

## SYMBOL TABLE for tstcalculator.lusd

Total Tokenized Lexemes :488

Elapsed Time :0.002916574478149414

LINE	LEXEME	TOKEN
1	//	COMMENT_SINGLE
3	//	COMMENT_SINGLE
4	public	KEYWORD_PUBLIC
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

LINE	LEXEME	TOKEN
8	//	COMMENT_SINGLE
9	constant	KEYWORD_CONST
9	int	KEYWORD_INT
9	version	IDENTIFIER
9	=	OP_ASSIGNMENT
9	1	INT_LITERAL
9	;	STMT_TERMINATOR
9	//	COMMENT_SINGLE
10	constant	KEYWORD_CONST
10	bool	KEYWORD_BOOL
10	isOn	IDENTIFIER
10	=	OP_ASSIGNMENT
10	true	KEYWORD_TRUE
10	//	COMMENT_SINGLE
12	public	KEYWORD_PUBLIC
12	integer	KEYWORD_INT
12	testvar1	IDENTIFIER
12	=	OP_ASSIGNMENT
12	1	INT_LITERAL
12	;	STMT_TERMINATOR
12	//	COMMENT_SINGLE
13	public	KEYWORD_PUBLIC
13	character	KEYWORD_CHAR
13	char1	IDENTIFIER
13	=	OP_ASSIGNMENT
13	'z'	CHAR_LITERAL
13	;	STMT_TERMINATOR
13	//	COMMENT_SINGLE
15	//	COMMENT_SINGLE
16	int	KEYWORD_INT

LINE	LEXEME	TOKEN
16	8num	UNKNOWN_TOKEN
16	;	STMT_TERMINATOR
16	//	COMMENT_SINGLE
17	int	KEYWORD_INT
17	x	IDENTIFIER
17	=	OP_ASSIGNMENT
17	5	INT_LITERAL
17	&	UNKNOWN_TOKEN
17	3	INT_LITERAL
17	;	STMT_TERMINATOR
17	//	COMMENT_SINGLE
18	char	KEYWORD_CHAR
18	blood_type	IDENTIFIER
18	=	OP_ASSIGNMENT
18	'ab'	UNKNOWN_TOKEN
18	//	COMMENT_SINGLE
20	//	COMMENT_SINGLE
21	func	KEYWORD_FUNC
21	add	IDENTIFIER
21	(	OPEN_PARENTHESIS
21	float	KEYWORD_FLOAT
21	a	IDENTIFIER
21	,	SEPARATOR
21	float	KEYWORD_FLOAT
21	b	IDENTIFIER
21	)	CLOSE_PARENTHESIS
21	{	OPEN_CURLY_BRACKET
22	return	KEYWORD_RETURN
22	a	IDENTIFIER
22	+	OP_ADDITION

LINE	LEXEME	TOKEN
22	b	IDENTIFIER
22	;	STMT_TERMINATOR
23	}	CLOSE_CURLY_BRACKET
25	//	COMMENT_SINGLE
26	func	KEYWORD_FUNC
26	calculate	IDENTIFIER
26	(	OPEN_PARENTHESIS
26	)	CLOSE_PARENTHESIS
26	{	OPEN_CURLY_BRACKET
27	//	COMMENT_SINGLE
28	display	KEYWORD_DISPLAY
28	(	OPEN_PARENTHESIS
28	"Welcome to the "	STRING_LITERAL
28	+	OP_ADDITION
28	appName	IDENTIFIER
28	+	OP_ADDITION
28	"!"	STRING_LITERAL
28	)	CLOSE_PARENTHESIS
28	;	STMT_TERMINATOR
30	//	COMMENT_SINGLE
31	display	KEYWORD_DISPLAY
31	(	OPEN_PARENTHESIS
31	"Value of pi: "	STRING_LITERAL
31	+	OP_ADDITION
31	pi	IDENTIFIER
31	)	CLOSE_PARENTHESIS
31	;	STMT_TERMINATOR
32	display	KEYWORD_DISPLAY
32	(	OPEN_PARENTHESIS
32	"Value of gravity: "	STRING_LITERAL

