

Code :

```
%{
#include<stdio.h>
#include<string.h>
typedef struct node {
char ID[10], Type[10];
struct node *next;
}node;
node *head = NULL, *curr=NULL, *prev=NULL;
int declare_flag = 0;
}%

delim [ \t\n]
ws {delim}+
letter [a-zA-Z]
digit [0-9]
package {letter}+\.{letter}+(\.{letter}+[*]?) *
Id {letter}({letter}|{digit}|(\_))*
Number {digit}+(\.{digit}+)?(E[+-]?{digit}+)?
anychar ({letter}|{digit}|[ !@#$%^&*()_.,?:])*
String \"{anychar}\"

%%

{ws}
"System.out.println"| "System.out.print" {printf("\n%s : Print
Statement",yytext);}
{package} {printf("\n%s : Package",yytext);}
class+" "+{letter}+ {printf("\n%s : Class",yytext);}
import|if|else {printf("\n%s : Keyword",yytext);}
{String} {printf("\n%s : String",yytext);}
```

```

"+"| "-"| "*"| "/"| "=" {printf("\n%s : Operator",yytext);}
{Number} {printf("\n%s : Number",yytext);}
"("| ")"| "{"| "}"| ";"| ":" {printf("\n%s : Punctuation",yytext);}
int|float|char|double {
printf("\n%s : Data Type",yytext);
prev = curr;
curr = curr->next;
curr = (node*)malloc(sizeof(node));
prev->next=curr;
strcpy(curr->Type, yytext);
declare_flag=1;}
{Id} {
printf("\n%s : Identifier",yytext);
if(declare_flag==1)
{
strcpy(curr->ID, yytext);
declare_flag=0;
}}
"*/"
"//"+{anychar}*"\n"| "/*"+{anychar}*"/" {printf("\n%s :
Comment",yytext);}
%%
int main()
{
head = (node*)malloc(sizeof(node));
curr = head;
yyin=fopen("Java_Sample.txt", "r");
yylex();
printf("\n\nSymbol Table:\nID\tType\n");
node *temp = head->next;
while(temp!= NULL)
{

```

```
printf("%s\t%s\n", temp->ID, temp->Type);  
temp=temp->next;  
}  
}  
int yywrap()  
{  
return 1;  
}
```

Input (Java Code Txt File) :

```
import java.io.*  
class main  
{  
//This is a comment  
int a=1;  
int b=2;  
int c;  
c=a+b;  
System.out.println("Hello World: " +c);  
}
```

Output :

```
(base) digvijay@Digvijays-MacBook-Air LCA 05 % lex Code05.1
(base) digvijay@Digvijays-MacBook-Air LCA 05 % cc lex.yy.c
(base) digvijay@Digvijays-MacBook-Air LCA 05 % ./a.out
```

```
import : Keyword
java.io : Package.
* : Operator
class main : Class
{ : Punctuation
//This is a comment
  : Comment
int : Data Type
a : Identifier
= : Operator
1 : Number
; : Punctuation
int : Data Type
b : Identifier
= : Operator
2 : Number
; : Punctuation
int : Data Type
c : Identifier
; : Punctuation
c : Identifier
= : Operator
a : Identifier
+ : Operator
b : Identifier
; : Punctuation
System.out.println : Print Statement
( : Punctuation
"Hello World: " : String
+ : Operator
c : Identifier
) : Punctuation
; : Punctuation
} : Punctuation
```

Symbol Table:

ID	Type
a	int
b	int
c	int

```
(base) digvijay@Digvijays-MacBook-Air LCA 05 %
```