Alpha Programming Language

User Manual

1. Introduction

This document presents the Alpha programming language, a minimal and educational language developed as part of my Compiler Design coursework. Alpha is designed to be simple, readable, and easy to implement, allowing us to explore the core concepts of compiler construction such as lexical analysis, parsing, semantic rules, and interpretation. The language currently supports variable declarations, assignments, arithmetic expressions, input handling, output statements, and both forms of comments.

Alpha programs are executed through the compiler built using Flex and Bison. This manual provides all information necessary to write and run valid Alpha programs.

2. Getting Started

Alpha source code files use the extension .alpha.
To run a program, execute the compiler from the terminal as follows:

```
./alpha filename.alpha
```

The compiler evaluates statements sequentially and prints outputs as they occur. At the end of execution, the symbol table is displayed for reference.

3. Basic Syntax

- Alpha is case sensitive.
- Whitespace is ignored except inside text literals.
- Each statement must appear on a separate line.
- Keywords are reserved and must not be used as identifiers.

- Reserved keywords in Alpha

num

decimal

text

read

Show

4. Variables

Variables in Alpha must be declared before use. The language supports three types.

- num Integer values
- decimal Floating point values
- text String values

Examples of declarations:

- num a
- decimal b
- text msg

Assignments update previously declared variables:

a = 10

b = 3.14

msg = "HelloAlpha"

Assignments always replace the existing value of the variable.

5. Input and Output

Alpha supports basic input and output operations for user interaction.

Input

read a

Reads a value from the user and stores it in variable a. The value must match the type of the variable.

Output

show(a)

show("message")

Displays either a variable value or a text literal.

6. Comments

Alpha provides two comment systems for documentation.

Single line comment Begins with

alp>

and continues until the end of the line.

Multi line comment Begins with

alp{

and ends with

}alp

Examples:

```
alp> This is a single line comment


alp{
This is a multi line
comment example
}alp
```

Comments are completely ignored during compilation.

7. Arithmetic Expressions

Alpha supports basic arithmetic operators on numeric types.

Addition +
Subtraction -
Multiplication *
Division /

Examples:

```
a = a + 5
b = b * 2.0
b = b / 3.0
```

Only one arithmetic expression is permitted per assignment.

8. Example Program

The following example demonstrates all features currently available in Alpha.

```
num a
decimal b
text c

show("Initial assignments")
a = 10
b = 3.14
c = "HelloAlpha"

show(a)
show(b)
show(c)

a = a + 5
b = 6.28
c = "Updated"

show(a)
show(b)
show(c)

read(a)
read(b)
read(c)

show(a)
show(b)
show(c)

alp> Single line comment example

alp{
Multi line
comment example
}alp
```

9. Conclusion

Alpha is a compact language intentionally designed to highlight the essential components of compiler design. Through its simple syntax and limited but complete feature set, it provides an effective environment for demonstrating tokenization, parsing rules, semantic checks, and runtime evaluation. This manual summarizes all implemented features and serves as a guide for writing correct programs in Alpha.

10.Team Members

- Md Golam Sharoar Saymum 242220005101780
- Suriya Sharmin Mim 242220005101719
- Samia Islam 242220005101694
- Rifah Tashfia 242220005101709
- Muhsana Rajjak Rima 242220005101729