



Aalto University
School of Science

CS-C2120

Ohjelmointistudio 2: projekti

Alkaa 14.15

Lauri Malmi, Otto Seppälä

10.1.2024

General

- The course is a direct continuation of the course Programming 1.
 - Targeted to Computer Science and Information networks students.
 - Students from other programs can take it, too.
 - 5 ECTS
 - Continues the Programming MOOC with 2 credit extension
-

Two parallel courses

- CS-C2120, 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 2024.

Goals 1

- Learn some approaches in program design and implementation methods
 - OO design, Basics of UML, Testing, Implementing, Version controlling
 - Learn some new important features of Scala
 - File management, Exceptions, Types and type management
 - Basics of building graphical user interfaces
 - Threads
 - We use Scala version 3
-

What is new this year?

- More lectures to provide more examples of program design and development
- Several new exercises
- More learning resources about program design and implementation
- We recommend using ScalaFX graphics library
 - While Swing is still supported
- A new model project to provide an overview of an average, grade 3 project.
- Project is carried out in 2 week sprints.

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. Weekly exercises during period 3
 - Programming exercises, UML design, version control, GUI exercise
 2. Chapter feedback
 - The same practice as in Programming 1
 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 2023, but did not complete the project.
 - The grade of weekly exercises is still valid
 - You need to do only the project
 - However, we recommend doing the weekly exercises this spring (many new exercises)
 - If you continue the previous project
 - contact Lauri Malmi (Lauri.Malmi@aalto.fi)
 - You can begin working with the project immediately
 - If you restart or change topic
 - You must follow the instructions on this spring course.
 - Anyway, the project will be graded based on this spring requirements.
 - If you completed weekly exercises in spring 2022 or earlier, you need to redo the whole course.
-

Grading

- Weekly exercises (30%)
 - Automatic assessment
 - UML design task and GUI basics task manually graded
 - Project work (70%)
 - Project grading covers many aspects, e.g., program functionality, user interface features, code quality, data storage, testing, documentation and the project overall progress.
 - Bonus for submitting a working project a week early.
 - Note: MOOC students do only the exercises, not the project
-

Exercises

- Rounds open on Wednesdays and are closed on Fridays at 18.00 the following week.
 - Can be solved using pair programming
 - Register as a group in A+, if you want this.
 - 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.**
-

Personal project

- Designing and implementing a somewhat larger program independently (no pair programming)
 - You can discuss the project with peers but code the program yourself.
- Schedule
 - Topics published 22.1
 - Topic selection 31.1 (including own topics)
 - Project planning
 - General plan (DL 7.2)
 - Technical plan (DL 12.2)
 - Implementation
 - Carried out in 2 week sprints, where you meet the personal assistant every two weeks
 - Reviewing progress, discussing problems, setting goals for the next sprint
 - Submission
 - Implementation and documents (DL 25.4)
 - Demo (late April, May)

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.
 - *Follow announcements in MyCourses and A+*
-

Lectures

- Lectures (in Finnish),
 - Wednesdays 14.15-16, T1
 - Fridays 14.15-16, AS2

Other support

- Voluntary exercise sessions
 - Mondays, Tuesdays, Thursdays, Fridays
 - Teaching assistants are present to help you
 - Zulip discussion forum
 - Present questions there.
 - Assistants follow the forum and respond typically in a few hours, but at the latest on the next day, also during weekends.
 - You can get answers from peers, too.
 - Telegram is a not formal support forum, while getting support from peers is possible there, too.
-

Email

- *Do not* email to teaching assistants.
- You can email to 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, course learning content, teacher in charge)
- Teaching assistants
 - Head assistant: Eli Roslöf
 - Aino Kurri, An Bui, Fathima Afrooz Abdul Mahir, Hien Ta, Hung Nguyen, Niklas Koskela, Onni Komulainen, Quan Hoang, Roope Kettunen, Tuomo Ohvo, Ukko Miettinen

Language

- Many teaching assistants do not speak Finnish
 - Often support is given in English
 - If your personal teaching assistant in project does not understand Finnish, please, submit documents in English
 - Where possible, we try to organize the projects so that Finnish speaking students would have Finnish speaking assistant.
-

ChatGPT, CoPilot, ...

- You are learning to program and improving your programming skills.
- Therefore, AI-supported code generation tools are not allowed in the course to solve the exercises or the project.
- You may be allowed to use them in some future courses and probably will use them working life, but *before you can use them in a meaningful way, you need to be a competent programmer yourself.*

Questions?