LINE	LEXEME	TOKEN
32	,	SEPARATOR
32	gravity	IDENTIFIER
32	)	CLOSE_PARENTHESIS
32	;	STMT_TERMINATOR
33	display	KEYWORD_DISPLAY
33	(	OPEN_PARENTHESIS
33	"Application Name: "	STRING_LITERAL
33	,	SEPARATOR
33	appName	IDENTIFIER
33	)	CLOSE_PARENTHESIS
33	;	STMT_TERMINATOR
34	display	KEYWORD_DISPLAY
34	(	OPEN_PARENTHESIS
34	"Version: "	STRING_LITERAL
34	+	OP_ADDITION
34	version	IDENTIFIER
34	)	CLOSE_PARENTHESIS
34	;	STMT_TERMINATOR
36	//	COMMENT_SINGLE
37	num1	IDENTIFIER
37	=	OP_ASSIGNMENT
37	float	KEYWORD_FLOAT
37	(	OPEN_PARENTHESIS
37	ask	KEYWORD_ASK
37	(	OPEN_PARENTHESIS
37	"Enter the first number: "	STRING_LITERAL
37	)	CLOSE_PARENTHESIS
37	)	CLOSE_PARENTHESIS
37	;	STMT_TERMINATOR
38	num2	IDENTIFIER

LINE	LEXEME	TOKEN
38	=	OP_ASSIGNMENT
38	float	KEYWORD_FLOAT
38	(	OPEN_PARENTHESIS
38	ask	KEYWORD_ASK
38	(	OPEN_PARENTHESIS
38	"Enter the second number: "	STRING_LITERAL
38	)	CLOSE_PARENTHESIS
38	)	CLOSE_PARENTHESIS
38	;	STMT_TERMINATOR
40	//	COMMENT_SINGLE
41	display	KEYWORD_DISPLAY
41	(	OPEN_PARENTHESIS
41	"Sum: "	STRING_LITERAL
41	+	OP_ADDITION
41	add	IDENTIFIER
41	(	OPEN_PARENTHESIS
41	num1	IDENTIFIER
41	,	SEPARATOR
41	num2	IDENTIFIER
41	)	CLOSE_PARENTHESIS
41	)	CLOSE_PARENTHESIS
41	;	STMT_TERMINATOR
42	display	KEYWORD_DISPLAY
42	(	OPEN_PARENTHESIS
42	"Difference: "	STRING_LITERAL
42	,	SEPARATOR
42	(	OPEN_PARENTHESIS
42	num1	IDENTIFIER
42	-	OP_SUBTRACTION
42	num2	IDENTIFIER

LINE	LEXEME	TOKEN
42	)	CLOSE_PARENTHESIS
42	)	CLOSE_PARENTHESIS
42	;	STMT_TERMINATOR
43	display	KEYWORD_DISPLAY
43	(	OPEN_PARENTHESIS
43	"Product: "	STRING_LITERAL
43	,	SEPARATOR
43	(	OPEN_PARENTHESIS
43	num1	IDENTIFIER
43	*	OP_MULTIPLICATION
43	num2	IDENTIFIER
43	)	CLOSE_PARENTHESIS
43	)	CLOSE_PARENTHESIS
43	;	STMT_TERMINATOR
44	display	KEYWORD_DISPLAY
44	(	OPEN_PARENTHESIS
44	"Quotient: "	STRING_LITERAL
44	,	SEPARATOR
44	(	OPEN_PARENTHESIS
44	num1	IDENTIFIER
44	/	OP_DIVISION
44	num2	IDENTIFIER
44	)	CLOSE_PARENTHESIS
44	)	CLOSE_PARENTHESIS
44	;	STMT_TERMINATOR
45	display	KEYWORD_DISPLAY
45	(	OPEN_PARENTHESIS
45	"Remainder: "	STRING_LITERAL
45	,	SEPARATOR
45	(	OPEN_PARENTHESIS

