

## SYMBOL TABLE for tstcalculator.lusd

Total Tokenized Lexemes :434

Elapsed Time :0.0010008811950683594

LINE	LEXEME	TOKEN
1	//	COMMENT_SINGLE
3	//	COMMENT_SINGLE
4	class	KEYWORD_CLASS
4	Calculator	IDENTIFIER
4	{	OPEN_CURLY_BRACKET
5	//	COMMENT_SINGLE
6	const	KEYWORD_CONST
6	float	KEYWORD_FLOAT
6	pi	IDENTIFIER
6	=	OP_ASSIGNMENT
6	3.14f	FLOAT_LITERAL
6	;	STMT_TERMINATOR
6	//	COMMENT_SINGLE
7	const	KEYWORD_CONST
7	double	KEYWORD_DOUBLE
7	gravity	IDENTIFIER
7	=	OP_ASSIGNMENT
7	9.8	DOUBLE_LITERAL
7	;	STMT_TERMINATOR
7	//	COMMENT_SINGLE
8	const	KEYWORD_CONST
8	str	KEYWORD_STR
8	appName	IDENTIFIER
8	=	OP_ASSIGNMENT
8	"Luseed Calculator"	STRING_LITERAL
8	;	STMT_TERMINATOR
8	//	COMMENT_SINGLE

LINE	LEXEME	TOKEN
9	const	KEYWORD_CONST
9	int	KEYWORD_INT
9	version	IDENTIFIER
9	=	OP_ASSIGNMENT
9	1	INT_LITERAL
9	;	STMT_TERMINATOR
9	//	COMMENT_SINGLE
11	//	COMMENT_SINGLE
12	func	KEYWORD_FUNC
12	add	IDENTIFIER
12	(	OPEN_PARENTHESIS
12	float	KEYWORD_FLOAT
12	a	IDENTIFIER
12	,	SEPARATOR
12	float	KEYWORD_FLOAT
12	b	IDENTIFIER
12	)	CLOSE_PARENTHESIS
12	{	OPEN_CURLY_BRACKET
13	return	KEYWORD_RETURN
13	a	IDENTIFIER
13	+	OP_ADDITION
13	b	IDENTIFIER
13	;	STMT_TERMINATOR
14	}	CLOSE_CURLY_BRACKET
16	//	COMMENT_SINGLE
17	func	KEYWORD_FUNC
17	calculate	IDENTIFIER
17	(	OPEN_PARENTHESIS
17	)	CLOSE_PARENTHESIS
17	{	OPEN_CURLY_BRACKET

LINE	LEXEME	TOKEN
18	//	COMMENT_SINGLE
19	display	KEYWORD_DISPLAY
19	(	OPEN_PARENTHESIS
19	"Welcome to the "	STRING_LITERAL
19	+	OP_ADDITION
19	appName	IDENTIFIER
19	+	OP_ADDITION
19	"!"	STRING_LITERAL
19	)	CLOSE_PARENTHESIS
19	;	STMT_TERMINATOR
21	//	COMMENT_SINGLE
22	display	KEYWORD_DISPLAY
22	(	OPEN_PARENTHESIS
22	"Value of pi: "	STRING_LITERAL
22	+	OP_ADDITION
22	pi	IDENTIFIER
22	)	CLOSE_PARENTHESIS
22	;	STMT_TERMINATOR
23	display	KEYWORD_DISPLAY
23	(	OPEN_PARENTHESIS
23	"Value of gravity: "	STRING_LITERAL
23	+	OP_ADDITION
23	gravity	IDENTIFIER
23	)	CLOSE_PARENTHESIS
23	;	STMT_TERMINATOR
24	display	KEYWORD_DISPLAY
24	(	OPEN_PARENTHESIS
24	"Application Name: "	STRING_LITERAL
24	+	OP_ADDITION
24	appName	IDENTIFIER

LINE	LEXEME	TOKEN
24	)	CLOSE_PARENTHESIS
24	;	STMT_TERMINATOR
25	display	KEYWORD_DISPLAY
25	(	OPEN_PARENTHESIS
25	"Version: "	STRING_LITERAL
25	+	OP_ADDITION
25	version	IDENTIFIER
25	)	CLOSE_PARENTHESIS
25	;	STMT_TERMINATOR
27	//	COMMENT_SINGLE
28	num1	IDENTIFIER
28	=	OP_ASSIGNMENT
28	float	KEYWORD_FLOAT
28	(	OPEN_PARENTHESIS
28	ask	KEYWORD_ASK
28	(	OPEN_PARENTHESIS
28	"Enter the first number: "	STRING_LITERAL
28	)	CLOSE_PARENTHESIS
28	)	CLOSE_PARENTHESIS
28	;	STMT_TERMINATOR
29	num2	IDENTIFIER
29	=	OP_ASSIGNMENT
29	float	KEYWORD_FLOAT
29	(	OPEN_PARENTHESIS
29	ask	KEYWORD_ASK
29	(	OPEN_PARENTHESIS
29	"Enter the second number: "	STRING_LITERAL
29	)	CLOSE_PARENTHESIS
29	)	CLOSE_PARENTHESIS
29	;	STMT_TERMINATOR

