ASSIGNMENT 1: ANALYSIS USING AST2JSON PACKAGE

PROCESS OF BUILDING THE TOOL:

- Learnt about ***ast2json*** package from this link, about what this package does, how to install it and use it in this tool.
- Made several simple small python code snippets (like x = 5, if(x>y): print(x)) and observed the resulting tree structure and different node attributes. ***Easy to read and understand***
- When tried with lengthy and nested code, resulting ***JSON*** data got ***harder to read and understand*** as there was a lot of data and multi-level nodes.
- So used this website, which takes in ***JSON*** data as input and outputs a ***tree like structure***, which depicts the structure visually, making it ***easier to understand*** on what level that node is, what nodes are its child and the attributes each node has, easing the reading and understanding of data thus making analysis of the ***Abstract Syntax Tree (AST) simple and intuitive***.
- Using the '***_type***' attribute of a node, it determines whether that node represents a '***assignment statement***' or '***branch condition***' or '***loop condition***'
- Based on the ***parent node*** type, child nodes were parsed and visualizing how the desired output looks, different formats was created for different types of child nodes ***(Expr, Call, Iter, elts)***, so as to get an output which is user-friendly.

EXECUTION FLOW OF THE TOOL:

- User inputs the python file path as command line argument to the tool, file is then opened and parsed using parse() function from ***ast*** package and then converted into JSON data using ***ast2json*** function.
- A for loop passes each child of the root node to the ***checker()*** function one at a time
- ***checker()***, using '_type' attribute, determines the statement type and classifies them into one of three categories and appends the generated output to the respective category list.
- Based upon the previously determined type, an output format is created which is populated with values returned by ***typechecker()*** function, to which node's children's appropriate attributes are passed to be read and processed to return the needed data.
- In cases where parent node has statement blocks (if block or For loop block), statements are handled recursively.
- After all required nodes are parsed, list of each type of statement is outputted to the user.

STEPS TO RUN THE TOOL:

1. Check if python packages like ast2json and json are installed properly, if not you can install it by running command "***pip <space> install <space> <package name>***" in any terminal
2. Open any terminal (***Command Prompt, Windows PowerShell, Git Bash***, etc.) and move into the directory where ***run.sh*** is present
3. Type "bash ***./run.sh <space> file path/python file name.py***"
   For example, to run one of the testcases provided in package, type: ***bash ./run.sh <space> ./testcases/test.py***
4. Press Enter
5. Output will be displayed in terminal
6. To see JSON data for debugging, uncomment line "print***(json.dumps(ast,indent=4))***" in source code file, that is ***Assign.py*** in Source directory and save the file and again run the tool, you will be able to see both JSON data and output

HARSH AGARWAL
21111030
MTECH CSE 2021-23
PROGRAM ANALYSIS, VERIFICATION AND TESTING