Group 14 Final Project Write-up

GitHub Repo: <https://github.com/farreled/CS3003Summer2021FinalProject>

YouTube Video: <https://youtu.be/5d30PlA6IQk>

Project Responsibilities:

Ned Farrell (farreled)

* Binary operator (ding, +\*)

Armen Krikorian (krikorad)

* New control structure (until), final video composition and posting

Will Schreiner (schreiwa)

* test.cpp and it’s testing

Kevin Slyh (slyhkt)

* New data type (double)

How we changed CLite:

The biggest changes we made to CLite was the implementation of double, until, and ding. Each of these new features were added to the Parser.java and Lexer.java logic to allow the language to recognize these keywords. Had we continued to implement more logic for these features, the language would be able to semantically analyze these keywords and apply additional logic to implement the correct procedure. In addition to adding to Parser.java and Lexer.java, we also had to add logic to AbstractSyntax.java, Token.java, and TokenType.java that both Parser.java and Lexer.java could reference when it encountered these new keywords. This allowed Parser.java and Lexer.java to correctly identify the keywords and label them accordingly.

Documentation Additions

A.1 Lexical and Concrete Syntax of CLite

Type -> int | bool | float | char | double

Literal -> Integer | Boolean | Float | Char | Double

Float -> Integer . Integer f

Double -> Integer . Integer

AddOp -> + | - | +\*

UntilStatement -> until ( Expression ) Statement

A.2 Abstract Syntax of CLite

Type -> int | bool | float | char | double

ArithmeticOp = + | - | \* | / | +\*

DoubleValue = Double doubleValue

A.3 Type System of CLite

(Our implementation of double falls in the “otherwise” clause in rule A.4, #2, section (e))

Type Rule A.6, #3, section (a): If the Expression is a Binary, then: (a) If the Operator is arithmetic ( +.,-, \*, **+\*** or /), then its terms…

A.4 Semantics of CLite

Meaning Rule A.8, add in: if the operator is a double operator; then double arithmetic using the IEEE standard is performed on the double operands, resulting in a double result.

Test Code (test.cpp):

int main() {

// Variable declarations

int intVal;

float floatValA, floatValB, addresult;

double doubleValA, doubleValB, doubleValC, subresult, divMultResult;

// Variable assignment

intVal = 5;

floatValA = 0.7f;

floatValB = 0.3f;

doubleValA = 0.05;

doubleValB = 0.07;

// Basic arithmatic tests

addresult = floatValA + floatValB;

subresult = doubleValB - doubleValA;

divMultResult = doubleValA \* doubleValB;

divMultResult = doubleValA / doubleValB;

doubleValC = doubleValA +\* doubleValB;

until (intVal >= 10) {

intVal = intVal + 1;

}

}

Screenshot/Demonstration of Test Code

Lexer:

![Graphical user interface, application

Description automatically generated]()![Graphical user interface, application

Description automatically generated]()

Parser:![Graphical user interface

Description automatically generated with low confidence]()

![Graphical user interface

Description automatically generated with low confidence]()![Text

Description automatically generated]()