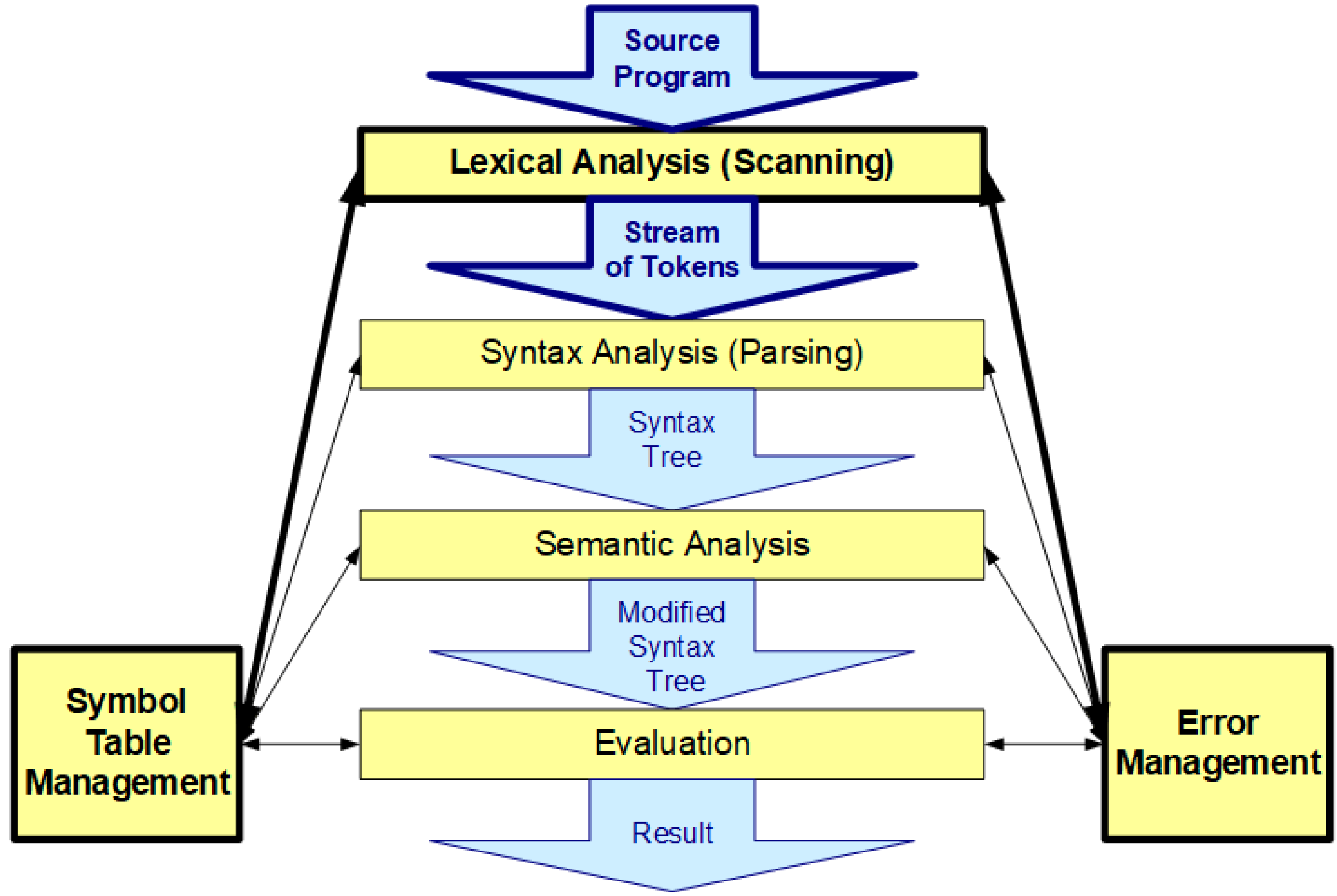
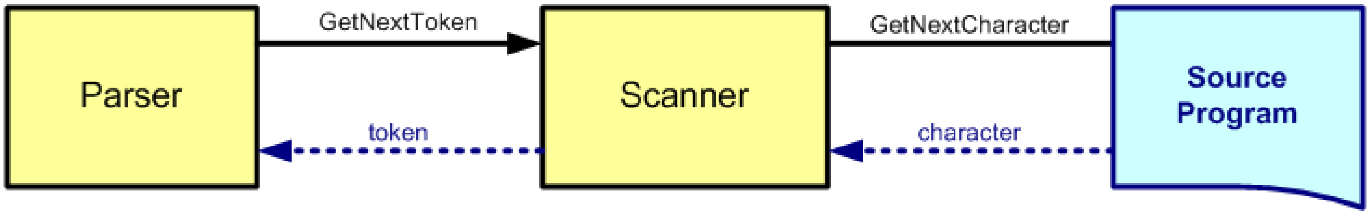
SCANNING

INTERPRETATION PHASES



LEXICAL ANALYSIS = SCANNING = LINEAR ANALYSIS



- The source program in HLL is a stream of characters read from left to right
- Scanner reads characters from that stream and groups them into tokens returned to the parser.
- Tokens are sequences of characters which form the low-level constructs of the HLL (e.g. variable names, keywords, labels, operators)
- Whitespace (blanks, tabs, returns) and comments are eliminated
- Tokens have two fields

