

Csc 312

NAME: SHOSAN HADIJAT ABIMBOLA

MATRIC NO:17/SCI01/076

<u>Lexemes</u>	<u>Tokens</u>
A lexeme is a sequence of characters in the source program that matches the pattern for a token and is identified by the lexical analyzer as an instance of that token.	A token is a pair consisting of a token name and an optional attribute value. The token name is an abstract symbol representing a kind of lexical unit, e.g., a particular keyword, or sequence of input characters denoting an identifier. The token names are the input symbols that the parser processes.
Lexemes are the words derived from the character input stream.	Tokens are lexemes mapped into a token-name and an attribute-value.
A lexeme is a sequence of characters from the input that match a pattern (and hence constitute an instance of a token).	Tokens are symbolic names for the entities that make up the text of the program; e.g. if for the keyword if, and id for any identifier.
SCANNER simply looks repeatedly for a lexeme in source program text until input is exhausted	Tokens are returned one at a time (by Scanner when requested by Parser) each time Scanner finds a (valid) lexeme. Scanner creates ,if not already present, a symbol-table entry (having attributes : mainly token-category and few others) , when it finds a lexeme, in order to generate it's token

Examples of lexemes and tokens

1). $x = a + b * 2$

Which yields the lexemes: {x, =, a, +, b, *, 2}

With corresponding tokens: {<id, 0>, <=>, <id, 1>, <+>, <id, 2>, <*>, <id, 3>}

2). Consider this expression in the programming language C:sum = 3 + 2;

Lexemes	Token category
Sum	Identifier

=	Assignment operator
3	Integer literal
+	Addition operator
2	Integer literal
;	End of statement