SYMBOL TABLE for expression.lusd

Total Tokenized Lexemes :618
Elapsed Time :0.0030069351196289062

LINE	LEXEME	TOKEN
1	from	KW_FROM
1	lib	IDENTIFIER
1	import	KW_IMPORT
1	all	KW_ALL
1	;	DLM_TRMNTR
1	\n	WHT_NEWLINE
2	\n	WHT_NEWLINE
3	main	KW_MAIN
3	(	DLM_LPRN
3	)	DLM_RPRN
3	:	DLM_CODEBLK
3	\n	WHT_NEWLINE
4	{	DLM_LCURLY
4	\n	WHT_NEWLINE
5	int	KW_DATA_INT
5	expr	IDENTIFIER
5	=	OP_ASGN
5	1	LIT_INT
5	+	OP_PLUS
5	1	LIT_INT
5	*	OP_MUL
5	2	LIT_INT
5	;	DLM_TRMNTR
5	\n	WHT_NEWLINE
6	//const int var1 = 5;	CMNT_SINGLE
6	\n	WHT_NEWLINE
7	public	KW_PUB

LINE	LEXEME	TOKEN
7	int	KW_DATA_INT
7	var1	IDENTIFIER
7	=	OP_ASGN
7	5	LIT_INT
7	;	DLM_TRMNTR
7	\n	WHT_NEWLINE
8	string	KW_DATA_STR
8	hello	IDENTIFIER
8	=	OP_ASGN
8	5	LIT_INT
8	;	DLM_TRMNTR
8	\n	WHT_NEWLINE
9	\n	WHT_NEWLINE
10	i	IDENTIFIER
10	+=	OP_ADD_ASGN
10	5	LIT_INT
10	;	DLM_TRMNTR
10	\n	WHT_NEWLINE
11	var1	IDENTIFIER
11	~=	OP_FLRDIVD_ASGN
11	true	KW_TRUE
11	;	DLM_TRMNTR
11	\n	WHT_NEWLINE
12	\n	WHT_NEWLINE
13	ask	KW_ASK
13	(	DLM_LPRN
13	"Hellp"	LIT_STR
13	)	DLM_RPRN
13	;	DLM_TRMNTR
13	\n	WHT_NEWLINE

LINE	LEXEME	TOKEN
14	int	KW_DATA_INT
14	var1	IDENTIFIER
14	=	OP_ASGN
14	(	DLM_LPRN
14	int	KW_DATA_INT
14	)	DLM_RPRN
14	Num1	IDENTIFIER
14	;	DLM_TRMNTR
14	\n	WHT_NEWLINE
15	str	KW_DATA_STR
15	name	IDENTIFIER
15	=	OP_ASGN
15	ask	KW_ASK
15	(	DLM_LPRN
15	"Enter your name"	LIT_STR
15	)	DLM_RPRN
15	;	DLM_TRMNTR
15	\n	WHT_NEWLINE
16	display	KW_DISPLAY
16	(	DLM_LPRN
16	"I love cookies\t"	LIT_STR
16	,	DLM_SPRTR
16	"hehe"	LIT_STR
16	)	DLM_RPRN
16	;	DLM_TRMNTR
16	\n	WHT_NEWLINE
17	display	KW_DISPLAY
17	(	DLM_LPRN
17	IDENTI	IDENTIFIER
17	)	DLM_RPRN

LINE	LEXEME	TOKEN
17	;	DLM_TRMNTR
17	\n	WHT_NEWLINE
18	display	KW_DISPLAY
18	(	DLM_LPRN
18	" "	LIT_STR
18	)	DLM_RPRN
18	;	DLM_TRMNTR
18	\n	WHT_NEWLINE
19	\n	WHT_NEWLINE
20	if	KW_IF
20	(	DLM_LPRN
20	x	IDENTIFIER
20	==	OP_EQ
20	5	LIT_INT
20	)	DLM_RPRN
20	then	KW_THEN
20	:	DLM_CODEBLK
20	{	DLM_LCURLY
20	\n	WHT_NEWLINE
21	x	IDENTIFIER
21	++	OP_INCR
21	;	DLM_TRMNTR
21	\n	WHT_NEWLINE
22	}	DLM_RCURLY
22	\n	WHT_NEWLINE
23	\n	WHT_NEWLINE
24	if	KW_IF
24	(	DLM_LPRN
24	x	IDENTIFIER
24	>	OP_GRTR_THAN

