CSC4510 Programming Language Design and Translation

Lexical Analysis Abstractions

scanner (string source_file, id_table id_t, error_handler* e);*
// Constructor for the class scanner.
// Opens the specified source file to read tokens from.
// Id_t is the identifier table for the compiler. It is only used by the scanner to implement the
// behavior of pragmas.
// E is the error handler for the scanner to use.

bool eof_flag;
// Boolean flag to indicate if the scanner has reached the end of the source file.

token get_token ();*
// Gets the next token from the input stream and returns it. The token is held in the private variable
// current_token which is returned by the function this_token() if requested by the parser.

bool have (symbol::symbol_type s);
// Returns true if the current token is an s symbol, false otherwise.

void must_be (symbol::symbol_type s);
// The current token must be an s symbol otherwise it is a syntax error. If the current token matches
// the symbol s, then the scanner discards the token and advances to the next token in the source file.

token this_token ();*
// Returns the current token, without advancing to the next token in the input stream.

Michael Oudshoorn
August 13, 2023