COM-3645 Compilers and Tools
Compiler Semester Project

A. Define your own Toy language

For your project, you will define your own toy programming language. You are free to choose and define the syntax and semantics of your language. Your language many be based on an existing language, or it may derive from a combination of multiple existing languages; however, it cannot be an exact simple subset of an existing language.

At the minimum, your language must include the following constructs:

1) Data Types

a) Primitive types, along with type checking, covering: integer/real/string/boolean values
b) The ability to cast or convert between primitive types
c) User-defined record types (e.g., struct in C, record in Pascal)
d) Constants (of primitive types)
e) Mutable variables
f) Arrays of the above types

2) Operators and Expressions

a) Basic arithmetic expressions with operator precedence and associativity, covering the following operators:
   - Addition / Subtraction
   - Multiplication / Division
   - Unary negation operator
   - String concatenation

b) Basic relational expressions with operator precedence and associativity, covering the following operators:
   - Equals / Greater than / less than

c) Basic logic expressions with operator precedence and associativity, covering the following operators:
   - AND / OR / NOT

3) Language Constructs

a) Assignment statement
b) Selection: decision branching; choose between 2 or more alternative paths
c) Repetition: looping; i.e., repeating a piece of code multiple times in a row
d) Function and Procedures calls
e) Command to print formatted text to standard output
f) The ability to accept and read program command-line arguments
g) The ability to add comments

Deliverable 1: [5 points] (03/28/2023)

The deliverable for this phase is a valid ANTLR4 g4 grammar file, along with a valid source text file for your language that can be tokenized and parsed. Parse the source text file using the ANTLR4 tools set.

Note, you should provide examples that cover each of the language construct, operators and data types.

Toy.g4	*– the grammar file for your toy language*
Deliverable1.toy	*– sample source file to parse*
Deliverable1.tree	*– the parse tree created from parsing your source file*

<u>Deliverable 2</u>: [10 points] (04/06/2023)

The deliverable for this phase is to build and run a semantic type checker that will type check your program and raise the following issues:

```
UNDECLARED_IDENTIFIER       ("Undeclared identifier"),
REDECLARED_IDENTIFIER       ("Redeclared identifier"),
INVALID_SIGN                ("Invalid sign"),
INVALID_TYPE                ("Invalid type"),
INVALID_VARIABLE            ("Invalid variable"),
TYPE_MISMATCH               ("Mismatched datatype"),
TYPE_MUST_BE_INTEGER        ("Datatype must be integer"),
TYPE_MUST_BE_NUMERIC        ("Datatype must be integer or real"),
TYPE_MUST_BE_BOOLEAN        ("Datatype must be boolean"),
INCOMPATIBLE_ASSIGNMENT     ("Incompatible assignment"),
INCOMPATIBLE_COMPARISON     ("Incompatible comparison"),
DUPLICATE_CASE_CONSTANT     ("Duplicate CASE constant"),
INVALID_CONTROL_VARIABLE    ("Invalid control variable datatype"),
NAME_MUST_BE_PROCEDURE      ("Must be a procedure name"),
NAME_MUST_BE_FUNCTION       ("Must be a function name"),
ARGUMENT_COUNT_MISMATCH     ("Invalid number of arguments"),
ARGUMENT_MUST_BE_VARIABLE   ("Argument must be a variable"),
INVALID_RETURN_TYPE         ("Invalid function return type"),
TOO_MANY_SUBSCRIPTS         ("Too many subscripts"),
INVALID_FIELD               ("Invalid field");
```