Exercise set 1

Return your solutions via Moodle page **no later than 21.03.2016 by 10AM strictly**. We do not count belated returnings. You have to use .pdf file type for any written answers, and .scala for your code. Do not return any project or object files, etc. Mark your **full name and student number** clearly to all your solution files. Prepare to explain and discuss your solutions on Friday exercise session. Each exercise will be graded pass/fail.

Exercises 1-3 are warm ups to the course. There can be more than only one correct answers. Feel free to use references, figures, etc. in your solutions.

Exercise 1

a. What does it mean when we discuss Big Data? What makes it “Big”? Give some explanations you can find from literature/Internet as well as your own opinion as a short essay. Use references where needed.
b. Describe a practical use case or application for each of these datasets:
    i. All flights by commercial airlines from year 2014 (*Addition:* You can assume that the data have all the departures and arrivals, with times and airlines, from all the airports.)
    ii. Applications and system settings of 750.000 mobile devices from three years
    iii. Entire English Wikipedia text dump with editing history

Exercise 2

Describe a data mining / machine learning algorithm of your choice, e.g. from courses you have passed in your earlier studies (Introduction to Machine Learning, Tietorakenteet ja Algoritmit, etc.). Give an example, for which purpose your algorithm works, and what kind of problems you could face when implementing it in the distributed environment. You are not supposed to implement anything, but discuss the problems and possible solutions.

Exercise 3

Describe problems and possible solutions you can face when managing, storing, and analyzing:

a. A 10TB text data set in the cloud, where each line represents one element of the data and each file contains from 10.000 to 100.000 lines
b. A data stream of 1000 elements per second, without a need to store everything but collect some useful information

---

Exercises 4-6 help you to understand Scala and functional programming before we go to Spark. Please return your solution code with comments that explain what you have done. Be sure that you can run Scala in your computer or use Department's machines (we prefer 2.10, but some of the Department's servers have 2.9.1 that will be satisfactory). DO NOT run any code of this course in the Department's remote servers (melkki, melkinkari). You can use them to connect to the Ukko cluster from home and run your code on it: https://www.cs.helsinki.fi/en/compfac/high-performance-cluster-ukko

Exercise 4

Find one or two Scala tutorials from Internet that you like. Finnish speaking students can also benefit materials of the Department's Bachelor degree Scala course: http://www.cs.helsinki.fi/u/wikla/OTS/Sisalto/

Write a basic Hello World program that prints a sentence you want, iterates ten times, and every time changes one letter of a sentence so that result is something funny. You can use for loop there.

Exercise 5

Make yourself familiar with the Scala interpreter (called as REPL). You can run this exercise in REPL, but please still return a .scala file as a solution.

Write a function that creates a random array of 100 elements. Use the nextGaussian() function. Then compute mean and standard deviation from the array without using a for loop. Instead, use e.g. map, reduce and filter functions.

Exercise 6

Make yourself familiar with Scala Objects and Classes. If you are familiar with Java, take into account how they differ.

Write a Class that describes a Student, with fields for a name, a student ID, an address, a year class, and a list of passed courses with their grades. Let the Class have a Companion Object, that includes apply method and other functions necessary so that you can:

- Create a new Student only knowing his/her name, and let other fields to be empty or defaults
- Print student information (nicely formatted): a name, a year class, mean of grades, and a list of passed courses

You can test your Class in REPL (you have to write it to a separate file first) or write a different main program to run tests. If you want an extra challenge, use ScalaTest for unit tests (not mandatory).