ICSS Final Projects Guide

David Garcia, Winter Semester 2025/26

Grading of Introduction to Computing for the Social Sciences (Practice)

ICSS (Theory) and *ICSS (Practice)* are graded separately and it is possible to pass one and not the other. You do not need to pass both parts the same year but we strongly recommend that you focus on passing both in the first semester.

The Grading of *ICSS (Practice)* is separated in two parts:

- **Mandatory assignments:** To be allowed to submit a final project, you need to pass each of the four assignments during the semester. These are ungraded, they only have a pass/fail grade, but you need to pass each one of them. If you don't pass the mandatory assignments, you will have to do *ICCS (Practice)* next year. We will send the grade of the last assignment no later than two weeks after the deadline.

- **ICSS Final Project:** The final numeric grade of *ICSS (Practice)* is the grade of the ICSS Final Project. To submit an ICSS Final Project, you have to register in Zeus for the exam of *ICSS (Practice)*. The grades of *ICSS (Theory)* and *ICSS (Practice)* are independent and each course is worth 6 ECTS.

Aims of the ICSS Final Project

The **general aim** of the ICSS Final Project is for individual students to **develop and showcase their practical learning** through a programming project of their own choice. Toward this, there are three more specific aims:

1. Apply course skills to **build software**. The central product of an ICSS Final Project is the software, not the report or the video. You will make presentations and write papers in other courses, here you have to focus on building and explaining how and why.
2. **Learn by doing** and expand skills beyond what is covered in the course. You should choose at least one component not covered in class and learn it on your own. For example an API, a new library, another algorithm, a method, etc.
3. Develop the skill of **planning a project.** This is an iterative project, not a final exam. During the semester, there will be plenty of chances to get feedback from lecturers and tutors, especially to discuss your ideas, feasibility, resources, etc.

The workload estimated for the ICSS Final Project is a total of 90 hours of work. This should be done during the second half of the semester and during the exams period. Start working on your ideas from December and discuss your approach and aims with your colleagues, lecturers and tutors. Do not leave this for the last weeks, a project is best done over a longer period of time than in a stressing crunch.

Grading criteria for ICSS Final Projects

The numeric grade of an ICSS Final Project is a combination of the grading of code and report. The precise weighting of each part depends on the project tasks and aims but each part will weigh no less than 30% of the final grade. The final grade is composed of the three following parts:

1. **Code grade (numeric).** Your code will be audited as part of the grading process. The final code grade will be composed of the following components:

    - **Code readability:** The code is implemented following good programming practices covered in the course (e.g. clear variable names, structured flow). In-code documentation (e.g. docstrings) must be useful and comments should be meaningful without excess, justifying the rationale behind the code.
    - **Code organization:** The project is properly structured and separated into different files. There is no code redundancy, no unnecessary code or functions, nor repeated code that could be reused as functions. The code should be organized in a way that makes it possible to maintain, extend, and inspect possible errors. The coding languages chosen are appropriate for the tasks.
    - **Code execution:** Code installation and running are documented in a way that allows a skilled user to execute it. As a standard, a fellow student should need at most 10 minutes to be able to install resources and run the code on their own. Executing the code allows the lecturer to reproduce the results of the project and code execution does not show bugs or unstable behavior.

2. **Report grade (numeric).** The report helps the grading of parts of the project that are not immediately visible in the code but that pertain to the motivation, planning, and results of the project. This report is not a research paper but it has to give context, motivation, and interpretation of the result or product of the project**.** It should cover the following parts, each one receiving its own mark:

    - **Motivation:** Briefly describe the task or problem tackled by the project and why this development should be done.
    - **Design:** outline your approach to developing the project, breaking it down into subtasks, their relationships, and describing the overall structure of your software. Here, a diagram or a figure explaining the design is usually helpful.
    - **Innovation:** Detail what are the new components of your project that have not been covered in class. Give details on what you learned in terms of algorithms, packages, resources, etc. The level of learning in this new skill development will be graded. What is not graded is the novelty of the project as if it was a research project, it is OK to rebuild something that already exists. *You must show clearly what you have developed yourself and what you use from the work of others.*
    - **Results:** Illustrate outcomes or findings of the project and/or demo of the functionality of the software.
    - **Outlook:** Brief report on challenges, issues, and further ideas building on the project.

