before the DL.**
-

# Exercise deadlines

- Chapter 14 (UML)
  - 27.1 at noon
- Chapter 15 (Exceptions)
  - 3.2 at noon
- Chapter 16 (File management)
  - 10.2 at noon
- Chapter 17 (Version control, testing)
  - 17.2 at noon
- Chapter 18 (Types)
  - 3.3 at noon
- Chapter 19 (Design patterns, graphical user interfaces)
  - 10.3.3 at noon



No DL on exam week

# Chapter feedback

- We collect feedback from each chapter to improve the course material, and follow how much time you used
  - Your exercise points are recorded when your feedback has been accepted.
- Some form of weekly summary is created after chapter DL.

# Course feedback

- Collected using standard practice at the end of the course
  - It is also important for us to improve the course.
- Changes implemented since last year:
  - IntelliJ used instead Eclipse. The same integration to A+ as in Programming 1
  - Some new small exercises are added.
  - Course learning resources have been polished.
  - Some clarifications for project topics.
  - A realistic document of a program development process



# Personal project 1

- Designing and implementing a somewhat larger program independently
  - Applying methods and practices learned during the weekly exercises.
- Parts
  - General plan (DL 17.2)
  - Technical plan (DL 19.2)
  - Interim reporting in version control
  - Optional interim meetings in March / April
  - Implementation and documents (DL 28.4)
  - Demo (late April, May)

# Personal project 2

- This is a *personal* task.
    - You can discuss the project with peers but you code the program yourself.
  - You can choose from many different topics
    - Own topics can be suggested, and accepted if they meet the project goals.
    - Suggestions to Lauri Malmi by Feb 3rd
-

# Resources

- Online course material in A+
  - All assignments are in A+.
- MyCourses is used
  - To give general announcements, for example, changes in schedules or practical arrangements.
  - To publish lecture materials

# Lectures

- Lectures in Finnish, Wednesdays 14.15-16 (Zoom)
    - 13.1 Introduction, program design
    - 20.1 Program design cont., UML
    - 27.1 Program design cont.
    - 3.2 Version control, testing, project introduction
    - 10.2 Project planning
    - 17.2 Graphical user interfaces
-

# Lectures

- Additional demo sessions, in English Fridays 12.15-14
  - Joint session for Studio 2 and Studio A
  - Practical design cases, demonstrations, live coding examples
  - **NOT on Jan 15th**

# Exercise sessions

- Zoom sessions to get personal guidance from course teaching assistants.
  - Voluntary, recommendable
  - Period 3 (starting at 22.1)
    - Mondays 14.15-18
    - Tuesdays 12.15-16
    - Fridays, 14.15-16
  - Period 4
    - Mondays 14.15-16
    - Thursdays 12.15-14
-

# Zulip

- Zulip discussion forum (replaces Piazza)
- Present questions there.
- Assistants follow the forum and try to respond within 24 hours.
- You can get answers from peers, too.
  
- Telegram is a not formal support forum, while getting support from peers is possible there, too.

# No Email

- Do not email to teaching assistants.
- You can email to [Lauri.Malmi@aalto.fi](mailto:Lauri.Malmi@aalto.fi), but quick responses cannot be guaranteed.
  - No programming guidance



# Course staff

- Lauri Malmi (lectures, course organization, teacher in charge)
  - Otto Seppälä (lectures, demo sessions, course learning content, teacher in charge)
  - Teaching assistants
    - Ray Atreya, Khoa Lai, Trang Nguyen, Tuan Nguyen, Linnea Risku, Alena Shchevyeva, Valtteri Valtonen, Taige Wang ja Sergey Zakuraev
-

# Language

- Since most teaching assistants do not speak Finnish
  - Most support given in English
  - Please, submit documents in English

# Questions?