LINE	LEXEME	TOKEN
31	//	COMMENT_SINGLE
32	display	KEYWORD_DISPLAY
32	(	OPEN_PARENTHESIS
32	"Sum: "	STRING_LITERAL
32	+	OP_ADDITION
32	add	IDENTIFIER
32	(	OPEN_PARENTHESIS
32	num1	IDENTIFIER
32	,	SEPARATOR
32	num2	IDENTIFIER
32	)	CLOSE_PARENTHESIS
32	)	CLOSE_PARENTHESIS
32	;	STMT_TERMINATOR
33	display	KEYWORD_DISPLAY
33	(	OPEN_PARENTHESIS
33	"Difference: "	STRING_LITERAL
33	+	OP_ADDITION
33	(	OPEN_PARENTHESIS
33	num1	IDENTIFIER
33	-	OP_SUBTRACTION
33	num2	IDENTIFIER
33	)	CLOSE_PARENTHESIS
33	)	CLOSE_PARENTHESIS
33	;	STMT_TERMINATOR
34	display	KEYWORD_DISPLAY
34	(	OPEN_PARENTHESIS
34	"Product: "	STRING_LITERAL
34	+	OP_ADDITION
34	(	OPEN_PARENTHESIS
34	num1	IDENTIFIER

LINE	LEXEME	TOKEN
34	*	OP_MULTIPLICATION
34	num2	IDENTIFIER
34	)	CLOSE_PARENTHESIS
34	)	CLOSE_PARENTHESIS
34	;	STMT_TERMINATOR
35	display	KEYWORD_DISPLAY
35	(	OPEN_PARENTHESIS
35	"Quotient: "	STRING_LITERAL
35	+	OP_ADDITION
35	(	OPEN_PARENTHESIS
35	num1	IDENTIFIER
35	/	OP_DIVISION
35	num2	IDENTIFIER
35	)	CLOSE_PARENTHESIS
35	)	CLOSE_PARENTHESIS
35	;	STMT_TERMINATOR
36	display	KEYWORD_DISPLAY
36	(	OPEN_PARENTHESIS
36	"Remainder: "	STRING_LITERAL
36	+	OP_ADDITION
36	(	OPEN_PARENTHESIS
36	num1	IDENTIFIER
36	%	OP_MODULO
36	num2	IDENTIFIER
36	)	CLOSE_PARENTHESIS
36	)	CLOSE_PARENTHESIS
36	;	STMT_TERMINATOR
37	display	KEYWORD_DISPLAY
37	(	OPEN_PARENTHESIS
37	"Power: "	STRING_LITERAL

LINE	LEXEME	TOKEN
37	+	OP_ADDITION
37	(	OPEN_PARENTHESIS
37	num1	IDENTIFIER
37	**	OP_EXPONENTIATE
37	num2	IDENTIFIER
37	)	CLOSE_PARENTHESIS
37	)	CLOSE_PARENTHESIS
37	;	STMT_TERMINATOR
38	display	KEYWORD_DISPLAY
38	(	OPEN_PARENTHESIS
38	"Square Root of num1: "	STRING_LITERAL
38	+	OP_ADDITION
38	(	OPEN_PARENTHESIS
38	num1	IDENTIFIER
38	**	OP_EXPONENTIATE
38	0.5f	FLOAT_LITERAL
38	)	CLOSE_PARENTHESIS
38	)	CLOSE_PARENTHESIS
38	;	STMT_TERMINATOR
39	display	KEYWORD_DISPLAY
39	(	OPEN_PARENTHESIS
39	"Increment num1: "	STRING_LITERAL
39	+	OP_ADDITION
39	(	OPEN_PARENTHESIS
39	num1	IDENTIFIER
39	+	OP_ADDITION
39	1f	FLOAT_LITERAL
39	)	CLOSE_PARENTHESIS
39	)	CLOSE_PARENTHESIS
39	;	STMT_TERMINATOR