LINE	LEXEME	TOKEN
45	num1	IDENTIFIER
45	%	OP_MODULO
45	num2	IDENTIFIER
45	)	CLOSE_PARENTHESIS
45	)	CLOSE_PARENTHESIS
45	;	STMT_TERMINATOR
46	display	KEYWORD_DISPLAY
46	(	OPEN_PARENTHESIS
46	"Power: "	STRING_LITERAL
46	,	SEPARATOR
46	(	OPEN_PARENTHESIS
46	num1	IDENTIFIER
46	**	OP_EXPONENTIATE
46	num2	IDENTIFIER
46	)	CLOSE_PARENTHESIS
46	)	CLOSE_PARENTHESIS
46	;	STMT_TERMINATOR
47	display	KEYWORD_DISPLAY
47	(	OPEN_PARENTHESIS
47	"Square Root of num1: "	STRING_LITERAL
47	,	SEPARATOR
47	(	OPEN_PARENTHESIS
47	num1	IDENTIFIER
47	**	OP_EXPONENTIATE
47	0.5f	FLOAT_LITERAL
47	)	CLOSE_PARENTHESIS
47	)	CLOSE_PARENTHESIS
47	;	STMT_TERMINATOR
48	display	KEYWORD_DISPLAY
48	(	OPEN_PARENTHESIS



LINE	LEXEME	TOKEN
48	"Increment num1: "	STRING_LITERAL
48	,	SEPARATOR
48	(	OPEN_PARENTHESIS
48	num1	IDENTIFIER
48	+	OP_ADDITION
48	1f	FLOAT_LITERAL
48	)	CLOSE_PARENTHESIS
48	)	CLOSE_PARENTHESIS
48	;	STMT_TERMINATOR
49	display	KEYWORD_DISPLAY
49	(	OPEN_PARENTHESIS
49	"Decrement num2: "	STRING_LITERAL
49	,	SEPARATOR
49	(	OPEN_PARENTHESIS
49	num2	IDENTIFIER
49	-	OP_SUBTRACTION
49	1f	FLOAT_LITERAL
49	)	CLOSE_PARENTHESIS
49	)	CLOSE_PARENTHESIS
49	;	STMT_TERMINATOR
51	//	COMMENT_SINGLE
52	bool	KEYWORD_BOOL
52	resultAnd1	IDENTIFIER
52	=	OP_ASSIGNMENT
52	(	OPEN_PARENTHESIS
52	num1	IDENTIFIER
52	>	GREATER_THAN_OP
52	0f	FLOAT_LITERAL
52	)	CLOSE_PARENTHESIS
52	and	KEYWORD_AND

LINE	LEXEME	TOKEN
52	(	OPEN_PARENTHESIS
52	num2	IDENTIFIER
52	<	LESS_THAN_OP
52	10f	FLOAT_LITERAL
52	)	CLOSE_PARENTHESIS
52	;	STMT_TERMINATOR
53	bool	KEYWORD_BOOL
53	resultAnd2	IDENTIFIER
53	=	OP_ASSIGNMENT
53	(	OPEN_PARENTHESIS
53	num1	IDENTIFIER
53	==	EQUALITY_OP
53	0f	FLOAT_LITERAL
53	)	CLOSE_PARENTHESIS
53	and	KEYWORD_AND
53	(	OPEN_PARENTHESIS
53	num2	IDENTIFIER
53	!=	INEQUALITY_OP
53	0f	FLOAT_LITERAL
53	)	CLOSE_PARENTHESIS
53	;	STMT_TERMINATOR
54	bool	KEYWORD_BOOL
54	resultOr1	IDENTIFIER
54	=	OP_ASSIGNMENT
54	(	OPEN_PARENTHESIS
54	num1	IDENTIFIER
54	<	LESS_THAN_OP
54	0f	FLOAT_LITERAL
54	)	CLOSE_PARENTHESIS
54	or	KEYWORD_OR

