**CS1401 – Programming Assignment 11**

**Due: May 8, 2015 by 11:59pm**

**To be submitted via Piazza in the folder “lab12”**

**Welcome to Lab 12!**

This lab is your last lab. It is not to be done in Java like the previous 11 labs. This time, we ask you to do the following:

* Work in pairs (same pair as for Lab 11) for the Prolog exercises (Activities 1.1 to 1.5)
* Work individually on the Scratch assignment (Activity 2)

Lab 12 is to be done **over a period of almost two weeks**: please note that this time, the deadline is on a Friday, not on a Sunday, because the week after that is final exams’ week. There will be only one submission per pair for the Prolog exercises and one submission per student for the Scratch assignment.

Once again, do make sure that you follow instructions. This includes turning in all that is expected from you: look for “To be turned in” instructions in this document.

**Before you start:**

Please download swi-prolog on your laptop to be able to program on your own computer if you would like. To do this, please visit: <http://www.swi-prolog.org>.

Also, remember that for each submission, you do have to turn in your lab in the following format:

* Put all your files (the java files lab12.pl, your docx file) in a folder named after the last name followed by the first name of the first member of the pair, followed by the last name followed by the first name of the second member of the pair
* Compress this folder into a .zip file (nothing else than a .zip)
* Submit the .zip file

**Tips of the week!** Do ask your instructor / TA for help when necessary: we love to help and are always happy to review your code and get you unstuck!

Now, let’s get you started! Here are the two activities you need to complete. Have fun!

**Activity 1. {in pairs}** In this activity, you will have to practice programming in Prolog, mostly so that you see that there are multiple ways to implement the solution to a problem (not all solutions are in java).

**Activity 1.1.** Implement in prolog a predicate isEmpty that takes a list L. isEmpty(L) is true if L is empty, false otherwise.

*Hint: in prolog, you do not have to define what is false: you just do not mention it (a prolog file only contains true information).*

**Activity 1.2.** Implement in prolog a predicate isElement that takes a list L and an element E. isElement(E,L) returns true if E is an element of L, false otherwise.

*For instance: isElement(3,[1,2,3,4,5]) is true, but isElement(6,[1,2,3,4,5]) is not.*

**Activity 1.3.** Implement in prolog a predicate listSum that takes two parameters: a list L of integers and a number N. listSum(L,N) is true if N is equal to the sum of all elements in L, false otherwise.

*For instance: listSum([2,4,1],7) is true, but listSum([1,2,3],9) is not.*

**Activity 1.4.** Implement in prolog a predicate everyOther that takes two lists L1 and L2. everyOther(L1, L2) is true if L2 is the list made of every other element of L1, false otherwise.

*For instance: everyOther([1,2,3,4,5],[1,3,5]) is true, but everyOther([1,2,3,4,5],[1,2,3]) is not.*

**Activity 1.5.** Implement in prolog a predicate sudokuSolution that takes a Sudoku board as parameter and is true if this board satisfies the rules of Sudoku (numbers in the board are only numbers from 1 to 9, there are no duplicates in any row, column, 3x3 region) and false otherwise.

***Note:*** *you have implemented a similar method in Java in a previous lab. Please make sure to go back to that code and compare the approaches. You will see that for a given problems and a given approach to solving this problem, codes (from one language to another) can be very different.*

To be turned in:

* The pseudocode of each of the “methods” (prolog predicates) you are going to implement (in the docx file).
* The code (in your prolog file named lab12.pl) of each of the methods (predicated) as requested. In addition, for Activity 1.5., please tell us what you think about this code, compared to the Java code you wrote for the same problem a few labs ago.
* The tests you run on each of these predicates to see if your implementation is correct, and why (in the docx file).

**Activity 2. {individual}** In this activity, you have to design a Scratch video in which you are going to

1. explain and illustrate what you have learned this semester in CS1; and
2. tell us and illustrate how you picture yourself as a professional computer scientist (what type of position do you think you could see yourself in, among the many a computer scientist can take).

This video should be no shorter than 3 minutes and no longer than 8 minutes.

To be turned in:

* The script you followed for your video along with a short paragraph of the most important thing you have learned in CS1 this semester.
* The Scratch video (copy the link of your video in your docx file)