Phase 1 Documentation

All changes were made with reference to phase 1 documentation and project description. Documentation for the tests is available for each test in the /testSuite directory. They are complete and include our outputs for cisc458/like/examples.

Change made	**Location (where)**	**Explanation/Justification**
testSuite subdirectories	\`testSuite\`	🍩 Organize test files by topic being tested 🍩 Eg. \`/testSuite/comments\` contains all files (.pt, .txt., .eOutput) related to testing comments
Added script to automatically generate ssltrace test outputs	\`testSuite/runTests.sh\`	🍩 To automate test output generation 🍩 For each testSuite directory, perform the phase 1 ssltrace command on each *.pt file 🍩 Convention in the testSuite is that only test files are *.pt files
Update keywords	\`stdIdentifiers\` in keywords section	🍩 Removed old PT keywords 👉 array, begin, case, const, div, do mod, procedure, program, type, until 🍩 Added new Like keywords 👉 choose, elseif, fun, is, like, pkg, public, using, val, when 🍩 Run \`make scanner\` to generate scan.def 🍩 Copy contents of \`scan.def\` into \`parser.pt\`
Update predefined identifiers	\`stdIdentifiers\` in predeclared procedures and functions section	🍩 Removed old PT predefined identifiers 👉 read, readln, write, writeln 🍩 Added new Like predefined identifiers 👉 get, getln, put, putln 🍩 Run \`make scanner\` to generate scan.def 🍩 Copy contents of \`scan.def\` into \`parser.pt\`
Add new input character classes	‘Scan.ssl’ in input section “parser.pt’ in initialization of character map section	🍩 Added new input character classes since they are used in Like 👉 slash ("/"), percent ("%"), hash ("#"), bang (“!”), bar ("|") 🍩 Assigned new input characters a value for the class map 🍩 Run \`make scanner\` to generate scan.def 🍩 Copy contents of \`scan.def\` into \`parser.pt\`
Add new syntax tokens	‘Scan.ssl’ in non-compound tokens and rules section ‘parser.pt’ in section	🍩 Added syntax tokens for the new symbol tokens used in Like 👉 pSlash, pPercent, pHash, pBang, pBar, pDoubleBar, pPlusEquals, pMinusEquals, pStarEquals, pSlashEquals, pPercentEquals, pDoubleEquals 🍩 Add rules for the above to emit the correct token 🍩 Assign new syntax tokens a value

		🍩 Run `make scanner` to generate scan.def 🍩 Copy contents of `scan.def` into `parser.pt`
Add new syntax tokens for updated keywords	‘Scan.ssl’ in output section ‘parser.pt’ in output tokens	🍩 Added syntax tokens for the new keywords used in Like 👉 choose, elseif, fun, is, like, pkg, public, using, val, when 🍩 Assign new syntax tokens a value 🍩 Run `make scanner` to generate scan.def 🍩 Copy contents of `scan.def` into `parser.pt`
Remove/Replace syntax tokens	‘Scan.ssl’ in non-compound tokens section and rules section ‘Parser.pt’ in output tokens and semantic operations section	🍩 Removed syntax tokens because they’re not used in Like. 👉 pDot, pDotDot, pLeftBrace, pRightBrace, pColonEquals 🍩 Renumbered syntax tokens that are still being used after removing the unused syntax tokens inside output tokens and semantic operations. 🍩 Run `make scanner` to generate scan.def 🍩 Copy contents of `scan.def` into `parser.pt`
Change Scan rule	‘Scan.ssl’ in rules section	🍩 For lLeftParen ‘(‘ input, remove the choice selector, and replace with always emitting the .pLeftParen token, so the @AlternateComment rule is never called. 🍩 Remove any rules corresponding to a brace input ‘{‘ choice selection. Those should now emit as illegal characters, so the @Comment rule is never called here. 🍩 For the lSlash ‘/’ input, if another ‘/’ follows it, call the @Comment rule. If a lStar ‘*’ follows it, call the @AlternateComment rule.
Change Comment rule	‘Scan.ssl’ in rules section	🍩 Remove choice selection for ‘}’ input. 🍩 If lNewLine is detected, exit the loop. 🍩 Two slashes is the input stream that triggers the @Comment rule (instead of lBrace). 🍩 A lNewLine input exits the loop (which is the end of the rule).
Change AlternateComment rule	‘Scan.ssl’ in rules section	🍩 When a lStar ‘*’ is detected, replace the ‘)’ with ‘/’ to indicate that the end of the comment is reached. 🍩 A slash followed by a star input stream now triggers the @AlternateComment rule (instead of a lParen followed by a star). 🍩 A star followed by a slash input stream now exits the loop (which is the end of the rule).
Change quote from ‘ to “ for string literals	‘Parser.pt’ in input tokens section (under misc. specifically)	🍩 Change ‘’’ to ““‘. 🍩 Double quotes should now be accepted as a “quote” token by the screener (instead of single quotes). This is now the input character that triggers the string literal rule in scan.ssl.