

Card Game

Your task is to build a card game for 2 players played on the **command line** (not a web application).

Organizational

The rules of the game are listed below and split into tasks. For each task, please implement the indicated unit-tests. You should not work longer than 4 hours total on this assignment. The order of tasks is just a suggestion, feel free to implement in whatever order you see fit. Please submit your solution even if you were not able to finish everything within 4 hours.

The task should be completed using either Java, JavaScript/TypeScript (NodeJS), Go or Python. Make sure to put all required files in a **.zip** archive (not .rar or .7z) and upload this archive to our submission system (see e-mail for details). Avoid uploading unnecessary files like binaries or dependencies, but please do include a `README.md` that describes how to install dependencies, compile and execute the solution.

If you have questions regarding this task, feel free to send an email and we will get back to you as soon as possible.

Task 1: Create a shuffled deck of cards

Each card shows a number from 1 to 10. Each number is in the deck four times for a total of 40 cards. At the beginning of the game, the deck is shuffled (*Hint: Look up `Fisher-Yates Shuffle Algorithm`*) to make sure it is in a random order.

Each player receives a stack of 20 cards from the shuffled deck as their draw pile. The draw pile is kept face-down in front of the player. Each player also keeps a discard pile (see "Task 3" for more).

Tests:

1. A new deck should contain 40 cards.
2. A shuffle function should shuffle a deck.
 - *Hint: Consider mocking `Math.random()` or the equivalent of your chosen language.*

Task 2: Draw cards

Each turn, both players draw the top card. If there are no more cards in the draw pile, shuffle the discard pile and use those cards as the new draw pile. Once a player has no cards in either their draw or discard pile, that player loses.

Test: If a player with an empty draw pile tries to draw a card, the discard pile is shuffled into the draw pile.

Task 3: Playing a turn

The players present their drawn card and compare the values. The player with the higher value card, takes both cards and adds them to their discard pile, next to the draw pile. If the cards show the same value, the winner of the next turn wins these cards as well.

Hint: The game will likely result in a stalemate, if this rule is not implemented.

Tests:

1. When comparing two cards, the higher card should win.
2. When comparing two cards of the same value, the winner of the next round should win 4 cards.

Task 4: Console Output

Your program should output the cards that are played each turn and who wins. At the end the program should output the player that won.

Example output:

Player 1 (20 cards): 8
Player 2 (20 cards): 1
Player 1 wins this round

Player 1 (21 cards): 1
Player 2 (19 cards): 10
Player 2 wins this round

[...]

Player 1 (38 cards): 4
Player 2 (2 cards): 4
No winner in this round

Player 1 (37 cards): 7
Player 2 (1 cards): 3
Player 1 wins this round

Player 1 wins the game!