Test Cases and Test paths

_____

End to End test is at the bottom

Unit Test

_____

_____

Test Paths for main

_____

1) main[1, 2, 6, 7, 8, 9, 10, 11, 12]

Input: String[] args = [], Console Input = "23 #a and ("

Expected Output:

Console Output =

numeric,23.

character,"a".

keyword,"and".

lparen. "

---

2) main[1, 3, 4, 6, 7, 8, 9, 10, 11, 12]

Input: String[] args =[], "D:\MyStuff\Main\College\UTArlington\FALL2022\CSE4321\myInput.txt"

File Format is:

;

(, or `

23 if #a

Expected Output:

Console Output =

comment,"; ".

lparen.

comma.

keyword,"or".

bquote.

numeric,23.

keyword,"if".

character,"a".

---

3) main[1, 3, 5]

Input: String[] args = [], Console Input = ""

Expected Output: Console Output = "[App Warning]:Error! Please give the token stream"

Test Paths for open_token_stream **main function

---

1) open_token_stream[1, 3, 4]

Input: String fname = ""

Expected Output: an instance of buffer reader (br), passed null to its stream.

---

2) open_token_stream[1, 2, 4]

Input: String fname = "D:\MyStuff\Main\College\UTArlington\FALL2022\CSE4321\myInput.txt"

Expected Output: an instance of buffer reader (br), passed the filename to its stream

---

Test Paths for open_character_stream

---

1) open_character_stream[1, 2, 4]

Input: String fname = "null", Console Input = "23 #a and ("

Expected Output: an instance of the buffer reader

---

2) open_character_stream[1, 3, 4]

Input: String fname = "D:\MyStuff\Main\College\UTArlington\FALL2022\CSE4321\myInput.txt"

Expected Output: an instance of buffer reader

'remaining to cover: [26, 31, 32, 33, 37, 38]'

Test Paths for get_token **main function

---

1) get_token[1, 2, 3, 4]

Input: BufferedReader br = ""

Expected Output: Console Output = please provide a valid file path or a token in console!

---

2) get_token[1, 2, 3, 5, 6, 7, 5, 8, 9]

Input: BufferedReader br = " ", (here i will account for space and new line in seperate cases in junit)

Expected Output: Console Output = null

---

3) get_token[1, 2, 3, 5, 8, 10, 11]

Input: BufferedReader br = "("

Expected Output: Console Output = (

---

4) get_token[1, 2, 3, 5, 8, 10, 12, 13, 14, 16, 17, 18, 19]

Input: BufferedReader br = "{\\"

Expected Output: Console Output = {\\

---

5) get_token[1, 2, 3, 5, 8, 10, 12, 13, 14, 16, 17, 18, 19]

Input: BufferedReader br = "'xyz'"

Expected Output: Console Output = 'xyz'

---

6) get_token[1, 2, 3, 5, 8, 10, 12, 14, 15, 16, 17, 18, 19]

Input: BufferedReader br = ";"

Expected Output: Console Output = ;

---

7) get_token[1, 2, 3, 5, 8, 10, 12, 14, 15, 16, 17, 20, 21, 27, 28, 29]

Input: BufferedReader br = "xyz "

Expected Output: Console Output = ;

---

8) get_token[1, 2, 3, 5, 8, 10, 12, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 27, 30, 31, 32]

Input: BufferedReader br = "xyz,"

Expected Output: Console Output = xyz

---

9) get_token[1, 2, 3, 5, 8, 10, 12, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 27, 30, 33, 34, 35]

Input: BufferedReader br = "xyz'"

Expected Output: Console Output = xyz'

WILL USE THIS FOR MY END TO END SINCE IT COVERS THE MOST NODES

---

10) get_token[1, 2, 3, 5, 8, 10, 12, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 27, 30, 33, 36, 37, 38]

Input: BufferedReader br = "abc;"

Expected Output: Console Output = abc

---

11) get_token[1, 2, 3, 5, 8, 10, 12, 14, 15, 16, 17, 20, 21, 22, 23, 24, 25, 27, 30, 33, 36, 39]

Input: BufferedReader br = "{abc\\"

Expected Output: Console Output = {abc\\

Test Paths for get_char

_____

1) get_char[1, 2]

Input: BufferedReader br = ")"

