



Aalto University
School of Science

CS-C2105

Programming Studio A

Lauri Malmi, Otto Seppälä

12.1.2022

Lecture practices

- Keep your mic closed in Zoom to avoid distractive 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)
 - Occasionally, I might set up anonymous zoom polls (interactive question)
 - Consider adding a zoom profile picture – It's more fun seeing people (or fun avatars) than black screens
-

General

- The course is a direct continuation of the course Programming 1.
 - Target audience: Data Science students of the Aalto Bachelor's Programme in Science and Technology.
 - English speaking students from other programs can take it, too.
 - 5 ECTS
 - Continues the Programming MOOC with 2 credit extension
-

Two parallel courses

- Programming Studio A and CS-C2120, Programming Studio 2 are parallel courses
 - Studio A given in English
 - Studio 2 given in Finnish
 - Joint organization and schedule
 - The course contents and requirements have fairly small differences in Spring 2022.

Goals 1

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

Goals 2

- Learn some new important features of Scala and programming in general
 - 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 refresh 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 last year, but did not complete the project.
 - You need to do only the project, but earlier
 - Contact the lecturer immediately
- Partially completed exercises are not considered

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/FiTech students do only the exercises, not the project

Exercises 1

- Includes
 - Some multiply choice questions 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)
 - 25.1 at noon
- Chapter 15 (Exceptions)
 - 2.2 at noon
- Chapter 16 (File management)
 - 9.2 at noon (**likely to change**)
- Chapter 17 (Version control, testing)
 - 16.2 at noon (**likely to change**)
- Chapter 18 (Types)
 - 2.3 at noon (**likely to change**)
- Chapter 19 (Design patterns, graphical user interfaces)
 - 9.3 at noon (**likely to change**)



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 at the end of the course
 - Used to improve the course.
- Changes implemented compared to Studio 2 course last year:
 - Some new small exercises will be added.
 - Course learning resources have been polished.
 - More projects directed to Data Science topics.
 - A debugging exercise / guide

Personal project 1

- Designing and implementing a somewhat larger program independently
 - Applying methods and practices learned during the weekly exercises.
- Parts
 - General plan (DL 16.2)
 - Technical plan (DL 18.2)
 - Interim reporting in version control
 - Optional interim meetings in March / April
 - Implementation and documents (DL 27.4) **(likely to change, will be coordinated with other courses)**
 - 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 a set of topics
 - Own topics can be suggested, and accepted if they meet the project goals.
 - Suggestions to Otto Seppälä by Feb 3rd

Resources

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

Lectures

- Lectures in Finnish, Wednesdays 12.15-14 (Zoom)
 - 12.1 Introduction, program design
 - 19.1 Program design cont., UML
 - 26.1 Program design cont.
 - 2.2 Version control, testing, project introduction
 - 9.2 Project planning
 - 16.2 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

Exercise sessions

- Zoom sessions to get personal guidance from course teaching assistants.
 - Voluntary, recommendable
 - Period 3 (starting at 17.1)
 - 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
- 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

- Use the exercise groups and the discussion forums for programming related questions
- Do not email to teaching assistants.
- You can email the lecturer, but quick responses cannot be guaranteed.

Course staff

- Otto Seppälä (lectures, demo sessions, course learning content, teacher in charge)
- Teaching assistants

Questions?