Andrew King & Haochi Du

CSC 402

Final Project Proposal

12 November 2021

**Overview**

This will be a general purpose, functional programming language. Our goal in designing this language is to reduce the unnecessary key words and make the program more readable and simple to interpret. Variables are statically-typed, and the type must be specified in the variable declaration. Programs execute from top to bottom, and there is no default driver function (such as the “main” function in C).

Our language will support basic logical and mathematical operations as well as input and output. It will also support control flow structures like if statements, for loops, and while loops. Functions, like variables, are statically typed and the type is specified in the function declaration. When a function is called, actual parameters are assigned to formal parameters positionally. Our language is statically scoped, and uses scoping rules similar to those in C: when a variable is being accessed, the program starts at the current scope and traverses upward along the stack of scopes until it finds the given identifier.

In addition to common data types like int, str, bool, float, and char, our language also supports a data type called fract, which stands for fraction. The fract data type allows programmers to store division expressions as values without actually computing the expressions. This reduces the loss of precision in float values and also allows the programmer to work more naturally with non-metric counting systems such as imperial measurements and musical time signatures. Our language also supports arrays, which can hold only the data type specified in the array declaration.

**Language Example**

# this is a comment

func int foo\_add(int a, int b)

return a + b

# declare some variables and assign values

int x <- 10

int y <- 8

int z <- foo\_add(x, y)

out z

# loop z from 1 to 10 (inclusive), increment by 2

for z <- 1:10:2

z <- foo\_add(z, 1)

out “z = “ z

**Milestone Schedule, Team Members & Tasks**

| **stage** | **tasks** | **Date** | **Assigned** |
| --- | --- | --- | --- |
| **Prep** | Define the grammar | 11/19 | Andy |
| **Build** | Lexer | 11/26 | Andy |
|  | State structure | 11/26 | Haochi |
|  | Interpreter | 12/3 | Haochi & Andy |
|  | Program walker | 12/3 | Haochi & Andy |
| **Testing** | Make example programs | 12/3 | Haochi |
|  | Debugging & improvement | 12/8 | Haochi & Andy |
|  | Est. Due Date | 12/10 |  |

**Resources Required**

Software: Python, IDE (PyCharm & VisualStudio Code), Hardware: Macbook