CSC 3301 ASSIGNMENT 1: UNDERSTANDING PROGRAM (SOURCE CODE) STRUCTURE.

Writing code and passing it through the java parser to analyze the abstract syntax tree.

## ABSTRACT SYNTAX TREE

## WHAT IS AN ABSTRACT SYNTAX TREE?

An abstract syntax tree (AST) is a language-agnostic, hierarchical representation of the elements comprising the source code of a computer program. With the use of an abstract syntax tree, it is possible to reproduce code with the same functionality of the original source code. This makes it possible to transform source code of one programming language to another programming language.

# TOKENS

## WHAT ARE TOKENS?

We have sets of rules that are used to describe a magnitude of code. Code can be broken down into simpler parts known as Lexems.

Lexems cannot be defined by these rules on a lower scope so we use lexical specification/ Lexical Analysis to describe these smaller components by looking at them individually.

These Lexems are commonly referred to as Tokens. They can be grouped as follows:

* Literals
* Identifiers
* Keywords
* Operators

The Abstract Syntax Tree is a graph of these tokens, specifically designed to preserve the context of each element and its attributes.

# OBJECTIVE

In the given Assignment, we were tasked to create an order tracker and parse it to produce an abstract syntax tree and to identify its tokens.

After the AST was retrieved, (N.B the diagram is on GitHub), the following are the tokens that were found:

Number of **identifiers** in OrderTracker.java: **17**

Number of **literals** in OrderTracker.java: **40**

Number of **operators** in OrderTracker.java: **21**

Number of **keywords** in OrderTracker.java: **27**

Number of **comments** in OrderTracker.java: **6**

**In conclusion,** the java parser proved to be a very good exercise for us as it required figuring out how it works and how to incorporate it into our code.

GROUP MEMBERS

1. GAIN NAMBEYE
2. MWIZA CHIWALE
3. THANDO TEMBO
4. TABAKAMULAMU SEBITWANE
5. GABRIEL CHILATU SINKALA
6. EMMANUEL GODFREY