LINE	LEXEME	TOKEN
24	5	LIT_INT
24	)	DLM_RPRN
24	then	KW_THEN
24	:	DLM_CODEBLK
24	\n	WHT_NEWLINE
25	{	DLM_LCURLY
25	\n	WHT_NEWLINE
26	display	KW_DISPLAY
26	(	DLM_LPRN
26	"HELLO"	LIT_STR
26	)	DLM_RPRN
26	;	DLM_TRMNTR
26	\n	WHT_NEWLINE
27	}	DLM_RCURLY
27	\n	WHT_NEWLINE
28	elif	KW_ELIF
28	(	DLM_LPRN
28	x	IDENTIFIER
28	<	OP_LSS_THAN
28	5	LIT_INT
28	)	DLM_RPRN
28	then	KW_THEN
28	:	DLM_CODEBLK
28	\n	WHT_NEWLINE
29	{	DLM_LCURLY
29	\n	WHT_NEWLINE
30	display	KW_DISPLAY
30	(	DLM_LPRN
30	"HI"	LIT_STR
30	)	DLM_RPRN

LINE	LEXEME	TOKEN
30	;	DLM_TRMNTR
30	\n	WHT_NEWLINE
31	}	DLM_RCURLY
31	\n	WHT_NEWLINE
32	else	KW_ELSE
32	:	DLM_CODEBLK
32	\n	WHT_NEWLINE
33	{	DLM_LCURLY
33	\n	WHT_NEWLINE
34	return	KW_RETURN
34	;	DLM_TRMNTR
34	\n	WHT_NEWLINE
35	return	KW_RETURN
35	1	LIT_INT
35	;	DLM_TRMNTR
35	\n	WHT_NEWLINE
36	}	DLM_RCURLY
36	\n	WHT_NEWLINE
37	\n	WHT_NEWLINE
38	repeat	KW_LOOP_REPEAT
38	(	DLM_LPRN
38	10	LIT_INT
38	)	DLM_RPRN
38	:	DLM_CODEBLK
38	\n	WHT_NEWLINE
39	{	DLM_LCURLY
39	\n	WHT_NEWLINE
40	break	KW_BREAK
40	;	DLM_TRMNTR
40	\n	WHT_NEWLINE

LINE	LEXEME	TOKEN
41	break	KW_BREAK
41	loop1	IDENTIFIER
41	;	DLM_TRMNTR
41	\n	WHT_NEWLINE
42	}	DLM_RCURLY
42	\n	WHT_NEWLINE
43	\n	WHT_NEWLINE
44	display	KW_DISPLAY
44	(	DLM_LPRN
44	x	IDENTIFIER
44	)	DLM_RPRN
44	;	DLM_TRMNTR
44	\n	WHT_NEWLINE
45	\n	WHT_NEWLINE
46	loop	KW_LOOP
46	Identifier	IDENTIFIER
46	:	DLM_CODEBLK
46	for	KW_LOOP_FOR
46	(	DLM_LPRN
46	int	KW_DATA_INT
46	i	IDENTIFIER
46	=	OP_ASGN
46	5	LIT_INT
46	;	DLM_TRMNTR
46	i	IDENTIFIER
46	<	OP_LSS_THAN
46	10	LIT_INT
46	;	DLM_TRMNTR
46	i	IDENTIFIER
46	--	OP_DECR

LINE	LEXEME	TOKEN
46	)	DLM_RPRN
46	:	DLM_CODEBLK
46	\n	WHT_NEWLINE
47	{	DLM_LCURLY
47	\n	WHT_NEWLINE
48	break	KW_BREAK
48	Identifier	IDENTIFIER
48	;	DLM_TRMNTR
48	\n	WHT_NEWLINE
49	}	DLM_RCURLY
49	\n	WHT_NEWLINE
50	\n	WHT_NEWLINE
51	while	KW_LOOP_WHILE
51	(	DLM_LPRN
51	j	IDENTIFIER
51	==	OP_EQ
51	5	LIT_INT
51	)	DLM_RPRN
51	:	DLM_CODEBLK
51	\n	WHT_NEWLINE
52	{	DLM_LCURLY
52	\n	WHT_NEWLINE
53	x	IDENTIFIER
53	+=	OP_ADD_ASGN
53	5	LIT_INT
53	;	DLM_TRMNTR
53	\n	WHT_NEWLINE
54	}	DLM_RCURLY
54	\n	WHT_NEWLINE
55	\n	WHT_NEWLINE

