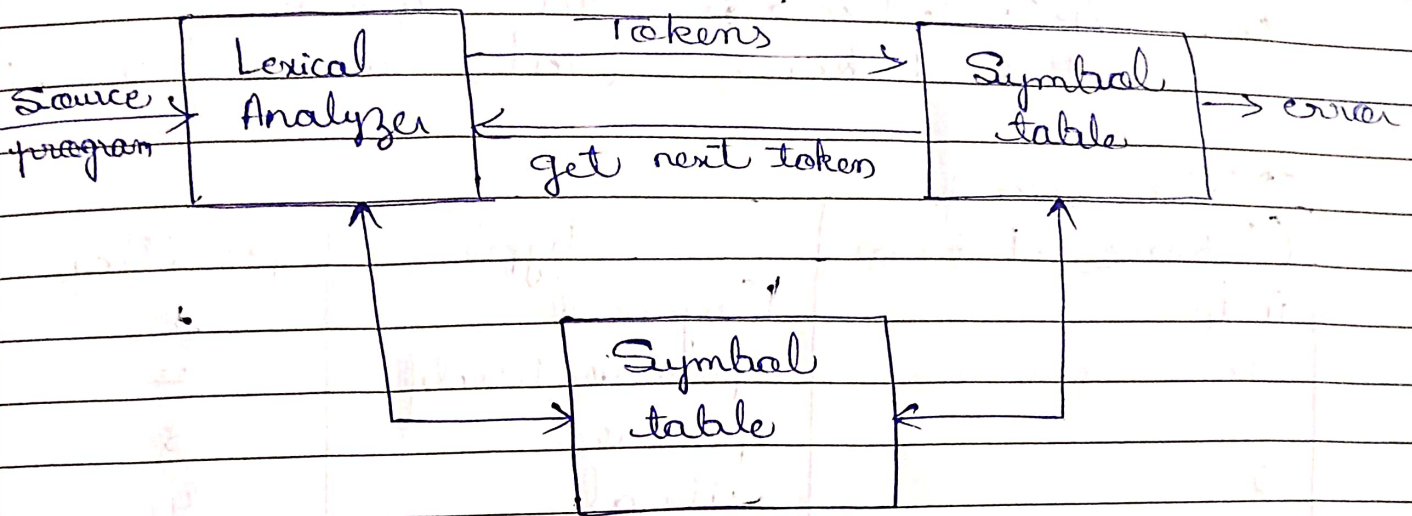


experiment 3 .



Role of Lexical Analyzer.

five stages of Lexical Analyzer.

1. Input preprocessing :

Ex: i/p : `int x=5;` // x holds value 5
o/p : `int x=5;`

2. Tokenization

Ex: i/p : `int x=5;` →

`int` → keyword
`x` → identifier
`=` → operator
`5` → literal
`;` → Delimiter

3. Token classification

Ex: (a) `int` matches with the patterns for keyword.

(b) `x` matches the pattern for identifiers. [a-z]

(c) `5` matches the pattern for integer [0-9]

4. Token validation

Ex: ~~i/p~~ i/p : `int 1x=5;`

o/p : `1x` is invalid because identifier can not start with a digit.

5. Output generation:

Ex: i/p : ~~int x = 5;~~ int x = 5;

o/p : Tokens = [int, x, =, 5, ;]

Meta data:

tokens tokens	Type	position
int	Keyword	1
x	operator identifier	2
=	operator	3
5	literal	4
;	delimiter	5