CE305 Assignment One (2020-2021)

Expression Analyser

The assignment is worth 15% of the total mark for this module. In this assignment, you are required to **specify** and **build** an expression analyser that takes arithmetic expressions. You are also required to produce both **pretty-print** and **evaluator** by using Antlr and Java. The main features of your expression analyser should support both integer number and variable, hence support assignment statement. You are also expected to add some extended features to the expression language. The **pretty-print** should display the internal representation of parsed expressions in an easily readable form. The **evaluator** should evaluate the value of given expressions. The final program should produce sensible error messages given ill-formed expressions. Finally, you are required to submit a report which describes the specification of the language, and the implementation of main and extended features.

The Specification

You are expected to provide a specification of the languages. The main features of your expression language should support integer number and variable, and also support assignment. These should include:

- The tokens of the expression language.
- The syntax of the expression language.

The Implementation

You need to produce both **pretty-print** and **evaluator** by using Antlr and Java.

- The **pretty-print** should display the internal representation of parsed expressions in an easily readable form, such as LISP-like text form (*root children*).
- The **evaluator** should evaluate the value of given expressions.
- Sensible error messages should be produced if ill-formed expressions are given.

Extended Features

- Visual rendering of Abstract Syntax Tree (AST) using tools such as java swing or graphviz or similar tools. You will be expected to select and learn about the tools by yourself.
- Handling of both integer and floating-point numbers.
- Support for multiple statements and control flow for multiple statements with Boolean expressions.

The Report

The report should be in PDF and must include a cover page with the module name, student name and ID. The source code (for the node you developed) should be made as an zip file to be submitted together with the report. The maximum length of the report is of 10 pages (include the appendix). The report should include

- A section that describes the specification.
- A section that describes the implementation.
- A section that describes how the tools were applied.

Marking Scheme
Your assignment marks are broken down into the following:

- 60% for the design and implementation of the main features
    - Definition of Tokens (25%)
    - Grammar of Expressions (25%)
    - Coding quality and/or use of tools (10%)
- 30% for design and implementation of extended features
    - If you are adding extended features, these need to be documented as per stated documentation for the main features.
- 10% for clarity of the report.

Submission:
The report should be submitted through the online coursework submission system (FASER). You should submit a single pdf file and a zip file.

Late submission and plagiarism
This assignment is to be done individually, i.e. whatever you submit must be your own individual work. Any software, figures, or any other materials that you use in this assignment, whether previously published or not, must be referred to and properly acknowledged. Please be aware that the module supervisor may ask students for an interview to explain their submitted work.