```
Name: Rohit Saini
Roll No: PC41
Erp: 1032200897
Code:
%{
#include <stdio.h>
#include <string.h>
int count;
%}
%option noyywrap
letter [a-zA-Z]
digit [0-9]
space [' ']
identifier ({letter}({letter}|{digit})*)
classname (class{space}({letter}|{digit})*)
quote \"
datatype (int|char|float|double|void)
keywords
(void|class|throws|import|try|catch|finally|static|implements|extends|main
booleanvalues (true|false)
loops (while|for)
conditionalStatements (if|else|switch)
string ({quote}({letter}|{digit})*{quote})
array ({identifier}(\[\d+\])*)
parenthesis ([{}();,])
comment "//".*
%%
{space};
[ \t\n]; // Ignore whitespace
```

```
"import"[^\n]*";" { printf("Preprocessor Directive: %s\n", yytext); }
"public|private|protected" { printf("Access Specifier: %s\n", yytext); }
                { printf("Integer: %s\n",yytext);}
{digit}
                { printf("Keyword: %s\n", yytext); }
{keywords}
                { printf("Datatype: %s\n", yytext); }
{datatype}
{booleanvalues} { printf("Boolean Value: %s\n", yytext);}
                { printf("Parenthesis: %s\n", yytext);}
{parenthesis}
                { printf("Identifier: %s\n", yytext);}
{identifier}
{classname}
                { printf("Classname: %s\n", yytext);}
                { printf("Array: %s\n", yytext);}
{array}
                { printf("String: %s\n", yytext);}
{string}
                { printf("Quote: %s\n", yytext);}
{quote}
                { printf("Comment: %s\n", yytext);}
{comment}
"&&|\|\||!|=" { printf("Logical Operator: %s\n", yytext); }
"<|>|<=|>=|==" { printf("Relational Operator: %s\n", yytext); }
"\+\+|--" { printf("Unary Operator: %s\n", yytext); }
"[+\-/%\*]" { printf("Arithmetic Operator: %s\n", yytext); }
"=" { printf("Assignment Operator: %s\n", yytext); }
. { printf("Unknown Token: %s\n", yytext); }
%%
int main() {
   yyin = fopen("input.java", "r");
   yylex();
   fclose(yyin);
   return 0;
}
```

Output:

```
rohit@DESKTOP-3DK43OM MINGW64 ~/Documents/GitHub/sem_7/ssc/lab5 (main)
$ wsl
 _$ ./a.out
Identifier: public
Classname: class input
Parenthesis: {
Unknown Token:
Identifier: public
Keyword: static
Keyword: void
Keyword: main
Parenthesis: (
Identifier: String
Unknown Token: [
Unknown Token: ]
Identifier: args
Parenthesis: )
Parenthesis: {
Unknown Token:
Comment: // This is a comment
Identifier: System
Unknown Token: .
Identifier: out
Unknown Token: .
Identifier: println
Parenthesis: (
Quote: "
Identifier: Hello
Parenthesis: ,
Identifier: World
Unknown Token: !
Quote: "
Parenthesis: )
Parenthesis: ;
Unknown Token:
Datatype: int
Identifier: num1
```