**Design of Interpreter for Language of Typed Arithmetic Expressions**

This document details the steps taken to implement the interpreter for the language of Typed Arithmetic Expressions.

**Programming Language used:** Python

An outline of the steps is given here:

1. Build a lexical analyser to parse the given input string into tokens and their types.

2. Construct an Abstract Syntax Tree from the tokens obtained.

3. Perform type checking on the AST.

4. If there are no type errors, evaluate the expression using the evaluation rules for the language.

5. If there are type errors, report “Type Error”.

**Step 1- Lexical Analyser**