

ENSC 251 Final Project

Part 2 - Implementation

Weight: 60% of Final Project Grade

Teams: Same Teams as Part 1

1. Introduction

In this phase, your team will implement the software design developed during Part 1. The final result should be a complete, playable card game written in C++.

Although changes to your original design are expected, your final implementation should remain faithful to the goals and scope proposed in Part 1.

2. General Requirements

Your program must:

- compile in the course development environment
- be playable from beginning to end
- consist of multiple source files
- demonstrate object-oriented design principles
- implement the agreed-upon minimum viable game
- include appropriate documentation and testing

Networking and graphical interfaces are not required.

3. Deliverables

Submit a zip archive containing:

- Final report in PDF format
- Complete source code (.cpp and .hpp/.h files)
- Any additional documentation required to run and evaluate your project

4 Final Report (35 marks total)

4.1 Final Architecture (5 marks)

Provide an overview of the final software design and describe the purpose of the major classes in your implementation. This section should briefly explain how the classes interact and how responsibilities are divided throughout the program.

4.2 Design Changes (5 marks)

Describe any significant differences between the original design and the final implementation.

Examples include:

- classes that were added or removed
- major interface changes
- changes to the division of responsibilities
- changes to the game mechanics

Explain why these changes were necessary.

4.3 Testing and Validation (15 marks)

Describe the testing that was performed.

Include:

- features tested
- bugs discovered
- bugs fixed
- known limitations
- examples of important test cases

You should demonstrate that your program has been systematically tested.

4.4 Lessons Learned (5 marks)

Discuss what worked well during the project, what difficulties were encountered, and what you would do differently if you were to complete the project again.

4.5 Future Improvements (5 marks)

Describe additional features that could be added if more time were available.

Examples include improved computer players, additional game modes, networking, graphical interfaces, or additional rule variations.

4.6 Game Rules

Include a description of the game rules as implemented. If it is unchanged from part 2, that text can be included directly. If you have made adjustments to the game make sure to include that here.

4.7 Team Responsibilities

Compare the planned responsibilities from Part 1 with the actual responsibilities during implementation.

Explain any major changes and describe the contributions of each team member.

5. Functionality Demonstration (50 marks total)

During the demonstration every team member should be able to:

- explain the overall design
- discuss the code that they were primarily responsible for
- demonstrate the functionality of the program
- answer questions regarding implementation decisions.

Functionality includes:

- correctness of the implementation
- completeness of the game
- adherence to the documented rules
- robustness and input handling

6. Code Organization and Commenting (15 Marks)

- The final project will be evaluated using the following criteria:
 - readability and naming conventions;
 - organization of source files and classes;
 - quality and usefulness of comments;
 - overall maintainability and professionalism of the code.

The complexity and ambition of the chosen project may be considered when evaluating the overall quality of the implementation.