

**EASTERN MEDITERRANEAN UNIVERSITY**

**COMPUTER ENGINEERING DEPARTMENT**

**SOFTWARE ENGINEERING**

**CMSE 318 – CMPE410**

**ASSIGNMENT 2**

**ASSOC PROF. DR. ZEKİ BAYRAM**

**GROUP 02**

**TOLGAHAN DAYANIKLI – 17330159**

**BERKAN ERGİL – 16000070**

TABLE OF CONTENTS

Table of contents .

lıst of fıgures .

1. ıntroductıon 1

2. project plannıng & management 1

3. ınformatıon about websıtes 2

4. testıng strategy 10

5. test plan 12

6. tests completed 12

7. test plan 12

8. tests ongoıng 12

9. conclusıon 12

10. references 12

LIST OF FIGURES

FIGURE 1 2

FIGURE 2 3

FIGURE 3 3

FIGURE 4 4

FIGURE 5 4

FIGURE 6 4

FIGURE 7 5

FIGURE 8 5

FIGURE 9 6

FIGURE 10 7

FIGURE 11 8

FIGURE 12 9

FIGURE 13 9

FIGURE 14 10

FIGURE 15 10

FIGURE 16 10

FIGURE 17 11

FIGURE 18 11

**1.PROBLEM DEFINITION**

In this assignment, we are asked to write a console program about the lexical analyzer that recognizes integers, floating point numbers, identifiers, logic operators, &, &&, |, ||, as well as reserved words “for”, “while”, “if” and “else”. The program is based on reading the input from a text file and return the object as a result each time it is called. The returned object should have attributes for the token, index into symbol table, value in case of an integer, value in case of a floating-point number, and string of characters in case of an unrecognized lexeme. The program is going to have a menu for the users with the choices of calling lex() function, showing symbol table, and exiting. Exit is supposed the terminate the program. On the other hand, to make the program more realistic while presenting, we added time.sleep(0.5) which is driven from time module. However, we are not going to give any space for that lines in our documentation.

**2. TOOLS & ENVIRONMENT AND DIVISON OF WORK**

**2.1 TOOLS & ENVIRONMENT**

For the assignment, we are informed to use the programming language Python. Nowadays, Python is used for data analysis with the compatible environments and frameworks such as Pandas. However, we are not going to use Pandas since our assignment is not a big project. We are using Visual Studio Code (VS Code) for our programming due to its visual, layout, clean-coding and being user-friendly.

**2.2 DIVISION OF WORK**

Since we are not working on a very big project, there were no specific work division among us. We just shared the functions between us and after that, we bring all of them together. We had couple of meetings online to discuss our priorities, and for project management. We are correcting each other’s mistakes and teaching each other when necessary.

**3. SYSTEM**

As we mentioned before, the system needs to recognize the input weather they are integers or floating-point numbers, or identifiers, or logic operators, or reserved words etc. In this section, we are going to describe our code step by step. Moreover, in the Testing section (*SECTION* *4*), we are going to run our program to test and show what the program does.

**Importing Library**

We are importing “enum” library. This library can be used for creating enumerations, which are set of symbolic names (members) bound to unique, constant values. The members of an enumeration can be compared by these symbolic names, and the enumeration itself can be iterated over. With the aid of enum, we can evaluate string operations, check the types of enum types, etc.



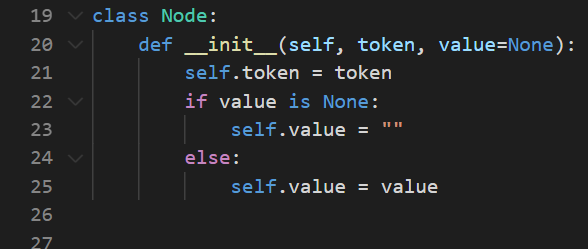
*Figure 1 demonstrates the import of enum library.*

**Class Tokens**

 Tokens class is based on the enum library that we have imported above. With the method (enum.Enum), we can create enumerations without getter, setter methods. This method also brings security that prevents any change inside the tokens because there is no setter method defined.

