LEXER Test Cases & Results

Successful Lex Code Block	Log Results
{}$	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_CLOSING_BRACE [}] on line 1... LEXER --> | T_EOPS [$] on line 1... Lex Completed With 0 WARNING(S) and 0 ERROR(S)...
{ **print("there is no spoon")** **}$**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_PRINT [print] on line 2... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 2... LEXER --> | T_QUOTE ["] on line 2... LEXER --> | T_CHAR [t] on line 2... LEXER --> | T_CHAR [h] on line 2... LEXER --> | T_CHAR [e] on line 2... LEXER --> | T_CHAR [r] on line 2... LEXER --> | T_CHAR [e] on line 2... LEXER --> | T_WHITE_SPACE [ ] on line 2... LEXER --> | T_CHAR [i] on line 2... LEXER --> | T_CHAR [s] on line 2... LEXER --> | T_WHITE_SPACE [ ] on line 2... LEXER --> | T_CHAR [n] on line 2... LEXER --> | T_CHAR [o] on line 2... LEXER --> | T_WHITE_SPACE [ ] on line 2... LEXER --> | T_CHAR [s] on line 2... LEXER --> | T_CHAR [p] on line 2... LEXER --> | T_CHAR [o] on line 2... LEXER --> | T_CHAR [o] on line 2... LEXER --> | T_CHAR [n] on line 2... LEXER --> | T_QUOTE ["] on line 2... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 2... LEXER --> | T_CLOSING_BRACE [}] on line 3... LEXER --> | T_EOPS [$] on line 3... Lex Completed With 0 WARNING(S) and 0 ERROR(S)...
{ **print((false == true))** **print((true != true))** **print((false != false))** **print((false != true))** **}**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_PRINT [print] on line 2... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 2... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 2... LEXER --> | T_BOOLEAN_VALUE [false] on line 2... LEXER --> | T_EQUALITY_OP [==] on line 2... LEXER --> | T_BOOLEAN_VALUE [true] on line 2... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 2... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 2...

	LEXER --> | T_PRINT [print] on line 3... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 3... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 3... LEXER --> | T_BOOLEAN_VALUE [true] on line 3... LEXER --> | T_INEQUALITY_OP [!=] on line 3... LEXER --> | T_BOOLEAN_VALUE [true] on line 3... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 3... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 3... LEXER --> | T_PRINT [print] on line 4... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 4... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 4... LEXER --> | T_BOOLEAN_VALUE [false] on line 4... LEXER --> | T_INEQUALITY_OP [!=] on line 4... LEXER --> | T_BOOLEAN_VALUE [false] on line 4... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 4... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 4... LEXER --> | T_PRINT [print] on line 5... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 5... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 5... LEXER --> | T_BOOLEAN_VALUE [false] on line 5... LEXER --> | T_INEQUALITY_OP [!=] on line 5... LEXER --> | T_BOOLEAN_VALUE [true] on line 5... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 5... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 5... LEXER --> | T_CLOSING_BRACE [}] on line 6... LEXER --> | WARNING! NO EOPS [$] detected. Added to end-of-file at line 6... Lex Completed With 1 WARNING(S) and 0 ERROR(S)...
{ **int a** **int b** **a = 0** **b = 0** **while (a != 3) {** **print(a)** **while (b != 3) {** **print(b)** **b = 1 + b** **if (b == 2) {** **print("there is no spoon")** **}** **}** **b = 0**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_VARIABLE_TYPE [int] on line 2... LEXER --> | T_ID [a] on line 2... LEXER --> | T_VARIABLE_TYPE [int] on line 3... LEXER --> | T_ID [b] on line 3... LEXER --> | T_ID [a] on line 5... LEXER --> | T_ASSIGNMENT_OP [=] on line 5... LEXER --> | T_DIGIT [0] on line 5... LEXER --> | T_ID [b] on line 6... LEXER --> | T_ASSIGNMENT_OP [=] on line 6... LEXER --> | T_DIGIT [0] on line 6... LEXER --> | T_WHILE [while] on line 8... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 8... LEXER --> | T_ID [a] on line 8... LEXER --> | T_INEQUALITY_OP [!=] on line 8... LEXER --> | T_DIGIT [3] on line 8... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 8...

