PL Homework Assignment 1:

There are two parts in this homework assignment:

Part 1. (30%) **Course Website and Definitions about Programming Languages** (web posting)

Using your athena account, create a course web site for this course (you already have a default one). For this assignment you should post:

(1) a brief self-introduction including name, email, interests and career goal. You should include one picture of yourself or your favorite things/objects.

(2) a concise description of your programming languages knowledge and interests: make 2 or more lists to let us know which languages you know well, and which languages you want to learn more.

(3) Find and give a concise definition for each of the following terms regarding programming languages and post them at your website under PL Assignment 1:

(a) syntax

(b) semantics

(c) lexical scanning

(d) parsing and parser

Note: It is very important to give references for all resources you used for the definitions displayed at your website (you can link to the URL where you found the definition or list a book or paper in which you quoted the definition).

After completing the web page, enter your URL at here. If you need some help to start your web page construction, check here.

Part 2. (70%) **Syntax Description Methods and Recursive Decent Parsing** (Canvas submission)

(1) Consider the following BNF grammar:

```
    EXP  ::= (  LIST  )  |  a
    LIST ::= LIST , EXP | EXP
```

Note that there are FOUR tokens: (,) a
and two nonterminals: EXP LIST

(a) Draw a parse tree for ((a,a),a,(a))
(b) Translate it into EBNF
(c) Draw syntax diagrams
(d) Compute First and Follow sets for each of the non-terminals

(2) Consider the following BNF grammar that we saw in class:

```
EXP    ::= EXP + TERM    | EXP - TERM    | TERM
TERM   ::= TERM * FACTOR | TERM / FACTOR | FACTOR
FACTOR ::= ( EXP ) | DIGIT
DIGIT  ::= 0 | 1 | 2 | 3
```

(a) Translate into EBNF.
(b) Draw syntax diagrams.
(c) What are the two requirements on a grammar for a predictive parser to be able to make right choice?
(d) Compute First and Follow sets for each of the non-terminals
(e) Prove that the grammar satisfy the two requirements defined in (c).

(3) **Recursive Descent Recognizer Pseudocode**

In LP Assignment 4 you are to implement a recursive-descent recognizer with a **web interface** for the grammar in (2). This will encourage you to take the opportunity to develop the web programming skills required by today's job market. In this assignment, you are only required to write **a recursive descent recognizer pseudocode,** and you will later implemente it in PL assignment 4 with web interface. The following is a set of requirements for this recognizer (parser):

- Ask the user for an input stream.
- Report "legal" or "errors found" (not both!).
- Assume the input stream is the token stream.
- Assume the input stream terminates with a $.
- Assume there is no white space.
- Use a form to collect input and return the output.
- Test your recognizer with illegal and legal strings.
- Give a brief description of this recognizer for the user on how to use and what method that it based on.

Note: check out examples in Canvas and Parsing courseware, and start learning and preparing for LP Assignment 4.