**CS 4308 – Concepts of Programming Languages** **Project**

**3 Deliverables (300 pts Total – 100 pts Per Deliverable)**

The project for this course is the development of an interpreter of a language implemented in your choice of programming languages. You may use the same language for each part, or different languages.

**The Interpreter Project**

### This project consists of **developing an interpreter** in 3 parts: lexical analyzer (scanner), syntax analyzer (parser) and interpreter. The **interpreter** isfor a **minimal form of the Julia language (see JuliaLang.org)**. This minimal form of Julia has only 1 data type, integer, and the only identifiers are single letters. Additional information about the Julia language is below.

The interpreter will process a Julia file and build some intermediate data structures. These data structures will then be interpreted to execute the program. All tokens in this language are separated by white space. The parsing algorithm should detect any syntactical or semantic error. The first such error discovered should cause an appropriate error message to be printed, and then the interpreter should terminate. Run-time errors should also be detected with appropriate error messages being printed.

**Deliverables (see D2L Assignments for due dates):**

1. **Deliverable 1 Scanner – (100 points) Due Oct 10, 2021**

Develop a complete scanner. Write a short report describing the work performed, include the source code, and show the input and output. You must show the execution of this program by using appropriate input files and the program must show a list of the tokens scanned; provide screenshots in addition to your source code.

1. **Deliverable 2 Parser – (100 points) Due Oct 24, 2021**

Develop a complete parser that executes with the scanner. You must show the execution of this program by using several relevant source lines as input, the program must show the corresponding statement recognized. Write a short report describing the work performed. Include the source program, input and output.

3.  **Deliverable 3 Interpreter – (100 points) Due Nov 14, 2021**

Develop a complete interpreter that includes the scanner and parser. You must show the execution of this program by using a relevant source line as input, the program must show the results after executing the statement recognized by the parser. Write a short report describing the work performed. Include the source program, input and screenshots of output.

## **Grammar for the language**

**Syntax Analyzer**

<program> → function id ( ) <block> end

<block> → <statement> | <statement> <block>

<statement> → <if\_statement> | <assignment\_statement> | <while\_statement> |

<print\_statement> | <repeat\_statement>

<if\_statement> → if <boolean\_expression> then <block> else <block> end

<while\_statement> → while <boolean\_expression> do <block> end

<assignment\_statement> -> id <assignment\_operator> <arithmetic\_expression>

<repeat\_statement> -> repeat <block> until <boolean\_expression>

<print\_statement> → print ( <arithmetic\_expression> )

<boolean\_expression> → <relative\_op> <arithmetic\_expression> <arithmetic\_expression>

<relative\_op> → le\_operator | lt\_operator | ge\_operator | gt\_operator | eq\_operator | ne\_operator

<arithmetic\_expression> → <id> | <literal\_integer> | <arithmetic\_op> <arithmetic\_expression>

<arithmetic\_expression>

<arithmetic\_op> → add\_operator | sub\_operator | mul\_operator | div\_operator

**Lexical Analyzer**

id → letter

literal\_integer → digit literal\_integer | digit assignment\_operator → =

le\_operator → <= lt\_operator → < ge\_operator → >=

gt\_operator → > eq\_operator → = = ne\_operator → ~= add\_operator → + sub\_operator → - mul\_operator → \* div\_operator → /