a = 1 + a **}** **}$**	LEXER --> | T_OPENING_BRACE [{] on line 8... LEXER --> | T_PRINT [print] on line 9... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 9... LEXER --> | T_ID [a] on line 9... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 9... LEXER --> | T_WHILE [while] on line 10... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 10... LEXER --> | T_ID [b] on line 10... LEXER --> | T_INEQUALITY_OP [!=] on line 10... LEXER --> | T_DIGIT [3] on line 10... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 10... LEXER --> | T_OPENING_BRACE [{] on line 10... LEXER --> | T_PRINT [print] on line 11... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 11... LEXER --> | T_ID [b] on line 11... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 11... LEXER --> | T_ID [b] on line 12... LEXER --> | T_ASSIGNMENT_OP [=] on line 12... LEXER --> | T_DIGIT [1] on line 12... LEXER --> | T_ADDITION_OP [+] on line 12... LEXER --> | T_ID [b] on line 12... LEXER --> | T_IF [if] on line 13... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 13... LEXER --> | T_ID [b] on line 13... LEXER --> | T_EQUALITY_OP [==] on line 13... LEXER --> | T_DIGIT [2] on line 13... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 13... LEXER --> | T_OPENING_BRACE [{] on line 13... LEXER --> | T_PRINT [print] on line 14... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 14... LEXER --> | T_QUOTE ["] on line 14... LEXER --> | T_CHAR [t] on line 14... LEXER --> | T_CHAR [h] on line 14... LEXER --> | T_CHAR [e] on line 14... LEXER --> | T_CHAR [r] on line 14... LEXER --> | T_CHAR [e] on line 14... LEXER --> | T_WHITE_SPACE [ ] on line 14... LEXER --> | T_CHAR [i] on line 14... LEXER --> | T_CHAR [s] on line 14... LEXER --> | T_WHITE_SPACE [ ] on line 14... LEXER --> | T_CHAR [n] on line 14... LEXER --> | T_CHAR [o] on line 14... LEXER --> | T_WHITE_SPACE [ ] on line 14... LEXER --> | T_CHAR [s] on line 14... LEXER --> | T_CHAR [p] on line 14... LEXER --> | T_CHAR [o] on line 14... LEXER --> | T_CHAR [o] on line 14... LEXER --> | T_CHAR [n] on line 14...

	LEXER --> | T_QUOTE ["] on line 14... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 14... LEXER --> | T_CLOSING_BRACE [}] on line 15... LEXER --> | T_CLOSING_BRACE [}] on line 16... LEXER --> | T_ID [b] on line 18... LEXER --> | T_ASSIGNMENT_OP [=] on line 18... LEXER --> | T_DIGIT [0] on line 18... LEXER --> | T_ID [a] on line 19... LEXER --> | T_ASSIGNMENT_OP [=] on line 19... LEXER --> | T_DIGIT [1] on line 19... LEXER --> | T_ADDITION_OP [+] on line 19... LEXER --> | T_ID [a] on line 19... LEXER --> | T_CLOSING_BRACE [}] on line 20... LEXER --> | T_CLOSING_BRACE [}] on line 21... LEXER --> | T_EOPS [$] on line 21... Lex Completed With 0 WARNING(S) and 0 ERROR(S)...
{ **int a** **a = 1** **}**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_VARIABLE_TYPE [int] on line 2... LEXER --> | T_ID [a] on line 2... LEXER --> | T_ID [a] on line 3... LEXER --> | T_ASSIGNMENT_OP [=] on line 3... LEXER --> | T_DIGIT [1] on line 3... LEXER --> | T_CLOSING_BRACE [}] on line 4... LEXER --> | WARNING! NO EOPS [$] detected. Added to end-of-file at line 4... Lex Completed With 1 WARNING(S) and 0 ERROR(S)...
{ **int a** **a = 1** **if(a == 1) {** **a = 2** **}** **if(a != 1) {** **a = 3** **}** **} $**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_VARIABLE_TYPE [int] on line 2... LEXER --> | T_ID [a] on line 2... LEXER --> | T_ID [a] on line 3... LEXER --> | T_ASSIGNMENT_OP [=] on line 3... LEXER --> | T_DIGIT [1] on line 3... LEXER --> | T_IF [if] on line 5... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 5... LEXER --> | T_ID [a] on line 5... LEXER --> | T_EQUALITY_OP [==] on line 5... LEXER --> | T_DIGIT [1] on line 5... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 5... LEXER --> | T_OPENING_BRACE [{] on line 5... LEXER --> | T_ID [a] on line 6... LEXER --> | T_ASSIGNMENT_OP [=] on line 6... LEXER --> | T_DIGIT [2] on line 6...

