Description of Minimal Dialect

Include a control flow construct for definite iteration, like a for loop.

- for (*assignment*; *comparison*; *increment* or *decrement*) { //code block }
- for (int i = 0; i < 10; i++) { // code block }
- Assignment is executed once before the code block is executed. Comparison defines the condition for executing the code block. Increment is executed each time after the code block has been run.

Include a control flow construct for indefinite iteration, like a while loop.

- while (*comparison*) { //code block }
- The code block is executed each time that the condition is true. Once the condition is false, the code block is no longer executed.
- Concrete example:

```
while(i < 5){
        i++;
}
```

Include a control flow construct for conditional branching, like an if/else/else-if.

- if (*comparison*) { //code block 1 (run if true) }
- else { //code block 2 (run if false) }
- If the condition in the if statement is true, the first code block will execute once. If that condition is false, the second code block will execute.
- Concrete example:

```
if (i < 10){
        i++;
}
 else {
        i = i+3;
}
```

Include variable declarations and assignment statements. **This section defined assignment statements.**

- [type] *variableName* = [value];
- The type that *variableName* is declared as must match the data type of the corresponding value that is provided for the variable initialization.
- Concrete example
    - int i = 5;

Include data types for integer variables and floating point variables.

- int [1 or more numeric characters]
- float [1 or more numeric characters]**.**[1 or more numeric characters]
- Concrete example:
    - int a = 579;

        - float r = 57.9;

Include arithmetic operators for addition, subtraction, multiplication, and division.

- Addition: +
    - Abstract Example: [int or float] + [int or float]
    - Concrete Example: 61 + 46
    - [int a = 61;
      int b = 46;
      int z = a + b]
- Subtraction: -
    - Abstract Example: [int or float] - [int or float]
    - Concrete Example: 98 - 22
- Multiplication: *
    - Abstract Example: [int or float] * [int or float]
    - Concrete Example: 16 * 16
- Division: /
    - Abstract Example: [int or float] / [int or float]
    - Concrete Example: 65 / 5

Include numeric comparison operators for equal, unequal, lesser, lesser or equal, greater, and greater or equal.

- Comparison operators are used to evaluate integer, float, or variable values in comparison statements.
- Equal: ==
    - Abstract Example: [int or float or variable] == [int or float or variable]
    - Concrete Example: 8 == 8.0
- Unequal: !=
    - Abstract Example: [int or float or variable] != [int or float or variable]
    - Concrete Example: 6 != 19
- Lesser: <
    - Abstract Example: [int or float or variable] < [int or float or variable]
    - Concrete Example: 8 < variable1
- Lesser or Equal: <=
    - Abstract Example: [int or float or variable] <= [int or float or variable]
    - Concrete Example: 7.8 <= 7.9
- Greater: >
    - Abstract Example: [int or float or variable] > [int or float or variable]
    - Concrete Example: 89 > 76
- Greater or Equal: >=
    - Abstract Example: [int or float or variable] >= [int or float or variable]
    - Concrete Example: variable2 >= 9.0

Support conventional precedence and associativity for arithmetic operators and grouping symbols like parentheses.

- Left to Right:()
    - Example: (a*b)c = *a*(b*c)
    - This indicates associativity applies to the placement of the parentheses, not the values. In which, precedence is an attribute to which operation would be performed first.
- Order of Operator Precedence:
    - Precedence 1 (Highest Precedence): ()
    - Precedence 2: *, /
    - Precedence 3: +, -
    - Precedence 4: <, <=, >, >=
    - Precedence 5: ==, !=
    - Precedence 6 (Lowest Precedence): =
    - Note that the precedence of the operations is represented on a numeric scale where 1 represents the highest precedence.

Statement Types Supported:

- Comparison:
    - Abstract Example: [identifier or numeric] [comparison_op] [identifier or numeric]
    - Concrete Example: index <= 100
- Increment:
    - Abstract Example: [identifier] [++]
    - Concrete Example: i++
- Decrement:
    - Abstract Example: [identifier] [--]
    - Concrete Example: i--