    The report should be at most about 3000 words long but this limit is not strictly enforced. There is no limit on the number of pages as some projects might need more figures or diagrams than others. There is no template but the report has to be a **PDF file** (no Word or other formats).

3. **Short video grade (pass/fail).** The short video is only graded with pass/fail and it is necessary to pass it in order to pass the ICSS Final Project. The short video offers a channel for you to show insights that are not captured well in code, comments, or the report. In the short video, you have to answer some questions (see below) which are designed to show your critical understanding of the project. If you abuse Artificial Intelligence tools during your ICSS Final Project, it will be clear in the explanations you provide in the short video.

Producing a short video for ICSS Final Projects

The duration of the short video should be between 5 and 8 minutes. Duration limits will not be strictly enforced, the idea of the short video is to be a lightweight way for you to explain your code without a lot of investment into production.

In the video, your face should be visible and the code must be readable. Your face can be in a smaller window but should be visible at all times during the video.

The video must have the following structure and address the following questions:

1. **Introduction and project presentation**
    - Start your IDE and open the code of your project
    - Briefly present the motivation and aims of your project:
        - What is the aim of the project or problem tackled?

2. **Demonstration of the solution**
    - Run your code
        - Show that the program executes properly from your submitted code
    - Explain the solution approach
        - Guide the watcher over the structure of your code
        - Highlight the new part or parts of your project that were not skills covered in the course. Briefly explain how you learned about them

3. **Reflection. Answer the following questions:**
    - What difficulties did you encounter in the project? What was the hardest part?
    - How did you overcome or solve those difficulties?
    - What can be improved in your project and its code?

For the submission, upload the video to the university cloud (https://www.kim.uni-konstanz.de/services/datenserver-und-cloud/cloud/) and share the file with the lecturer (David Garcia Becerra). Include the link to your video in your code submission in the github repository.

Submission process

A final project submission is composed of:

1. **Project code.** Create a folder with your project on your classroom repository and push the code there. If any data or additional resources are small enough to be uploaded to Github, include them there. If you need files larger than what can be uploaded to Github, use the university cloud, share the file with the lecturer and add a link in your project.
2. **Project report as a PDF file.** Add this report to your folder on github with a clear name. You can add a README file to the folder to clarify where the code is, which is the report file, the link to the video, and how to install and run the code.
3. **Short video** file in MP4 format. Upload this file to the university cloud, share it with the lecturer, and add a link to it in the project repository.
4. **Signed declaration of authorship** as a PDF file in the same folder on github.

The deadline for the final project submission is 22.03.2026, 23:59

Honor code

The ICSS Final Project has to be your own, nobody else should write a single line of code, report text, or content for the video that appears to be produced by you in the final project. You are allowed to use packages or software developed by others but it has to be clear what is developed by you and what is by others. You are **not allowed to copy code** from online or offline resources for the project without clearly showing what you reuse, including any code generated by AI coding assistants. If you abuse AI or help from others, this will be visible in the quality of your code and of your video explanation.

Using AI, for example Large Language Models, is allowed as a substantive part of the project implementation, for example in a module to analyze text as part of your software or for a chatbot that is part of the project. Any use of AI to generate code or text for the final report must follow the university's principles of good scientific practice. In the case of an adoption of AI-generated code or text that go beyond purely formal, stylistic corrections, you have to clearly identify them and, in addition to reporting the AI tool used, you have indicated the exact prompts you used. You will have to submit a Declaration of Authorship with your final project in which you report if or how AI assistants were used. Abuse of AI for coding or writing can lead to plagiarism investigations. We expect you to behave honorably and properly credit the work of others without misrepresenting your own work. The template for the Declaration of Authorship is distributed with this guide.