	LEXER --> | T_CLOSING_BRACE [}] on line 7... LEXER --> | T_IF [if] on line 9... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 9... LEXER --> | T_ID [a] on line 9... LEXER --> | T_INEQUALITY_OP [!=] on line 9... LEXER --> | T_DIGIT [1] on line 9... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 9... LEXER --> | T_OPENING_BRACE [{] on line 9... LEXER --> | T_ID [a] on line 10... LEXER --> | T_ASSIGNMENT_OP [=] on line 10... LEXER --> | T_DIGIT [3] on line 10... LEXER --> | T_CLOSING_BRACE [}] on line 11... LEXER --> | T_CLOSING_BRACE [}] on line 12... LEXER --> | T_EOPS [$] on line 12... Lex Completed With 0 WARNING(S) and 0 ERROR(S)...
{ **print(while)** **}$**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_PRINT [print] on line 2... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 2... LEXER --> | T_WHILE [while] on line 2... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 2... LEXER --> | T_CLOSING_BRACE [}] on line 3... LEXER --> | T_EOPS [$] on line 3... Lex Completed With 0 WARNING(S) and 0 ERROR(S)...
{ **print("while")** **}$**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_PRINT [print] on line 2... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 2... LEXER --> | T_QUOTE ["] on line 2... LEXER --> | T_CHAR [w] on line 2... LEXER --> | T_CHAR [h] on line 2... LEXER --> | T_CHAR [i] on line 2... LEXER --> | T_CHAR [l] on line 2... LEXER --> | T_CHAR [e] on line 2... LEXER --> | T_QUOTE ["] on line 2... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 2... LEXER --> | T_CLOSING_BRACE [}] on line 3... LEXER --> | T_EOPS [$] on line 3... Lex Completed With 0 WARNING(S) and 0 ERROR(S)...

Failed Lex Code Block	Log Results
{}$	Beginning Lexing Session... *Stings Treated As CharList*