LINE	LEXEME	TOKEN
56	do	KW_LOOP_DO
56	:	DLM_CODEBLK
56	\n	WHT_NEWLINE
57	{	DLM_LCURLY
57	\n	WHT_NEWLINE
58	pass	KW_PASS
58	;	DLM_TRMNTR
58	\n	WHT_NEWLINE
59	break	KW_BREAK
59	IDETIFIETR	IDENTIFIER
59	;	DLM_TRMNTR
59	\n	WHT_NEWLINE
60	break	KW_BREAK
60	;	DLM_TRMNTR
60	\n	WHT_NEWLINE
61	}	DLM_RCURLY
61	\n	WHT_NEWLINE
62	until	KW_LOOP_UNTIL
62	(	DLM_LPRN
62	x	IDENTIFIER
62	>	OP_GRTR_THAN
62	5	LIT_INT
62	)	DLM_RPRN
62	;	DLM_TRMNTR
62	\n	WHT_NEWLINE
63	\n	WHT_NEWLINE
64	foreach	KW_LOOP_FOREACH
64	(	DLM_LPRN
64	int	KW_DATA_INT
64	i	IDENTIFIER

LINE	LEXEME	TOKEN
64	in	KW_LOOP_IN
64	list1	IDENTIFIER
64	)	DLM_RPRN
64	:	DLM_CODEBLK
64	\n	WHT_NEWLINE
65	{	DLM_LCURLY
65	\n	WHT_NEWLINE
66	--	OP_DECR
66	x	IDENTIFIER
66	;	DLM_TRMNTR
66	\n	WHT_NEWLINE
67	}	DLM_RCURLY
67	\n	WHT_NEWLINE
68	\n	WHT_NEWLINE
69	func	KW_FUNC
69	FUNC1	IDENTIFIER
69	(	DLM_LPRN
69	)	DLM_RPRN
69	:	DLM_CODEBLK
69	\n	WHT_NEWLINE
70	{	DLM_LCURLY
70	\n	WHT_NEWLINE
71	a	IDENTIFIER
71	=	OP_ASGN
71	'c'	LIT_CHAR
71	;	DLM_TRMNTR
71	\n	WHT_NEWLINE
72	}	DLM_RCURLY
72	\n	WHT_NEWLINE
73	\n	WHT_NEWLINE

LINE	LEXEME	TOKEN
74	private	KW_PRIV
74	func	KW_FUNC
74	FUNC2	IDENTIFIER
74	(	DLM_LPRN
74	int	KW_DATA_INT
74	a	IDENTIFIER
74	,	DLM_SPRTR
74	str	KW_DATA_STR
74	b	IDENTIFIER
74	,	DLM_SPRTR
74	list	KW_DATA_LIST
74	c	IDENTIFIER
74	)	DLM_RPRN
74	:	DLM_CODEBLK
74	\n	WHT_NEWLINE
75	{	DLM_LCURLY
75	\n	WHT_NEWLINE
76	protected	KW_PROT
76	int	KW_DATA_INT
76	d	IDENTIFIER
76	=	OP_ASGN
76	3	LIT_INT
76	;	DLM_TRMNTR
76	\n	WHT_NEWLINE
77	}	DLM_RCURLY
77	\n	WHT_NEWLINE
78	\n	WHT_NEWLINE
79	try	KW_TRY
79	:	DLM_CODEBLK
79	\n	WHT_NEWLINE

LINE	LEXEME	TOKEN
80	{	DLM_LCURLY
80	\n	WHT_NEWLINE
81	x	IDENTIFIER
81	=	OP_ASGN
81	(	DLM_LPRN
81	int	KW_DATA_INT
81	)	DLM_RPRN
81	y	IDENTIFIER
81	;	DLM_TRMNTR
81	\n	WHT_NEWLINE
82	}	DLM_RCURLY
82	\n	WHT_NEWLINE
83	catch	KW_CATCH
83	(	DLM_LPRN
83	Exception	IDENTIFIER
83	)	DLM_RPRN
83	:	DLM_CODEBLK
83	\n	WHT_NEWLINE
84	{	DLM_LCURLY
84	\n	WHT_NEWLINE
85	repeat	KW_LOOP_REPEAT
85	(	DLM_LPRN
85	10	LIT_INT
85	)	DLM_RPRN
85	:	DLM_CODEBLK
85	\n	WHT_NEWLINE
86	{	DLM_LCURLY
86	\n	WHT_NEWLINE
87	repeat	KW_LOOP_REPEAT
87	(	DLM_LPRN

