**USER’S MANUAL**

**Language Design:**

1)My language is based on C++. I will design the compiler in C++.

* Language supports only one dimensional array. Array size has to be declared initially. Cannot increase dynamically.
* Only functions are supported

2)Keywords are listed below. Keywords are reserved

while for AND

if else main

int void string

cout cin OR

3)The language is case-sensitive ; Ident, ident, and IDENT are the different identifier. Keywords are case sensitive since my language is based on C++( which is case sensitive)

Example: IF is not keyword

4)Identifiers are sequences of letters and digits. They must begin with a letter. The maximum length of a lexical element is 255 characters. Only the first 8 characters are significant

identifier ::= letter {letter\_or\_digit}

letter\_or\_digit ::= letter | digit

5)The numbers in the language are integers and float

Float=integer.integer

integer = digit+ (e.g. 23450)

digit=0|1|2|3|…|9

6)Strings are a combination of [A-Za-z] alphabets

string=character+

7) Blanks are delimiters. They are allowed but not required tokens

8)

Tokens are (listed in logical groups) LEXEME(S)

identifiers let(let|dig)\*

numeric literals integer, float

string literals " (any char)\* “

const token 'const'

data type tokens 'int' float ‘string’

operators: + - \* /

3 relational operators < > ==

semicolon ;

assignment =

left parentheses (

rt. parenthesis )

left curly brace {

left curly brace }

9) Since blanks are delimiters, they may not appear within a token (eg '= =' though this is legal, is reported as two tokens ':' and '=').

10) The basic character set is sufficient for writing any program. The characters included in each of the categories of basic graphic characters are defined as follows:

upper case letters

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

1. digits

0 1 2 3 4 5 6 7 8 9

1. special characters

" # & ' ( ) \* + , - . / : ; < = > \_ |

1. the space character

11)An array object is a composite object consisting of components that have the same subtype.

An array object is characterized by the number of indices (the dimensionality of the array), the type and position of each index, the lower and upper bounds for each index, and the type and possible constraint of the components. The order of the indices is significant.

A one-dimensional array has a distinct component for each possible index value

12) Functions can take only 3 formal parameters and can only return one data type (integer value).

13) Aritmetic operations are allowed only for Integer Values.

**Execution Steps:**

Unzip all the BabyC++Compiler.zip.

Run the below commands from command prompt

flex lex.l

bison -d compiler.y

g++ -o indu bison.tab.c compiler.yy.c HashMap.cpp LinkedHash.cpp ActiveBlocksStack.cpp Codegen.cpp Codegen.h

./indu<test.txt (Test.txt is file to be tested to see if it gets compiled by BabyC++Compiler)

After running these commands, CodeOutput.s file is generated. Open this file in SPIM Mars and execute it.

**LIMITATIONS :**

CodeOutput.s file is generated if Arithmetic Operations are performed only for Integer Values

Comments are not allowed in the program.

Codegenerated using MIPS is not optimized.

Only assignment of string values is allowed. Functions can have only <=3 formal parameters.

**Error Messages:**

1. “Multiple declaration of identifier” : This means a variable is declared multiple times in the same scope. This is non fatal error.
2. “Undeclared identifier” : This means a variable has not been declared in either local or global scope
3. Syntax Error: This means that input file is not in the Grammar accepted by Compiler. Look at the grammar and correct it.

4)CodeOutput.s: This file is not generated if there Data types of variables compared donot match