LINE	LEXEME	TOKEN
40	display	KEYWORD_DISPLAY
40	(	OPEN_PARENTHESIS
40	"Decrement num2: "	STRING_LITERAL
40	+	OP_ADDITION
40	(	OPEN_PARENTHESIS
40	num2	IDENTIFIER
40	-	OP_SUBTRACTION
40	1f	FLOAT_LITERAL
40	)	CLOSE_PARENTHESIS
40	)	CLOSE_PARENTHESIS
40	;	STMT_TERMINATOR
42	//	COMMENT_SINGLE
43	bool	KEYWORD_BOOL
43	resultAnd1	IDENTIFIER
43	=	OP_ASSIGNMENT
43	(	OPEN_PARENTHESIS
43	num1	IDENTIFIER
43	>	GREATER_THAN_OP
43	0f	FLOAT_LITERAL
43	)	CLOSE_PARENTHESIS
43	and	KEYWORD_AND
43	(	OPEN_PARENTHESIS
43	num2	IDENTIFIER
43	<	LESS_THAN_OP
43	10f	FLOAT_LITERAL
43	)	CLOSE_PARENTHESIS
43	;	STMT_TERMINATOR
44	bool	KEYWORD_BOOL
44	resultAnd2	IDENTIFIER
44	=	OP_ASSIGNMENT



LINE	LEXEME	TOKEN
44	(	OPEN_PARENTHESIS
44	num1	IDENTIFIER
44	==	EQUALITY_OP
44	0f	FLOAT_LITERAL
44	)	CLOSE_PARENTHESIS
44	and	KEYWORD_AND
44	(	OPEN_PARENTHESIS
44	num2	IDENTIFIER
44	!=	INEQUALITY_OP
44	0f	FLOAT_LITERAL
44	)	CLOSE_PARENTHESIS
44	;	STMT_TERMINATOR
45	bool	KEYWORD_BOOL
45	resultOr1	IDENTIFIER
45	=	OP_ASSIGNMENT
45	(	OPEN_PARENTHESIS
45	num1	IDENTIFIER
45	<	LESS_THAN_OP
45	0f	FLOAT_LITERAL
45	)	CLOSE_PARENTHESIS
45	or	KEYWORD_OR
45	(	OPEN_PARENTHESIS
45	num2	IDENTIFIER
45	>	GREATER_THAN_OP
45	10f	FLOAT_LITERAL
45	)	CLOSE_PARENTHESIS
45	;	STMT_TERMINATOR
46	bool	KEYWORD_BOOL
46	resultOr2	IDENTIFIER
46	=	OP_ASSIGNMENT

LINE	LEXEME	TOKEN
46	(	OPEN_PARENTHESIS
46	num1	IDENTIFIER
46	<=	LESS_OR_EQUAL_OP
46	5f	FLOAT_LITERAL
46	)	CLOSE_PARENTHESIS
46	or	KEYWORD_OR
46	(	OPEN_PARENTHESIS
46	num2	IDENTIFIER
46	>=	GREATER_OR_EQUAL_OP
46	5f	FLOAT_LITERAL
46	)	CLOSE_PARENTHESIS
46	;	STMT_TERMINATOR
47	bool	KEYWORD_BOOL
47	resultNot1	IDENTIFIER
47	=	OP_ASSIGNMENT
47	not	KEYWORD_NOT
47	(	OPEN_PARENTHESIS
47	num1f	IDENTIFIER
47	==	EQUALITY_OP
47	num2f	IDENTIFIER
47	)	CLOSE_PARENTHESIS
47	;	STMT_TERMINATOR
48	bool	KEYWORD_BOOL
48	resultNot2	IDENTIFIER
48	=	OP_ASSIGNMENT
48	not	KEYWORD_NOT
48	(	OPEN_PARENTHESIS
48	num1f	IDENTIFIER
48	>=	GREATER_OR_EQUAL_OP
48	num2f	IDENTIFIER

LINE	LEXEME	TOKEN
48	)	CLOSE_PARENTHESIS
48	;	STMT_TERMINATOR
49	bool	KEYWORD_BOOL
49	resultEqual	IDENTIFIER
49	=	OP_ASSIGNMENT
49	(	OPEN_PARENTHESIS
49	num1f	IDENTIFIER
49	==	EQUALITY_OP
49	num2f	IDENTIFIER
49	)	CLOSE_PARENTHESIS
49	;	STMT_TERMINATOR
50	bool	KEYWORD_BOOL
50	resultNotEqual	IDENTIFIER
50	=	OP_ASSIGNMENT
50	(	OPEN_PARENTHESIS
50	num1f	IDENTIFIER
50	!=	INEQUALITY_OP
50	num2f	IDENTIFIER
50	)	CLOSE_PARENTHESIS
50	;	STMT_TERMINATOR
51	bool	KEYWORD_BOOL
51	resultGreater	IDENTIFIER
51	=	OP_ASSIGNMENT
51	(	OPEN_PARENTHESIS
51	num1f	IDENTIFIER
51	>	GREATER_THAN_OP
51	num2f	IDENTIFIER
51	)	CLOSE_PARENTHESIS
51	;	STMT_TERMINATOR
52	bool	KEYWORD_BOOL

