Coding Dojo #1

Coding Dojo

- A Coding Dojo is a coding session centered around a small programming challenge, likely a Coding Kata.
  - One half of the pair is changed every 5 minutes.
  - The pair on the keyboard should continuously explain what they are doing.
  - The pair on the keyboard should stop when someone from the audience falls off the sled – and only continue when that someone is back on track again.
  - The audience should give comments on design only when there is green bar. (During red bar audience can only ask questions.)
  - The pair should not continue on writing new code if other participants are not happy with the current design. (The code should be always well refactored before starting to write new code.)

http://wiki.agilefinland.com/?CodingDojo
http://codekata.pragprog.com/2007/01/code_kata_backg.html

Kata Four: Data Munging

- **Part One: Weather Data**
  - In weather.dat you'll find daily weather data for Morristown, NJ for June 2002. Download this text file, then write a program to output the day number (column one) with the smallest temperature spread (the maximum temperature is the second column, the minimum the third column).
    - http://pragdave.pragprog.com/data/weather.dat

http://codekata.pragprog.com/2007/01/kata_four_data_.html

Kata Four: Data Munging

- **Part Two: Soccer League Table**
  - The file football.dat contains the results from the English Premier League for 2001/2. The columns labeled ‘F’ and ‘A’ contain the total number of goals scored for and against each team in that season (so Arsenal scored 79 goals against opponents, and had 36 goals scored against them). Write a program to print the name of the team with the smallest difference in ‘for’ and ‘against’ goals.
    - http://pragdave.pragprog.com/data/football.dat

http://codekata.pragprog.com/2007/01/kata_four_data_.html

Kata Four: Data Munging

- **Part Three: DRY Fusion**
  - Take the two programs written previously and factor out as much common code as possible, leaving you with two smaller programs and some kind of shared functionality.
- **Kata Questions**
  - To what extent did the design decisions you made when writing the original programs make it easier or harder to factor out common code?
  - Was the way you wrote the second program influenced by writing the first?
  - Is factoring out as much common code as possible always a good thing? Did the readability of the programs suffer because of this requirement? How about the maintainability?

http://codekata.pragprog.com/2007/01/kata_four_data_.html