

# CIS 1068 Assignment 6

## A Game

Due: ~~Friday, February 25~~ Monday, February 28

50 points

### a game

To pass the time during long winters, the ancient Nordic people would play the two-player game of *Uppföra*. In this assignment, you'll implement the game, but play will be against the computer.

During each round, players choose a move, which may be either Skadis, Tjusig, Klyket, Hovolt, or Pershult. The rules are:

- Pershult beats Klyket, Skadis
- Klyket beats Tjusig, Hovolt
- Tjusig beats Pershult, Skadis
- Skadis Beats Hovolt, Klyket
- Hovolt beats Pershult, Tjusig

The computer wins in the event of a tie.

Your program should behave as follows:

- [3 points] The rules should be printed to the screen
- [5 points] The user is asked if they'd like to play a round
  - if they choose 'y', a round is played
  - if they choose 'n', the program ends
  - Until the user has chosen to quit, another round is played.
- In a round of play:
  - [5 points] The user is asked to enter a move, which may be either Skadis, Tjusig, Klyket, Hovolt, or Pershult. The program should continue to prompt the user until a valid move is entered.
  - [4 points] The computer makes a move at random. (Hint: remember how we generated random numbers in class.)
  - [5 points] Determine the winner
  - The program prints the computer's move, the user's move, and who is the winner of this round.
  - The user is asked if they'd like to continue.
- [4 points] When the user has decided to quit the game, the program prints the number of:
  - rounds played
  - times the user won
  - times the computer won

### suggestions

An important skill in programming is learning how to break up a big job into smaller tasks.

- [5 points] Make an outline. Make sure that your outline makes sense. Test it out with real input using pencil and paper. Do this *before* you start writing code. Submit your outline as a separate file or within your java file as a comment.
- Turn some of the individual steps of your outline into functions. Some obvious choices might be a function which generates the computer's move at random. Another would be a function which is passed two moves and returns whether or not the user has won the round. The goal should be that the functions make your

code as readable as your English-language outline. You don't need to implement all of the functions at first. Just write placeholders (we call these stubs) first and fill them in later.

- Implement and test your stub functions.

Also remember to test things as you go. It's easier to find a mistake in 5 lines of code than it is to find a mistake in 500 lines of code.

## style

- [5 points] at least 5 separate functions defined and used
- [5 points] no global variables. Data is passed to and copied from functions
- [5 points] input from the user is validated. If the user enters invalid input, he or she should be prompted to re-enter the input until valid input is given. Note that you are not required to handle the problem of the program crashing when a Scanner detects input of the wrong type.
- [4 points] all input entered is echoed back to the user. For example, when the user has entered a move, you'd write back something like, "Your move is Skadis"

## testing (up to 5 extra credit points)

Remember that good code is readable and testable. In order to receive full credit for this assignment, you'll need to break up the problem into at least 5 methods separate from `main()`. Develop and submit JUnit tests for at least one of these.

Recall that to create most of the skeleton of these tests for yourself, in Eclipse, under the package tab, right-click on your .java file (CTRL-click on a Mac), select New, then JUnit Test Case. Check off the functions for which you'd like some starter tests written, and then click Finish.

Some functions that we've used that should be helpful are:

- `assertEquals(x, y)`
- `assertTrue(x)`
- `assertFalse(x)`

## what to submit

Please upload your .java file to Canvas. If you've completed the extra credit, instead zip the entire project directory and upload the zip.