Australian National University

- COMP2100/6442
- Lectures
- Labs
- Activities
- Exams

Lab 5

COMP 2100/6442: Lab 5 - Tokenization and Parsing

In this lab we will look at tokenization and parsing, so it may be worth review the lectures and lecture notes from week 8. The tokenization part of this lab aims to just give you a bit more practice in programming. This part is a little fiddly but not particularly hard technically and conceptually. The parsing aspect aims at giving you an understanding of how to implement a recursive descent parser and also giving you a more practical understanding of recursive data structures along with recursive methods. In this lab we will also incorporate your code from Lab 4 and Lab 5 into the application (either calculator, spreadsheet, or an IDE) you have been working on. Given it will again involve Andriod studio this comes with other challenges!

Once again get together at the beginning of the lab for your lab to discuss how people are progressing in the course. Your tutor will also go over what this lab involves and some of the key ideas behind parsing.

Stage 1 (first hour) [4 marks]

In the previous lab you developed classes that enabled you to represent mathematical expression. Basically in this part of the lab you will take a textual representation of such expression, tokenize it, and then parse it into tree representation for that expression. In this part of the lab do the coding outside of android studio (so Eclipse, Intelij), just extending the code base you had form the last lab. This stage involves 4 main and required steps:

- Create a grammar for your language. This grammar must be included, using Backus-Naur form or something similar, as a comment within your code base. This grammar is important in implementing the parser.
- Implement a tokenizer for your language. This must be implemented from scratch. So you can not just use a library, like StreamTokenizer or Scanner, for this aspect.
- Implement the parser for your language. This must create the expression trees that you implemented in the previous lab. Noting, it is often possible to evaluate such expression as you parse them, also it is possible to use a library for evaluating mathematical expression, however, as the point is to learn about recursive descent parsing and recursive data structures I don’t want you to take either of these options.
- Create some test code that shows the workings of your implementation.

Stage 2 (second hour) [4 marks]

Incorporate Stage 1 along with the expression code from Lab 4 into your Android project. This should enable you to have a complete working project. To gain full marks for this stage you must tokenize, parse and generated an expression tree which can then be evaluated/executed and the result shown to the user.

Stage 3 - Challenge (third hour) [2 marks]

To take on the challenge task you need to have completed the first two stages prior to the third hour. Your tutor will allocate your challenge task when you complete Stage 2.

UPDATED: / RESPONSIBLE OFFICER: **Head of School**/ PAGE CONTACT: **Dr Eric McCreath**