Course webpage: http://www.cs.ubbcluj.ro/~hfpop/teaching/2018/pfl/

Grading lab papers (grade Lab)

(1) Attendance at lab activities is compulsory in proportion of 90% (minimum 6 seminars out of 7). The lab attendance may be redone for at most **one** lab, during the two weeks allocated to that lab, with the explicit agreement of the lab teacher, but the lab doc is graded in the usual way, considering the delay penalty. In case of illness, the absence is justified to the lab instructor based on medical evidence only.

(2) The students without minimum 6 attendances at laboratory activities CANNOT participate to the written paper (during weeks 13-14 and during the reseat session) and CANNOT pass the exam.

(3) The lab grade will be determined as the average of the grades of all the lab works. If a lab doc is not delivered, its grade is 0.

(4) If a lab paper is submitted with another lab subgroup inside the same two weeks frame, the lab grade is **multiplied by 0.8**; if the submission is delayed one lab, the lab grade is **multiplied by 0.6**; if the delay is larger than one lab, the final grade is **1 (one)**.

(5) During one lab meeting **at most two lab papers** may be submitted, except for the labs with practical tests, where only one lab paper may be submitted. As well, the Prolog labs may be submitted by the end of week 7/8.

(6) A copied lab paper means cheating and will be graded with 0 (zero).

Each lab paper is evaluated with a grade from 1 to 10 in the following way:

- 1 point: start
- 1 point: lab code, lab topic description (title), submission deadline, problem statement
- 2 points: formal descriptions and explanations of recursive models
    - PROLOG - recursive models and flow models of all predicates used, meaning of all predicate parameters;
    - LISP - mathematical models (recursive formulas describing functions), meaning of all function parameters;
- 2 points: source code in Prolog/Lisp for all predicates / functions
- 1 points: testing examples covering as many testing cases as possible, for the essential predicates / functions
- 2 points: explanations of the written algorithm
- 1 points: execution verification of the written program