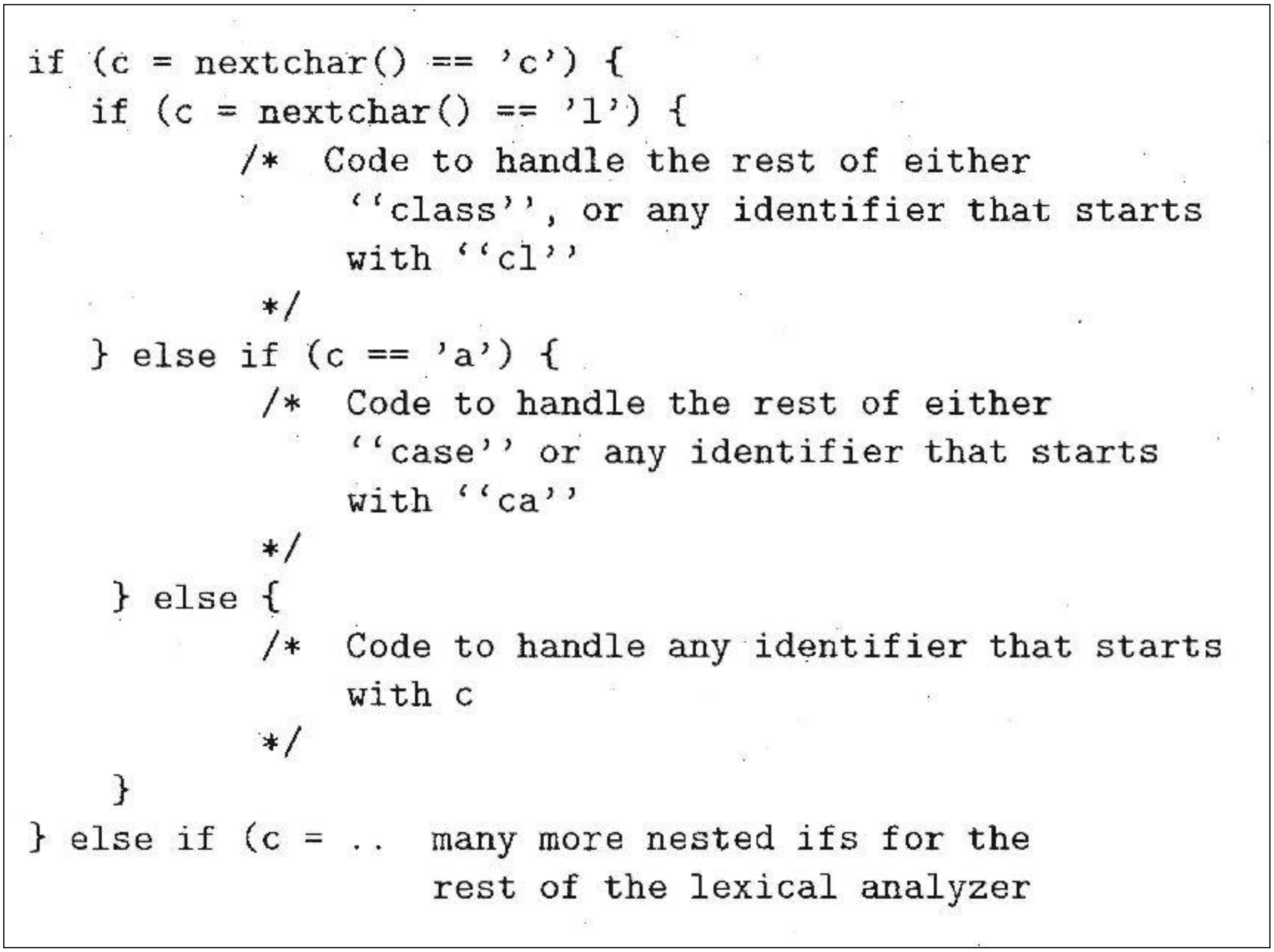
- Type (compulsory) often called “token”
- Value (optional)

Lexeme	25	hello	A	if	+	-	<	>
Type	const	ident	ident	IF	PLUS	MINUS	Comp.	Comp.
Value	25	hello	A				<	>

- Vocabulary:
  - the type is often called “token”
  - the string being scanned (instance) is called a “lexeme”

BRUTE FORCE SCANNING

Brute Force Scanner to handle identifiers starting with ‘c’:



```
if (c = nextchar() == ’c’) {
   if (c = nextchar() == ’l’) {
          /*  Code to handle the rest of either
               ‘‘class’’, or any identifier that starts
               with ‘‘cl’’
           */
   } else if (c == ’a’) {
           /*  Code to handle the rest of either
               ‘‘case’’ or any identifier that starts
               with ‘‘ca’’
           */
    } else {
           /*  Code to handle any identifier that starts
               with c
           */
    }
} else if (c = ..  many more nested ifs for the
                   rest of the lexical analyzer
```