****

**CC PROJECT MILESTONE**

**SYNTAX ANALYSIS**

**ASSIGNMENT#3**

**Group Partners:**

|  |  |  |
| --- | --- | --- |
| **NAME** | **SID** | **CID** |
| HAMMAD ABDUL BARI | 10708 | 111707 |
| HAFIZ FARZAN AHMED | 10712 |
| MUHAMMAD BILAL | 10559 |

**Code:**

def perform\_syntax\_analysis(tcl\_code):

lines = tcl\_code.splitlines()

for line in lines:

line = line.strip()

if not line:

continue

command, \*arguments = line.split()

if not command:

return "Invalid TCL syntax: Empty command"

if not command.isidentifier() or command.lower() not in ["puts", "set", "if", "while"]:

return f"Invalid TCL syntax: Invalid command: '{command}'"

for argument in arguments:

if not (argument.startswith('"') and argument.endswith('"')):

return f"Invalid TCL syntax: Invalid argument: '{argument}'"

return "Valid TCL syntax"

tcl\_code = input("Enter TCL code: ")

output = perform\_syntax\_analysis(tcl\_code)

print(output)

**Documentation:**

1. `Perform\_syntax\_analysis` is defined with a single parameter `tcl\_code`, which represents the TCL code that will be analyzed.

2. The TCL\ Input code is split into separate lines using the "splitlines()" method, and the resulting lines are stored in the "lines" variable.

3. "Valid\_commands"list is defined, which contains the commands considered to be valid in TCL code."puts", "Set", "If", and , "while".

4. The code enters a loop that iterates over each line in the 'lines" list.

5. The current line is stripped of leading and trailing whitespace using the `strip()` method.

6. If the line is empty (after stripping), the loop continues to the next iteration using the "continue" statement.

7. The line is split into words using the "split()" method, and the first word (the command) is assigned to the `command` variable. The remaining words (the arguments) are collected into the `arguments` list using the `\*arguments` syntax, which allows collecting any number of remaining values into a list.

8. Syntax analysis is performed on the command and arguments.

If the command is empty, so the function returns the string "Invalid TCL syntax: Empty command".

Our function will be iterate now on each argument in the 'arguments[]' list. If an argument does not start and end with double quotation marks, it means it is an invalid argument, and the function returns a string indicating the invalid argument.

9. If the loop completes without any invalid syntax being found, the function returns the string "Valid TCL syntax".

10. input() stored in the `tcl\_code` variable.

11. `perform\_syntax\_analysis` function is called with the `tcl\_code` variable as an argument, and the result is stored in the `output` variable.

12. `output` variable is printed to display the result of the syntax analysis.



