

# INTRODUCTION TO PROGRAMMING

## DM550, DM857, DS801 (Fall 2019)

**Exam project: Part I — Deadline: 23h59 on Friday, November 1st, 2019**

### Overview

In this part of the project, your task is to develop the top-level loop of the card-playing game. Your program needs to work with and maintain three objects: (a) an instance of class **Board**, which keeps track of the state of the game (including the state of the deck, the trump card, references to both players and information about whose turn it is to play); (b) an instance of class **Player**, which represents a player and keeps track of the cards that player has in his/her hand and the ones collected throughout the gameplay; (c) an instance of class **AutoPlayer**, which stores similar information about the automatic player, but also includes methods to decide on the next move.

Before starting the game, you need to interact with the user to obtain information about the player's name. Afterwards, you need to setup the initial state of the board and output all relevant information for playing.

Each iteration of the main loop should correspond to one round of the game, where you show the player's hand and ask for the next card he/she wants to play. Depending on whose turn it is to start the round, you may also have to output information about the automatic player's move.

When the game is over, you should compute the players' scores and announce the winner.

### Expected results

You must hand in a class **Game.java** implementing the functionality described above. Your class should be a client of the remaining classes, which are provided in compiled form. It should run directly from the command line, print all relevant information on the screen, and ask for any relevant input using only command-line interaction.

You should also hand in a report describing the algorithm implemented, any design choices that you think are important to document, and examples. One of the members of the group must be clearly identified on the title page as the coordinator for this part of the project. The source code for the developed classes should be included as appendix.

The report will be the basis for the evaluation.

### Testing and examples

Include some examples of how your program works. Rather than copying the output of a complete run in your report, you should describe any auxiliary tests that were made and explain how they can be used to ensure that specific parts of the code are running as expected.