**DCU School of Computing Assignment Submission**

**Student Name(s): Fawaz Alsafadi**

**Student Number(s): 15380871**

**Programme: BSc in Computer Applications**

**Project Title: Compilers Assignment 2**

**Module code: CA4003**

**Lecturer: David Sinclair**

**Project Due Date: 16/12/2019**

**Declaration**

**I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying is a grave and serious offence in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion, or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references.**

**I have not copied or paraphrased an extract of any length from any source without identifying the source and using quotation marks as appropriate. Any images, audio recordings, video or other materials have likewise been originated and produced by me or are fully acknowledged and identified.**

**This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. I have read and understood the referencing guidelines found at**

[**http://www.library.dcu.ie/citing&refguide08.pdf**](https://websvc.dcu.ie/%22http:/www.library.dcu.ie/citing%26refguide08.pdf/%22) **and/or recommended in the assignment guidelines.**

**I understand that I may be required to discuss with the module lecturer/s the contents of this submission.**

**I/me/my incorporates we/us/our in the case of group work, which is signed by all of us.**

**Signed: Fawaz Alsafadi**

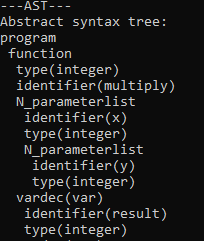
Assignment 2: Semantic Analysis and Intermediate Representation for the CCAL Language

# Abstract syntax tree

I first began by studying the notes and example provided in the course notes of a sample AST implemented using JJtree. Once I was comfortable with the syntax, I began decorating my “.jjt” file. At first the task for choosing the appropriate nodes was challenging and my AST initially had multiple unnecessary nodes and nesting issues. To resolve some of these I had to split some functions in order to visualise them better and decorate them. I added nodes into my AST by decorating with #node, I set the name for each node as appropriate. Following the course notes I also created a Terminal method called Identifier() which allowed me to set the image for ID where appropriate

To print the AST I called the dump() method in my “.jjt” file which I modified in the SimpleNode.java file as to allow me to show more information about the nodes and what their values were.

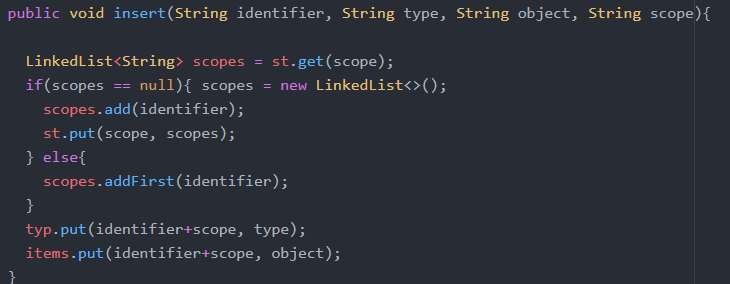
I then tested my “.jjt” file using the test files we were provided with in the first assignment below is the output.



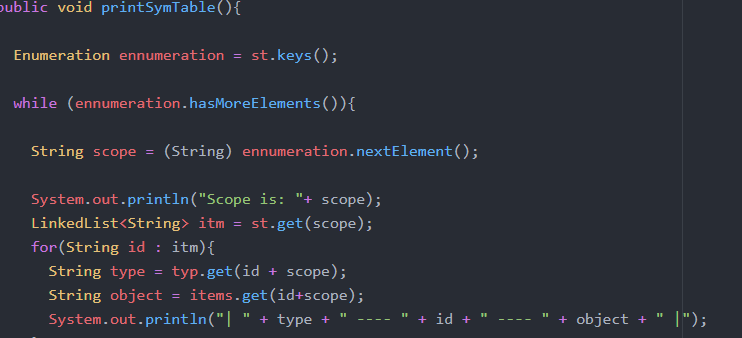
# Symbol table

Following the example from the course notes and online tutorials I implemented 3 Hash tables to store values. This was straightforward and allowed me to easily use and print the values.

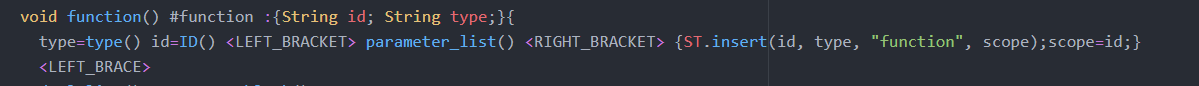
Firstly, I implemented an insert method that would allow me to insert values into the tables. This was used in my “.jjt” file and there I passed identifiers, type and importantly specified what the scope was changing to for example in a new function the scope was to be the name of that new function this is so that I adhered to the specs and made the symbol table handle scope.



To print the table, I implemented a print method to allow me to print the contents of the symbol table in the console. This worked by printing the scope and its Identifiers, types and items. The method iterated through the tables and printed the contents in a specific format.



I populated the symbol table by making calls in the “.jjt” file and supplied the necessary values. This method would assume scope is “global” unless specified otherwise at which point the scope changes to reflect the new scope in a new function for example.



# Semantic checks

I began attempting to complete the semantic checks however I struggled and was not able to finish them the file I java file I created to carry out the semantic checks is “SemanticCheckVisitor.java” it successfully implements the CcalParser visitor interface.

# Final note

I struggled greatly with this assignment and was not able to complete the semantic check aspect or generating the Intermediate code representation. Doing it again in the future I would have started earlier and attempted to better understanding the fundamentals before delving into the code as that tripped me up and I found that my compiler theory knowledge was lacking.

# References

1. Course notes

2. <https://stackoverflow.com/questions/13902239/how-to-implement-jjtree-on-grammar>

3. <https://www.tutorialspoint.com/compiler_design/compiler_design_symbol_table.htm>

4. <https://web.cs.wpi.edu/~kal/courses/compilers/JAVACC/JavaccSymobletable.htm>

5. <http://bigdatums.net/2016/07/19/how-to-create-a-hash-table-in-java-chaining-example/>

6. <https://www.youtube.com/watch?v=B4vqVDeERhI>