{{{{{{}}}}}}$** **{{{{{{}}}}}}$ **{int @}$**	LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_CLOSING_BRACE [}] on line 1... LEXER --> | T_EOPS [$] on line 1... LEXER --> | T_OPENING_BRACE [{] on line 2... LEXER --> | T_OPENING_BRACE [{] on line 2... LEXER --> | T_OPENING_BRACE [{] on line 2... LEXER --> | T_OPENING_BRACE [{] on line 2... LEXER --> | T_OPENING_BRACE [{] on line 2... LEXER --> | T_OPENING_BRACE [{] on line 2... LEXER --> | T_CLOSING_BRACE [}] on line 2... LEXER --> | T_CLOSING_BRACE [}] on line 2... LEXER --> | T_CLOSING_BRACE [}] on line 2... LEXER --> | T_CLOSING_BRACE [}] on line 2... LEXER --> | T_CLOSING_BRACE [}] on line 2... LEXER --> | T_CLOSING_BRACE [}] on line 2... LEXER --> | T_EOPS [$] on line 2... LEXER --> | T_OPENING_BRACE [{] on line 3... LEXER --> | T_OPENING_BRACE [{] on line 3... LEXER --> | T_OPENING_BRACE [{] on line 3... LEXER --> | T_OPENING_BRACE [{] on line 3... LEXER --> | T_OPENING_BRACE [{] on line 3... LEXER --> | T_OPENING_BRACE [{] on line 3... LEXER --> | T_CLOSING_BRACE [}] on line 3... LEXER --> | T_CLOSING_BRACE [}] on line 3... LEXER --> | T_CLOSING_BRACE [}] on line 3... LEXER --> | T_CLOSING_BRACE [}] on line 3... LEXER --> | T_CLOSING_BRACE [}] on line 3... LEXER --> | T_CLOSING_BRACE [}] on line 3... LEXER --> | T_EOPS [$] on line 3... LEXER --> | T_OPENING_BRACE [{] on line 4... LEXER --> | T_VARIABLE_TYPE [int] on line 4... LEXER --> | ERROR! Unrecognized or Invalid Token [@] on line 4 Lex Failed With 0 WARNING(S) and 1 ERROR(S)...
{ **print("$")** **}**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_PRINT [print] on line 2... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 2... LEXER --> | ERROR! Unrecognized or Invalid Token ["$"] on line 2 Lex Failed With 0 WARNING(S) and 1 ERROR(S)...
adsfadafd	Beginning Lexing Session... *Stings Treated As CharList*

	LEXER --> | ERROR! Unrecognized or Invalid Token [adsfadafd] on line 1 LEXER --> | ERROR! Input did not generate valid Token Array... Lex Failed With 0 WARNING(S) and 2 ERROR(S)...
{ **print("while + true")** **}$**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_PRINT [print] on line 2... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 2... LEXER --> | ERROR! Unrecognized or Invalid Token ["while + true"] on line 2 Lex Failed With 0 WARNING(S) and 1 ERROR(S)...
Empty Console	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | ERROR! Empty Input or Only White-Space Detected... Lex Failed With 0 WARNING(S) and 1 ERROR(S)...
{ **string a** **int b** **b = 0** **if (b == 0) {** **a = ["this", "won't", "work"]** **}** **}$**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_VARIABLE_TYPE [string] on line 2... LEXER --> | T_ID [a] on line 2... LEXER --> | T_VARIABLE_TYPE [int] on line 3... LEXER --> | T_ID [b] on line 3... LEXER --> | T_ID [b] on line 5... LEXER --> | T_ASSIGNMENT_OP [=] on line 5... LEXER --> | T_DIGIT [0] on line 5... LEXER --> | T_IF [if] on line 7... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 7... LEXER --> | T_ID [b] on line 7... LEXER --> | T_EQUALITY_OP [==] on line 7... LEXER --> | T_DIGIT [0] on line 7... LEXER --> | T_RIGHT_PARENTHESIS [ )] on line 7... LEXER --> | T_OPENING_BRACE [{] on line 7... LEXER --> | T_ID [a] on line 8... LEXER --> | T_ASSIGNMENT_OP [=] on line 8... LEXER --> | ERROR! Unrecognized or Invalid Token [[ ] on line 8 Lex Failed With 0 WARNING(S) and 1 ERROR(S)...
{ **print("There")** **}$**	Beginning Lexing Session... *Stings Treated As CharList* LEXER --> | T_OPENING_BRACE [{] on line 1... LEXER --> | T_PRINT [print] on line 2... LEXER --> | T_OPENING_PARENTHESIS [( ] on line 2...

	LEXER --> | ERROR! Unrecognized or Invalid Token ["There"] on line 2 Lex Failed With 0 WARNING(S) and 1 ERROR(S)...