



JINKA UNIVERSITY

COLLEGE OF NATURAL AND COMPUTATIONAL SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

COURSE TITLE:COMPILER DESIGN

COURSE CODE: CoSc

Name	ID_number
1. Elias Tadele	ugr/13776/14
2. Emkulu Teleale	ugr/13788/14
3. Anteneh Tekuash	uga/0174/14
4. Yadeta Yigazu	ugr/15480/14
5. Haymanot Sewbihon	ugr/14055/14
6. Yaikob Hiskel	ugr/15500/14
7. Abeje Ashenefi	ugr/13145/14
8. Rediat Demile	ugr/14744/14
9. Kasim Logita	ugr/14158/14

Submitted to Mr. Temesgen

27,Dec-2024/5
Jinka ,Ethiopia

SAMPLE C CODE

```
int main() {
    int x = 10;
    for (int i = 0; i < x; i++) {
        if (i % 2 == 0) {
            printf("%d is even\\n", i);
        }
    }
    return 10;
}
```

Saved by Sample_C_code.c

Using Flex for Lexical Analysis

Flex Code (Group_3.1):

```
%{
#include <stdio.h>
int yywrap(void) { return 1; } // Correct yywrap definition
}%

%%
"if"      { printf("Keyword: if\\n"); }
"for"     { printf("Keyword: for\\n"); }
"int"     { printf("Keyword: int\\n"); }
"return"  { printf("Keyword: return\\n"); }
[0-9]+    { printf("Literal: %s\\n", yytext); }
[a-zA-Z_][a-zA-Z0-9_]* { printf("Identifier: %s\\n", yytext); }
"=="|"!="|"<="|">="|"<|">" { printf("Operator: %s\\n", yytext); }
"="|"+"|"-"|"*"|"/"      { printf("Operator: %s\\n", yytext); }
";"|"|"(")"|"{"|"}"      { printf("Punctuation: %s\\n", yytext); }
[\\t\\n]+  { /* Ignore whitespace */ }
.          { printf("Unknown token: %s\\n", yytext); }

%%

int main() {
    yylex();
    return 0;
}
```

Expected Output

Keyword: int
Identifier: main
Punctuation: (
Punctuation:)
Punctuation: {
Keyword: int
Identifier: x
Operator: =
Punctuation: ;
Operator: <
Identifier: x
Punctuation: ;
Operator: i
Operator: ++
Punctuation:)
Punctuation: {
Keyword: if
Punctuation: (
Identifier: i
Operator: %
Literal: 2
Operator: ==
Literal: 0
Punctuation:)
Punctuation: {
Identifier: printf
Punctuation: (
String: "%d is even\\n"
Punctuation: ,
Identifier: i
Punctuation:)
Punctuation: ;
Punctuation: }
Punctuation: }
Keyword: return
Literal: 10
Punctuation: ;
Punctuation: }

Literal: 10
Punctuation: ;
Keyword: for
Punctuation: (
Keyword: int
Identifier: i
Operator: =
Literal: 0

Step to Run the code In VS code

1 Open Terminal in VS Code:

Open the terminal in VS Code by selecting Terminal > New Terminal.

2.Generate Lexical Analyzer using Flex:

In the terminal, navigate to the directory where GROUP_3.1 file is located

Run the following command to generate the C file for your lexical analyzer from the Flex source code:

Win_flex group_3.1

3.Compile the Generated C Code:

Compile the generated C code using the **gcc** compiler to produce an executable

gcc lex.yy.c -o GROUP

4. Run the Lexical Analyzer:

By Copy and paste the source code of C