

1. public String get_token(BufferedReader br)

Test #	Test Path	Test Data	Expected Output
1	[START, 1, 2, END]	Empty input	null
2	[START, 1, 3, 4, 6, 8, 9, END]	"("	"("
3	[START, 1, 3, 4, 6, 10, 12, 14, 16, 17, 21, 22, END]	"ABC"	"ABC"
4	[START, 1, 3, 4, 6, 10, 14, 16, 17, 18, 19, 23, 24, END]	"123"	"123"
5	[START, 1, 3, 4, 6, 10, 14, 16, 20, END]	"#c"	"#c"
6	[START, 1, 3, 4, 6, 10, 14, 16, 17, 21, 25, 26, END]	"; comment"	"; comment"
7	[START, 1, 3, 4, 6, 10, 14, 16, 17, 21, 25, 27, 28, END]	"\"hello\""	"\"hello\""
8	[START, 1, 3, 4, 6, 10, 14, 16, 17, 21, 29, END]	"#INVALID"	"error"
9	[START, 1, 3, 5, END]	Newline/Whitespace	Skips whitespace, processes next token.
10	[START, 1, 3, 4, 6, 10, 14, 16, 17, 21, 25, 26, END]	Invalid comment like "#,"	"error"

2. public static boolean is_token_end(int str_com_id, int res)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 2, End]	str_com_id = 0, res = -1	T
2	[Start, 1, 3, 4, End]	str_com_id = 1, res = ""	T
3	[Start, 1, 3, 6, End]	str_com_id = 1, res = 'A'	F
4	[Start, 1, 7, 8, 9, End]	str_com_id = 2, res = '\n'	T
5	[Start, 1, 7, 8, 10, End]	str_com_id = 2, res = 'A'	F
6	[Start, 1, 11, 12, End]	str_com_id = 0, res = '('	T
7	[Start, 1, 11, 13, 14, End]	str_com_id = 0, res = ' '	T
8	[Start, 1, 11, 15, End]	str_com_id = 0, res = 'A'	F

3. static int token_type(String tok)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, False, End]	Null	error
2	[Start, 1, True, 2, False, End]	""	error
3	[Start, 1, True, 2, True, 3, True, End]	"and"	keyword
4	[Start, 1, True, 2, True, 3, False, 4, True, End]	"("	Spec_symbol
5	[Start, 1, True, 2, True, 3, False, 4, False, 5, True, End]	"variable"	indentifier
6	[Start, 1, True, 2, True, 3, False, 4, False, 5, False, 6, True, End]	"123"	Num_constant

7	[Start, 1, True, 2, True, 3, False, 4, False, 5, False, 6, False, 7, True, End]	"\"string\""	Str_constant
8	[Start, 1, True, 2, True, 3, False, 4, False, 5, False, 6, False, 7, False, 8, True, End]	"#a"	Char_constant
9	[Start, 1, True, 2, True, 3, False, 4, False, 5, False, 6, False, 7, False, 8, False, 9, True, End]	";comment"	comment
10	[Start, 1, True, 2, True, 3, False, 4, False, 5, False, 6, False, 7, False, 8, False, 9, False, End]	Invalid token	err

4. void print_token(String tok)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 2, End]	invalid	error,"invalid"
2	[Start, 1, 3, 4, End]	"and"	keyword,"and"
3	[Start, 1, 3, 5, 6, End]	"("	lparen.
4	[Start, 1, 3, 7, 8, End]	"variable"	identifier,"variable".
5	[Start, 1, 3, 9, 10, End]	"123"	numeric,123.
6	[Start, 1, 3, 11, 12, End]	"\"string\""	string,"hello".
7	[Start, 1, 3, 13, 14, 15, 16, End]	"#a"	character,"a".
8	[Start, 1, 3, 13, 14, 15, End]	"#"	Char_constant
9	[Start, 1, 3, 13, 14, 15, End]	"#invalid"	character,"i".
10	[Start, 1, 3, 17, End]	":comment"	comment,";comment".

5. public static boolean is_comment(String ident)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 3, End]	Null	F
2	[Start, 1, 3, End]	" "	F
3	[Start, 1, 2, End]	"comment"	T
4	[Start, 1, 3, End]	"not a comment"	F

6. public static boolean is_keyword(String str)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 3, End]	null	F
2	[Start, 1, 2, End]	"and"	T
3	[Start, 1, 2, End]	"or"	T
4	[Start, 1, 2, End]	"if"	T
5	[Start, 1, 2, End]	"xor"	T

Cody Jones
1001986663
Test Cases

6	[Start, 1, 2, End]	"lambda"	T
7	[Start, 1, 2, End]	"=>"	T
8	[Start, 1, 3, End]	"else"	F
9	[Start, 1, 3, End]	"random"	F

7. public static boolean is_char_constant(String str)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 3, End]	null	F
2	[Start, 1, 3, End]	"a"	F
3	[Start, 1, 2, End]	"#a"	T
4	[Start, 1, 3, End]	"#1"	F
5	[Start, 1, 3, End]	"#abc"	F
6	[Start, 1, 3, End]	"#"	F
7	[Start, 1, 3, End]	null	F

8. public static boolean is_num_constant(String str)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 7, End]	null	F
2	[Start, 1, 7, End]	""	F
3	[Start, 1, 2, 3, 5, End]	"abc"	F
4	[Start, 1, 2, 3, 5, End]	"12a3"	F
5	[Start, 1, 2, 3, 4, 6, End]	"123"	T

9. public static boolean is_str_constant(String str)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 7, End]	null	F
2	[Start, 1, 7, End]	""	F
3	[Start, 1, 2, 3, 5, End]	"abc"	F
4	[Start, 1, 2, 3, 5, End]	"abc"	F
5	[Start, 1, 2, 3, 4, 6, End]	"hello"	T
6	[Start, 1, 2, 3, 4, 5, End]	"unescaped"	F
7	[Start, 1, 2, 3, 4, 6, End]	"a string"	T

10. public static boolean is_identifier(String str)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 7, End]	null	F
2	[Start, 1, 7, End]	""	F
3	[Start, 1, 2, 3, 5, End]	"123abc"	F
4	[Start, 1, 2, 3, 4, 6, End]	"validName"	T
5	[Start, 1, 2, 3, 4, 5, End]	"var123!"	F

Cody Jones
1001986663
Test Cases

6	[Start, 1, 2, 3, 4, 6, End]	"abc123"	T
---	-----------------------------	----------	---

11. static String getSpecSymbolOutput(String str)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 2, End]	("lparen."
2	[Start, 1, 3, 4, End])	"rparen."
3	[Start, 1, 5, 6, End]	["lsquare."
4	[Start, 1, 7, 8, End]]	"rsquare."
5	[Start, 1, 9, 10, End]	'	"quote."
6	[Start, 1, 11, 12, End]	`	"bquote."
7	[Start, 1, 13, 14, End]	,	"comma."
8	[Start, 1, 15, 16, End]	/	"slash."
9	[Start, 1, 17, End]	@	"error,\"@\"."

12. public static void main(String[] args)

Test #	Test Path	Test Data	Expected Output
1	[Start, 1, 2, End]	args = {}	None (Reads from standard input)
2	[Start, 1, 3, 4, 6, 7, 8, 9, 10, 8, 11, End]	args = {"validFile.txt"}	Outputs tokens from validFile.txt
3	[Start, 1, 3, 5, End]	args = {"file1", "file2"}	"Error! Please give the token stream\n"
4	[Start, 1, 3, 4, 6, 7, 8, 11, End]	args = {"emptyFile.txt"}	None (Handles empty input file gracefully)