Expected Output: instance of that character

Test Paths for is_spec_symbol

_____

1) is_spec_symbol[1,2, 3, 5, 7, 9, 11, 13, 15]

Input: Char c = (

Expected Output: boolean flag = true

---

2) is_spec_symbol[1, 3, 4, 5, 7, 9, 11, 13, 15]

Input: Char c =)

Expected Output: boolean flag = true

---

3) is_spec_symbol[1,3, 5, 6, 7, 9, 11, 13, 15]

Input: Char c = [

Output: boolean flag = true

---

4) is_spec_symbol[1, 3, 5, 7, 8, 9, 11, 13, 15]

Input: Char c =]

Output: boolean flag = true

---

5) is_spec_symbol[1,3, 5, 7, 9, 10, 11, 13, 15]

Input: Char c = /

Output: boolean flag = true

---

6) is_spec_symbol[1,3, 5, 7, 9, 11, 12, 13, 15]

Input: Char c = `

Output: boolean flag = true

---

7) is_spec_symbol[1, 3, 5, 7, 9, 11, 13, 14, 15]

Input: Char c = ,

Output: boolean flag = true

---

8) is_spec_symbol[1, 3, 5, 7, 9, 11, 13, 15]

Input: Char c = 0

Output: boolean flag = false

Test Paths for unget_char

_____

1)unget_char[1, 2]

Input: BufferedReader br = "(, or ` 23 if #a", ch = or

Expected Output: or

******check id's*****

Test Paths for is_token_end

_____

1)is_token_end[1, 3, 4, 6, 7, 11, 13, 15]

Input: Int str_com_id = "1", Int res = "1"

Expected Output: Boolean flag: false

2)is_token_end[1, 2, 3, 4, 5, 7, 11, 13, 15]

Input: Int str_com_id = "1", Int res = "-1"

Expected Output: Boolean flag: true

---

3)is_token_end[1, 3, 7, 8, 10, 11, 13, 15]

Input: Int str_com_id = "2", Int res = "1"

Expected Output: Boolean flag: false

4)is_token_end[1, 2, 3, 7, 8, 9, 11, 13, 15]

Input: Int str_com_id = "2", Int res = "-1"

Expected Output: Boolean flag: true

---

5)is_token_end[1, 3, 11, 12, 13, 15]

Input: Int str_com_id = "0", Int res = "1"

Expected Output: Boolean flag: false

6)is_token_end[1, 2, 3, 7, 11, 13, 14, 15]

Input: Int str_com_id = "0", Int res = "-1"

Expected Output: Boolean flag: true

Test Paths for print_token

_____

1)print_token[1, 2, 3, 4, 5, 7, 9, 11, 13, 15, 17]

Input: String tok = "$"

Output: error, "$".

---

2)print_token[1, 2, 3, 5, 6, 7, 9, 11, 13, 15, 17]

Input: String tok = "xor"

Output: keyword, "xor".

---

3)print_token[1, 2, 3, 5, 7, 8, 9, 11, 13, 15, 17]

Input: String tok = "`"

Output: bquote

---

4)print_token[1, 2, 3, 5, 7, 9, 11, 13, 15, 17]

Input: String tok = "a2"

Output: identifier, "a2".

---

5)print_token[1, 2, 3, 5, 7, 9, 11, 12, 13, 15, 17]

Input: String tok = "34"

Output: numeric, 34.

---

6)print_token[1, 2, 3, 5, 7, 9, 11, 13, 14, 15, 17]

Input: String tok = ""abc""

Output: string, "abc".

---

7)print_token[1, 2, 3, 5, 7, 9, 11, 13, 15, 16, 17]

Input: String tok = "#f"

Output: character, "f"

---

8)print_token[1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 18]

Input: String tok = ";"

Output: comment, ";".

