**PROJECT ABSTRACT**

**CS6109 - COMPILER DESIGN**

**Vivek Ramkumar, 2018103082**

**Kariketi Tharun Reddy, 2018103034**

**G. R. Srikanth, 2018103603**

**Objective:**

To write a compiler that converts a high-level language, namely Java, into machine code by following the core principles of compiler design. Compiler construction, in its entirety, takes into account thousands upon thousands of patterns from a source language, in order to achieve high flexibility in programming. We provide a miniature version, which understands a limited set of instructions, but nonetheless just as powerful.

**Introduction:**

**Functionalities:**

**Conclusion:**

**Team members:**

**Srikanth**

**Tharun**

**Vivek**

**This project has basic functionalities for number manipulation and boolean comparisons and can perform complex calculations with multiple parenthesis.**

##### **Main arithmetic operations.**

The main arithmetic operations available are +, -, \*, /, %.

##### **Conditional statements.**

This programming language supports various conditional operations like <, >, ==. <=, >=. It also supports assigning direct boolean values in the form of keywords 'True' and 'False'.

##### **Variables.**

The variables can be assigned from [a-z] and [A-Z]. The symbol table implementation involves a simple symbol table of 52 rows. And a simple hashing algorithm to fill in according to the ascending value of the variables is used. There is only one data type, i.e integer. The result of a conditional statement can also be stored inside a variable.

##### **Printing the result.**

The results can be print using the 'print' statement followed by an expression or an identifier.

##### **Sample program**

a=10+5/5+5+2\*1 != 8;

b = 5%2;

c = a==b;

print c;

print 500;

##### **Output (Evaluation)**

1

500

##### **Three address code generation.**

The three address code generation is maintained by using the following structure,

struct threeADD

{

char result[10];

char operand\_1[10];

char operand\_2[10];

char operator[10];

} quadraple[50];

int index;

This is updated with every valid line encountered and it is represented as the result with a prefix \_k and numbered from 0.

##### **Sample program**

a=10+5/5+5+2\*1 != 8;

b = 5%2;

c = a==b;

print c;

print 500;

##### **Output (Evaluation)**

1

500

##### **Output (Three address code)**

\_k0 := 5 / 5

\_k1 := 10 + \_k0

\_k2 := \_k1 + 5

\_k3 := 2 \* 1

\_k4 := \_k2 + \_k3

\_k5 := \_k4 != 8

a := \_k5

\_k7 := 5 % 2

b := \_k7

\_k9 := a == b

c := \_k9

Cprint c

Cprint 500