LINE	LEXEME	TOKEN
54	(	OPEN_PARENTHESIS
54	num2	IDENTIFIER
54	>	GREATER_THAN_OP
54	10f	FLOAT_LITERAL
54	)	CLOSE_PARENTHESIS
54	;	STMT_TERMINATOR
55	bool	KEYWORD_BOOL
55	resultOr2	IDENTIFIER
55	=	OP_ASSIGNMENT
55	(	OPEN_PARENTHESIS
55	num1	IDENTIFIER
55	<=	LESS_OR_EQUAL_OP
55	5f	FLOAT_LITERAL
55	)	CLOSE_PARENTHESIS
55	or	KEYWORD_OR
55	(	OPEN_PARENTHESIS
55	num2	IDENTIFIER
55	>=	GREATER_OR_EQUAL_OP
55	5f	FLOAT_LITERAL
55	)	CLOSE_PARENTHESIS
55	;	STMT_TERMINATOR
56	bool	KEYWORD_BOOL
56	resultNot1	IDENTIFIER
56	=	OP_ASSIGNMENT
56	not	KEYWORD_NOT
56	(	OPEN_PARENTHESIS
56	num1f	IDENTIFIER
56	==	EQUALITY_OP
56	num2f	IDENTIFIER
56	)	CLOSE_PARENTHESIS

LINE	LEXEME	TOKEN
56	;	STMT_TERMINATOR
57	bool	KEYWORD_BOOL
57	resultNot2	IDENTIFIER
57	=	OP_ASSIGNMENT
57	not	KEYWORD_NOT
57	(	OPEN_PARENTHESIS
57	num1f	IDENTIFIER
57	>=	GREATER_OR_EQUAL_OP
57	num2f	IDENTIFIER
57	)	CLOSE_PARENTHESIS
57	;	STMT_TERMINATOR
58	bool	KEYWORD_BOOL
58	resultEqual	IDENTIFIER
58	=	OP_ASSIGNMENT
58	(	OPEN_PARENTHESIS
58	num1f	IDENTIFIER
58	==	EQUALITY_OP
58	num2f	IDENTIFIER
58	)	CLOSE_PARENTHESIS
58	;	STMT_TERMINATOR
59	bool	KEYWORD_BOOL
59	resultNotEqual	IDENTIFIER
59	=	OP_ASSIGNMENT
59	(	OPEN_PARENTHESIS
59	num1f	IDENTIFIER
59	!=	INEQUALITY_OP
59	num2f	IDENTIFIER
59	)	CLOSE_PARENTHESIS
59	;	STMT_TERMINATOR
60	bool	KEYWORD_BOOL

LINE	LEXEME	TOKEN
60	resultGreater	IDENTIFIER
60	=	OP_ASSIGNMENT
60	(	OPEN_PARENTHESIS
60	num1f	IDENTIFIER
60	>	GREATER_THAN_OP
60	num2f	IDENTIFIER
60	)	CLOSE_PARENTHESIS
60	;	STMT_TERMINATOR
61	bool	KEYWORD_BOOL
61	resultLess	IDENTIFIER
61	=	OP_ASSIGNMENT
61	(	OPEN_PARENTHESIS
61	num1f	IDENTIFIER
61	<	LESS_THAN_OP
61	num2f	IDENTIFIER
61	)	CLOSE_PARENTHESIS
61	;	STMT_TERMINATOR
63	//	COMMENT_SINGLE
64	display	KEYWORD_DISPLAY
64	(	OPEN_PARENTHESIS
64	"Result of AND operation 1: "	STRING_LITERAL
64	,	SEPARATOR
64	resultAnd1	IDENTIFIER
64	)	CLOSE_PARENTHESIS
64	;	STMT_TERMINATOR
65	display	KEYWORD_DISPLAY
65	(	OPEN_PARENTHESIS
65	"Result of AND operation 2: "	STRING_LITERAL
65	,	SEPARATOR
65	resultAnd2	IDENTIFIER