Test Paths for token_type

---

1)token_type[1, 2]

Input: String tok = "and"

Output: 1

---

2)token_type[1, 3, 4, 5, 7, 9, 11, 13, 15]

Input: String tok = ")"

Output: 2

---

3)token_type[1, 3, 5, 6, 7, 9, 11, 13, 15]

Input: String tok = "a2"

Output: 3

---

4)token_type[1, 3, 5, 7, 8, 9, 11, 13, 15]

Input: String tok = "23"

Output: 41

---

5)token_type[1, 3, 5, 7, 9, 10, 11, 13, 15]

Input: String tok = ""abc""

Output: 42

---

6)token_type[1, 3, 5, 7, 9, 11, 12, 13, 15]

Input: String tok = "#a"

Output: 43

---

7)token_type[1, 3, 5, 7, 9, 11, 12, 13, 14, 15]

Input: String tok = ";"

Output: 5

---

8)token_type[1, 3, 5, 7, 9, 11, 13, 15]

Input: String tok = "^"

Output: 0

Test Paths for is_keyword

_____

1)is_keyword[1, 2, 4]

Input: String str = "and"

Output: Boolean flag = true

---

2)is_keyword[1, 3, 4]

Input: String str = "XXOR"

Output: Boolean flag = false

Test Paths for is_identifier(fault discovered in the true and else)

_____

1)is_identifier[1, 2, 3, 4]

Input: String str = "a2"

Output: Boolean flag = true

2)is_identifier[1, 2, 3, 4, 5, 6]

Input: String str = "2"

Output: Boolean flag = false

3)is_identifier[1, 2, 6]

Input: String str = ""

Output: [App Warning]: no argument is provided!

Test Paths for is_num_const

_____

1)is_num_const[1, 2, 3, 4, 6]

Input: String = "696"

Output: Boolean flag = true

---

2)is_num_const[1, 2, 3, 5, 7]

Input: String = "3a"

Output: Boolean flag = "false"

---

3)is_num_const[1, 7]

Input: String = "a"

Output: Boolean flag = "false"

Test paths for is_str_const

---

1)is_str_const[1, 2, 3, 4, 6]

Input: String = ' "" '

Output: Boolean flag = "true"

---

2)is_str_const[1, 2, 3, 5, 3, 5, 6]

Input: String = ' "abc" '

Output: true

3)is_str_const[1, 2, 7]

Input: String = ' " '

Output: Boolean flag = "false"

---

4)is_str_const[1, 7]

Input: String = " "

Output: Boolean flag = "false"

Test Paths for is_char_cosnt

---

1)is_char_const[1, 2]

Input: String = "#a"

Output: Boolean flag = "true"

---

2)is_char_const[1, 3]

Input: String = "#8"

Output: Boolean flag = "false"

Test Paths for is_comment

---

1)is_comment[1, 2]

Input: String ident = ";"

Output: Boolean flag = "true"

---

2)is_comment[1, 3]

Input: String ident = ":"

Output: Booelan flag = "false"

Test Paths for print_spec_symbol

---

1)print_spec_symbol[1, 2, 3, 5, 7, 9, 11, 13]

Input: String str = "("

Output: Console output = "lparen"

2)print_spec_symbol[1, 3, 4, 5, 7, 9, 11, 13]

Input: String str = ")"

Output: Console output = "rparen"

3)print_spec_symbol[1, 3, 5, 6, 7, 9, 11, 13]

Input: String = "["

Output: Console output = "lsquare"

4)print_spec_symbol[1, 3, 5, 7, 8, 9, 11, 13]

Input: String = "]"

Output: Console output = "rsquare"

5)print_spec_symbol[1, 2, 3, 5, 7, 9, 10, 11, 13]

Input: String str = " ' "

Output: Console output = "quote"

6)print_spec_symbol[1, 3, 5, 7, 9, 11, 12, 13]

Input: String str = " ` "

Output: Console output = "bquote"

7)print_spec_symbol[1, 2, 3, 5, 7, 9, 11, 13, 14]

Input: ,

Output: comma

****

2) get_token[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 21, 22, 23, 24, 25, 27, 28, 29, 30, 33, 34, 35, 36, 39]

****

End to End Test Paths

_____

END TO END Test for Printtokens.java

---

1) main[1, 2, 6, 7->open_token_stream[1,3->open_character_stream[1, 3, 4], 4, 8->get_token[1, 2, 3, 5, 8,

10->is_spec_symbol, 12, 14, 15, 16, 17, 20, 21->is_token_end[1, 3, 7, 8, 10, 11, 13, 15], 22, 23->get_char[1, 2],

24, 25, 27, 30, 33, 36, 37->unget_char[1,2], 38]

Input: String[] args = [], Console Input = "23 "abc" #a and ("

Output:

```
  Console Output =
    numeric,23.
    string,"abc".
    character,"a".
    keyword,"and".
    lparen.
```