LINE	LEXEME	TOKEN
87	10	LIT_INT
87	)	DLM_RPRN
87	:	DLM_CODEBLK
87	\n	WHT_NEWLINE
88	{	DLM_LCURLY
88	\n	WHT_NEWLINE
89	repeat	KW_LOOP_REPEAT
89	(	DLM_LPRN
89	10	LIT_INT
89	)	DLM_RPRN
89	:	DLM_CODEBLK
89	\n	WHT_NEWLINE
90	{	DLM_LCURLY
90	\n	WHT_NEWLINE
91	display	KW_DISPLAY
91	(	DLM_LPRN
91	"Hello"	LIT_STR
91	)	DLM_RPRN
91	;	DLM_TRMNTR
91	\n	WHT_NEWLINE
92	}	DLM_RCURLY
92	\n	WHT_NEWLINE
93	}	DLM_RCURLY
93	\n	WHT_NEWLINE
94	}	DLM_RCURLY
94	\n	WHT_NEWLINE
95	}	DLM_RCURLY
95	\n	WHT_NEWLINE
96	finally	KW_FINALLY
96	:	DLM_CODEBLK

LINE	LEXEME	TOKEN
96	\n	WHT_NEWLINE
97	{	DLM_LCURLY
97	\n	WHT_NEWLINE
98	x	IDENTIFIER
98	=	OP_ASGN
98	5	LIT_INT
98	;	DLM_TRMNTR
98	\n	WHT_NEWLINE
99	}	DLM_RCURLY
99	\n	WHT_NEWLINE
100	\n	WHT_NEWLINE
101	info	KW_INFO
101	(	DLM_LPRN
101	display	KW_DISPLAY
101	)	DLM_RPRN
101	;	DLM_TRMNTR
101	\n	WHT_NEWLINE
102	info	KW_INFO
102	(	DLM_LPRN
102	Testing	IDENTIFIER
102	)	DLM_RPRN
102	;	DLM_TRMNTR
102	\n	WHT_NEWLINE
103	swap	KW_SWAP
103	(	DLM_LPRN
103	a	IDENTIFIER
103	,	DLM_SPRTR
103	b	IDENTIFIER
103	,	DLM_SPRTR
103	c	IDENTIFIER

LINE	LEXEME	TOKEN
103	)	DLM_RPRN
103	;	DLM_TRMNTR
103	\n	WHT_NEWLINE
104	swap	KW_SWAP
104	(	DLM_LPRN
104	d	IDENTIFIER
104	,	DLM_SPRTR
104	e	IDENTIFIER
104	)	DLM_RPRN
104	;	DLM_TRMNTR
104	\n	WHT_NEWLINE
105	check	KW_CHECK
105	(	DLM_LPRN
105	value	IDENTIFIER
105	.	DLM_OBJECT
105	Value	IDENTIFIER
105	)	DLM_RPRN
105	;	DLM_TRMNTR
105	\n	WHT_NEWLINE
106	check	KW_CHECK
106	(	DLM_LPRN
106	TestClass	IDENTIFIER
106	.	DLM_OBJECT
106	sayName	IDENTIFIER
106	(	DLM_LPRN
106	name	IDENTIFIER
106	)	DLM_RPRN
106	)	DLM_RPRN
106	;	DLM_TRMNTR
106	\n	WHT_NEWLINE

LINE	LEXEME	TOKEN
107	int	KW_DATA_INT
107	a	IDENTIFIER
107	=	OP_ASGN
107	FuncDec	IDENTIFIER
107	(	DLM_LPRN
107	1	LIT_INT
107	)	DLM_RPRN
107	;	DLM_TRMNTR
107	\n	WHT_NEWLINE
108	FuncDec	IDENTIFIER
108	(	DLM_LPRN
108	1	LIT_INT
108	)	DLM_RPRN
108	;	DLM_TRMNTR
108	\n	WHT_NEWLINE
109	}	DLM_RCURLY
109	\n	WHT_NEWLINE
110	\n	WHT_NEWLINE
111	class	KW_CLASS
111	TestClass	IDENTIFIER
111	:	DLM_CODEBLK
111	\n	WHT_NEWLINE
112	{	DLM_LCURLY
112	\n	WHT_NEWLINE
113	class1var	IDENTIFIER
113	=	OP_ASGN
113	"This is a test string"	LIT_STR
113	;	DLM_TRMNTR
113	\n	WHT_NEWLINE
114	class1var2	IDENTIFIER