LINE	LEXEME	TOKEN
65	)	CLOSE_PARENTHESIS
65	;	STMT_TERMINATOR
66	display	KEYWORD_DISPLAY
66	(	OPEN_PARENTHESIS
66	"Result of OR operation 1: "	STRING_LITERAL
66	,	SEPARATOR
66	resultOr1	IDENTIFIER
66	)	CLOSE_PARENTHESIS
66	;	STMT_TERMINATOR
67	display	KEYWORD_DISPLAY
67	(	OPEN_PARENTHESIS
67	"Result of OR operation 2: "	STRING_LITERAL
67	,	SEPARATOR
67	resultOr2	IDENTIFIER
67	)	CLOSE_PARENTHESIS
67	;	STMT_TERMINATOR
68	display	KEYWORD_DISPLAY
68	(	OPEN_PARENTHESIS
68	"Result of NOT operation 1: "	STRING_LITERAL
68	,	SEPARATOR
68	resultNot1	IDENTIFIER
68	)	CLOSE_PARENTHESIS
68	;	STMT_TERMINATOR
69	display	KEYWORD_DISPLAY
69	(	OPEN_PARENTHESIS
69	"Result of NOT operation 2: "	STRING_LITERAL
69	,	SEPARATOR
69	resultNot2	IDENTIFIER
69	)	CLOSE_PARENTHESIS
69	;	STMT_TERMINATOR

LINE	LEXEME	TOKEN
70	display	KEYWORD_DISPLAY
70	(	OPEN_PARENTHESIS
70	"Result of Equal operation: "	STRING_LITERAL
70	,	SEPARATOR
70	resultEqual	IDENTIFIER
70	)	CLOSE_PARENTHESIS
70	;	STMT_TERMINATOR
71	display	KEYWORD_DISPLAY
71	(	OPEN_PARENTHESIS
71	"Result of Not Equal operation: "	STRING_LITERAL
71	,	SEPARATOR
71	resultNotEqual	IDENTIFIER
71	)	CLOSE_PARENTHESIS
71	;	STMT_TERMINATOR
72	display	KEYWORD_DISPLAY
72	(	OPEN_PARENTHESIS
72	"Result of Greater operation: "	STRING_LITERAL
72	,	SEPARATOR
72	resultGreater	IDENTIFIER
72	)	CLOSE_PARENTHESIS
72	;	STMT_TERMINATOR
73	display	KEYWORD_DISPLAY
73	(	OPEN_PARENTHESIS
73	"Result of Less operation: "	STRING_LITERAL
73	,	SEPARATOR
73	resultLess	IDENTIFIER
73	)	CLOSE_PARENTHESIS
73	;	STMT_TERMINATOR
75	//	COMMENT_SINGLE
76	display	KEYWORD_DISPLAY

LINE	LEXEME	TOKEN
76	(	OPEN_PARENTHESIS
76	"End of the Luseed program."	STRING_LITERAL
76	)	CLOSE_PARENTHESIS
76	;	STMT_TERMINATOR
77	}	CLOSE_CURLY_BRACKET
78	}	CLOSE_CURLY_BRACKET
80	main	KEYWORD_MAIN
80	(	OPEN_PARENTHESIS
80	)	CLOSE_PARENTHESIS
80	{	OPEN_CURLY_BRACKET
81	//	COMMENT_SINGLE
82	obj	KEYWORD_OBJ
82	calcObj	IDENTIFIER
82	=	OP_ASSIGNMENT
82	Calculator	IDENTIFIER
82	(	OPEN_PARENTHESIS
82	)	CLOSE_PARENTHESIS
82	;	STMT_TERMINATOR
84	//	COMMENT_SINGLE
85	calcObj	IDENTIFIER
85	.	OBJECT_DELIMITER
85	calculate	IDENTIFIER
85	(	OPEN_PARENTHESIS
85	)	CLOSE_PARENTHESIS
85	;	STMT_TERMINATOR
87	//	COMMENT_SINGLE
88	object	KEYWORD_OBJ
88	calcObj1	IDENTIFIER
88	=	OP_ASSIGNMENT
88	Calculator	IDENTIFIER



LINE	LEXEME	TOKEN
88	(	OPEN_PARENTHESIS
88	)	CLOSE_PARENTHESIS
88	;	STMT_TERMINATOR
90	//	COMMENT_SINGLE
91	calcObj1	IDENTIFIER
91	.	OBJECT_DELIMITER
91	calculate	IDENTIFIER
91	(	OPEN_PARENTHESIS
91	)	CLOSE_PARENTHESIS
91	;	STMT_TERMINATOR
92	}	CLOSE_CURLY_BRACKET