LINE	LEXEME	TOKEN
52	resultLess	IDENTIFIER
52	=	OP_ASSIGNMENT
52	(	OPEN_PARENTHESIS
52	num1f	IDENTIFIER
52	<	LESS_THAN_OP
52	num2f	IDENTIFIER
52	)	CLOSE_PARENTHESIS
52	;	STMT_TERMINATOR
54	//	COMMENT_SINGLE
55	display	KEYWORD_DISPLAY
55	(	OPEN_PARENTHESIS
55	"Result of AND operation 1: "	STRING_LITERAL
55	+	OP_ADDITION
55	resultAnd1	IDENTIFIER
55	)	CLOSE_PARENTHESIS
55	;	STMT_TERMINATOR
56	display	KEYWORD_DISPLAY
56	(	OPEN_PARENTHESIS
56	"Result of AND operation 2: "	STRING_LITERAL
56	+	OP_ADDITION
56	resultAnd2	IDENTIFIER
56	)	CLOSE_PARENTHESIS
56	;	STMT_TERMINATOR
57	display	KEYWORD_DISPLAY
57	(	OPEN_PARENTHESIS
57	"Result of OR operation 1: "	STRING_LITERAL
57	+	OP_ADDITION
57	resultOr1	IDENTIFIER
57	)	CLOSE_PARENTHESIS
57	;	STMT_TERMINATOR

LINE	LEXEME	TOKEN
58	display	KEYWORD_DISPLAY
58	(	OPEN_PARENTHESIS
58	"Result of OR operation 2: "	STRING_LITERAL
58	+	OP_ADDITION
58	resultOr2	IDENTIFIER
58	)	CLOSE_PARENTHESIS
58	;	STMT_TERMINATOR
59	display	KEYWORD_DISPLAY
59	(	OPEN_PARENTHESIS
59	"Result of NOT operation 1: "	STRING_LITERAL
59	+	OP_ADDITION
59	resultNot1	IDENTIFIER
59	)	CLOSE_PARENTHESIS
59	;	STMT_TERMINATOR
60	display	KEYWORD_DISPLAY
60	(	OPEN_PARENTHESIS
60	"Result of NOT operation 2: "	STRING_LITERAL
60	+	OP_ADDITION
60	resultNot2	IDENTIFIER
60	)	CLOSE_PARENTHESIS
60	;	STMT_TERMINATOR
61	display	KEYWORD_DISPLAY
61	(	OPEN_PARENTHESIS
61	"Result of Equal operation: "	STRING_LITERAL
61	+	OP_ADDITION
61	resultEqual	IDENTIFIER
61	)	CLOSE_PARENTHESIS
61	;	STMT_TERMINATOR
62	display	KEYWORD_DISPLAY
62	(	OPEN_PARENTHESIS

LINE	LEXEME	TOKEN
62	"Result of Not Equal operation: "	STRING_LITERAL
62	+	OP_ADDITION
62	resultNotEqual	IDENTIFIER
62	)	CLOSE_PARENTHESIS
62	;	STMT_TERMINATOR
63	display	KEYWORD_DISPLAY
63	(	OPEN_PARENTHESIS
63	"Result of Greater operation: "	STRING_LITERAL
63	+	OP_ADDITION
63	resultGreater	IDENTIFIER
63	)	CLOSE_PARENTHESIS
63	;	STMT_TERMINATOR
64	display	KEYWORD_DISPLAY
64	(	OPEN_PARENTHESIS
64	"Result of Less operation: "	STRING_LITERAL
64	+	OP_ADDITION
64	resultLess	IDENTIFIER
64	)	CLOSE_PARENTHESIS
64	;	STMT_TERMINATOR
66	//	COMMENT_SINGLE
67	display	KEYWORD_DISPLAY
67	(	OPEN_PARENTHESIS
67	"End of the Luseed program."	STRING_LITERAL
67	)	CLOSE_PARENTHESIS
67	;	STMT_TERMINATOR
68	}	CLOSE_CURLY_BRACKET
69	}	CLOSE_CURLY_BRACKET
71	main	KEYWORD_MAIN
71	(	OPEN_PARENTHESIS
71	)	CLOSE_PARENTHESIS

LINE	LEXEME	TOKEN
71	{	OPEN_CURLY_BRACKET
72	//	COMMENT_SINGLE
73	obj	KEYWORD_OBJ
73	calcObj	IDENTIFIER
73	=	OP_ASSIGNMENT
73	Calculator	IDENTIFIER
73	(	OPEN_PARENTHESIS
73	)	CLOSE_PARENTHESIS
73	;	STMT_TERMINATOR
75	//	COMMENT_SINGLE
76	calcObj	IDENTIFIER
76	.	OBJECT_DELIMITER
76	calculate	IDENTIFIER
76	(	OPEN_PARENTHESIS
76	)	CLOSE_PARENTHESIS
76	;	STMT_TERMINATOR
77	}	CLOSE_CURLY_BRACKET