Mini Compiler – Language Specification

Version: 1.0

Author: Sonali

Implementation Language: C++

Date: 30 Jan 2025

1. Introduction

This document defines the language rules for our Mini Compiler project.The language is a small C-style language supporting:

-Integer variables

-Assignment statements

-Arithmetic expressions

-Print statements

-If statements (without else)

-While loops (with equality check)

The goal is to keep the language simple and minimal, so that the compiler frontend (Lexer + Parser) and backend can be implemented clearly in C++, using basic standard libraries.

Note:The keyword print is a custom built-in function of our mini language (different from C's printf and c++'s cout).

2. Lexical Elements (Tokens):

2.1 Keywords

-Text
-int
-print

-if
-while

These words are reserved and cannot be used as variable names.

2.2 Identifiers (ID)

- Begin with a letter (a–z or A–Z)
- May contain letters or digits
- No spaces, no symbols

Examples:
a, x, num1, total

2.3 Numbers (NUM)

- Only integer constants allowed
- No decimal points
- No negative literals (use x = x - 1 instead)

Examples:
0, 4, 99, 1000

2.4 Operators:

-Arithmetic Operators:

+

-

*

/

-Assignment Operator:

=

-Comparison Operator:
==

These operators will be recognized by the Lexer and validated by the Parser.

2.5 Symbols:

```
;
(
)
{
}
```

These symbols mark statement boundaries and program structure.

2.6 Whitespace:

- Spaces, tabs, and newlines are allowed.
- They are ignored except to separate tokens.
- Extra whitespace does not affect the meaning of the program.

3. Supported Statements

3.1 Variable Declaration

Syntax:

Cpp

```
int <identifier>;
```

Examples:

Cpp

```
int a;
int value;
int num1;
```

3.2 Assignment Statement

Syntax:

Cpp

```
<identifier> = <expression>;
```

Examples:

Cpp

```
a = 5;
x = a + 3;
total = value * 2;
```

3.3 Print Statement

Syntax:

Cpp

```
print(<identifier>);
```

Example:

Cpp

```
print(a);
```

(Remember: print is our own language keyword, not actual C++.)

3.4 If Statement (without else)

Syntax:

Cpp

```
if (<identifier> == <expression>) {
<statements>
}
```

Example:

Cpp

```
if (a == 5) {
print(a);
}
```

3.5 While Loop

Syntax:

Cpp

```
while (<identifier> == <expression>) {
```

```
<statements>
}
```

Example:

Cpp

```
while (x == 0) {
x = x + 1;
}
```

4. Expressions

Expressions may include:

- Numbers
- Identifiers
- Arithmetic combinations

Examples:

Cpp

```
5
a
a + 3
a * (b + 2)
```

5. Sample Programs

⭐ Program 1 — Basic Declaration, Assignment & Print:

Cpp

```
int a;
a = 5;
print(a);
```

⭐ Program 2 — Arithmetic with Two Variables:

Cpp

```
int a;
int b;
a = 5;
b = a + 3;
print(b);
```

⭐ Program 3 — While Loop:

```
Cpp
int x;
x = 0;
while (x == 0) {
x = x + 1;
}
print(x);
```

⭐ Program 4 — Invalid Program (for Error Testing)

```
Cpp
a = 5;      // Error: 'a' not declared
print(a);   // Error: cannot print undeclared variable
```

6. End of Specification

This document defines the complete lexical and syntactic rules of the Mini Language.
The compiler frontend (Lexer + Parser) and backend will strictly follow this specification while being implemented in C++.