LINE	LEXEME	TOKEN
114	=	OP_ASGN
114	10.5f	LIT_FLT
114	;	DLM_TRMNTR
114	\n	WHT_NEWLINE
115	}	DLM_RCURLY
115	\n	WHT_NEWLINE
116	\n	WHT_NEWLINE
117	class	KW_CLASS
117	SayHi	IDENTIFIER
117	(	DLM_LPRN
117	Parent	IDENTIFIER
117	)	DLM_RPRN
117	:	DLM_CODEBLK
117	\n	WHT_NEWLINE
118	{	DLM_LCURLY
118	\n	WHT_NEWLINE
119	init	KW_INIT
119	(	DLM_LPRN
119	this	KW_THIS
119	,	DLM_SPRTR
119	str	KW_DATA_STR
119	name	IDENTIFIER
119	,	DLM_SPRTR
119	str	KW_DATA_STR
119	message	IDENTIFIER
119	)	DLM_RPRN
119	:	DLM_CODEBLK
119	\n	WHT_NEWLINE
120	{	DLM_LCURLY
120	\n	WHT_NEWLINE

LINE	LEXEME	TOKEN
121	this	KW_THIS
121	.	DLM_OBJECT
121	name	IDENTIFIER
121	=	OP_ASGN
121	name	IDENTIFIER
121	;	DLM_TRMNTR
121	\n	WHT_NEWLINE
122	this	KW_THIS
122	.	DLM_OBJECT
122	message	IDENTIFIER
122	=	OP_ASGN
122	message	IDENTIFIER
122	;	DLM_TRMNTR
122	\n	WHT_NEWLINE
123	}	DLM_RCURLY
123	\n	WHT_NEWLINE
124	\n	WHT_NEWLINE
125	\n	WHT_NEWLINE
126	public	KW_PUB
126	func	KW_FUNC
126	sayName	IDENTIFIER
126	(	DLM_LPRN
126	this	KW_THIS
126	)	DLM_RPRN
126	:	DLM_CODEBLK
126	\n	WHT_NEWLINE
127	{	DLM_LCURLY
127	\n	WHT_NEWLINE
128	display	KW_DISPLAY
128	(	DLM_LPRN

LINE	LEXEME	TOKEN
128	this	KW_THIS
128	.	DLM_OBJECT
128	name	IDENTIFIER
128	)	DLM_RPRN
128	;	DLM_TRMNTR
128	\n	WHT_NEWLINE
129	for	KW_LOOP_FOR
129	(	DLM_LPRN
129	i	IDENTIFIER
129	=	OP_ASGN
129	5	LIT_INT
129	;	DLM_TRMNTR
129	(	DLM_LPRN
129	i	IDENTIFIER
129	<	OP_LSS_THAN
129	5	LIT_INT
129	)	DLM_RPRN
129	and	KW_BOOL_AND
129	(	DLM_LPRN
129	i	IDENTIFIER
129	>	OP_GRTR_THAN
129	10	LIT_INT
129	)	DLM_RPRN
129	;	DLM_TRMNTR
129	i	IDENTIFIER
129	+=	OP_ADD_ASGN
129	10	LIT_INT
129	)	DLM_RPRN
129	:	DLM_CODEBLK
129	\n	WHT_NEWLINE

LINE	LEXEME	TOKEN
130	{	DLM_LCURLY
130	\n	WHT_NEWLINE
131	return	KW_RETURN
131	5	LIT_INT
131	+	OP_PLUS
131	3	LIT_INT
131	;	DLM_TRMNTR
131	\n	WHT_NEWLINE
132	}	DLM_RCURLY
132	\n	WHT_NEWLINE
133	}	DLM_RCURLY
133	\n	WHT_NEWLINE
134	\n	WHT_NEWLINE
135	public	KW_PUB
135	func	KW_FUNC
135	sayMes	IDENTIFIER
135	(	DLM_LPRN
135	this	KW_THIS
135	,	DLM_SPRTR
135	str	KW_DATA_STR
135	name	IDENTIFIER
135	)	DLM_RPRN
135	:	DLM_CODEBLK
135	\n	WHT_NEWLINE
136	{	DLM_LCURLY
136	\n	WHT_NEWLINE
137	display	KW_DISPLAY
137	(	DLM_LPRN
137	this	KW_THIS
137	.	DLM_OBJECT

LINE	LEXEME	TOKEN
137	message	IDENTIFIER
137	+	OP_PLUS
137	name	IDENTIFIER
137	)	DLM_RPRN
137	;	DLM_TRMNTR
137	\n	WHT_NEWLINE
138	}	DLM_RCURLY
138	\n	WHT_NEWLINE
139	}	DLM_RCURLY
139	\n	WHT_NEWLINE
140	\n	WHT_NEWLINE
141	obj	KW_DATA_OBJ
141	NewObj	IDENTIFIER
141	=	OP_ASGN
141	SayHi	IDENTIFIER
141	(	DLM_LPRN
141	"Gheru"	LIT_STR
141	,	DLM_SPRTR
141	"Hello "	LIT_STR
141	)	DLM_RPRN
141	;	DLM_TRMNTR