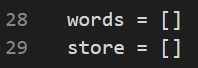
*Figure 2 demonstrates the Tokens class.*

**Class Node**

Node class is defined to give a value for the tokens.

*Figure 3 demonstrates the Node class.*

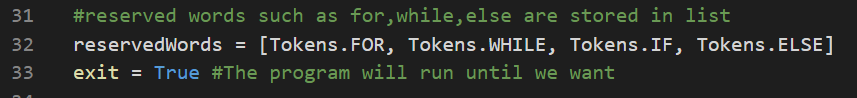
**Empty Lists**

This empty lists “words” and “store” are created to give an aid to our system. The “words” list is for reading the data from the text file, and the “store” list is used to append the values to the store them in the symbol table and to calculate how many values that are identified.

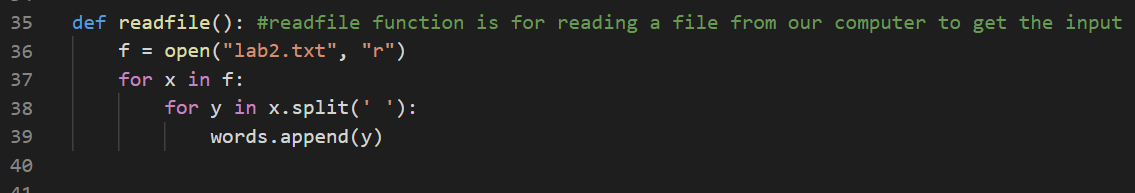
*Figure 4 demonstrates the empty lists for “words” and “store”.*

**Reserved Words and Exit**

We have created another list for the reserved words which are “for”, “while”, “if”, “else” to store them separately from the other types of data. Moreover, the “exit” variable is initially assigned to “True” to make our menu-based program to run until the user wants it to be terminated.

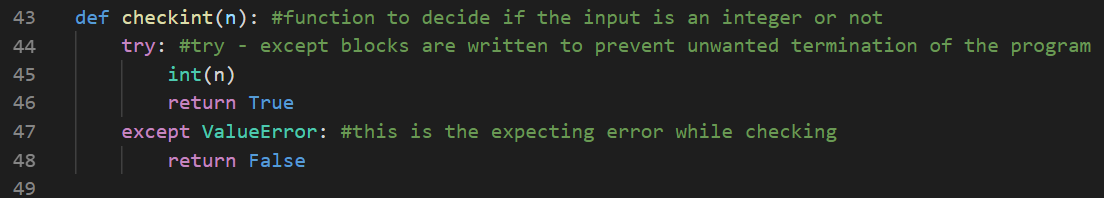
* Figure 5 demonstrates the reserved words’ list and initialization of exit variable.*

**Reading File**

We are getting the input as reading data from a text file with file extension of “.txt”. To achieve this, we created a function which reads file with the file operations. We opened the file in “read-only” mode to not to be able to make any changes by mistake, and we are appending every single data to the “words” list which declared above.

*Figure 6 demonstrates the function to read file from the system and appending them to empty list.*

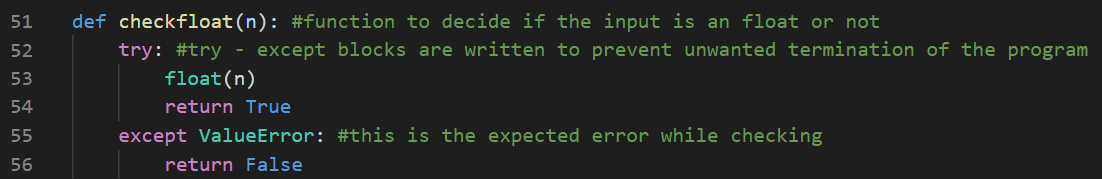
**Checking Integer Type**

In this part of code, we started to check the data types of the data that we read from the “.txt” file. In this section, try-except blocks are written to prevent the termination of the program when the time that the data is not an integer type.

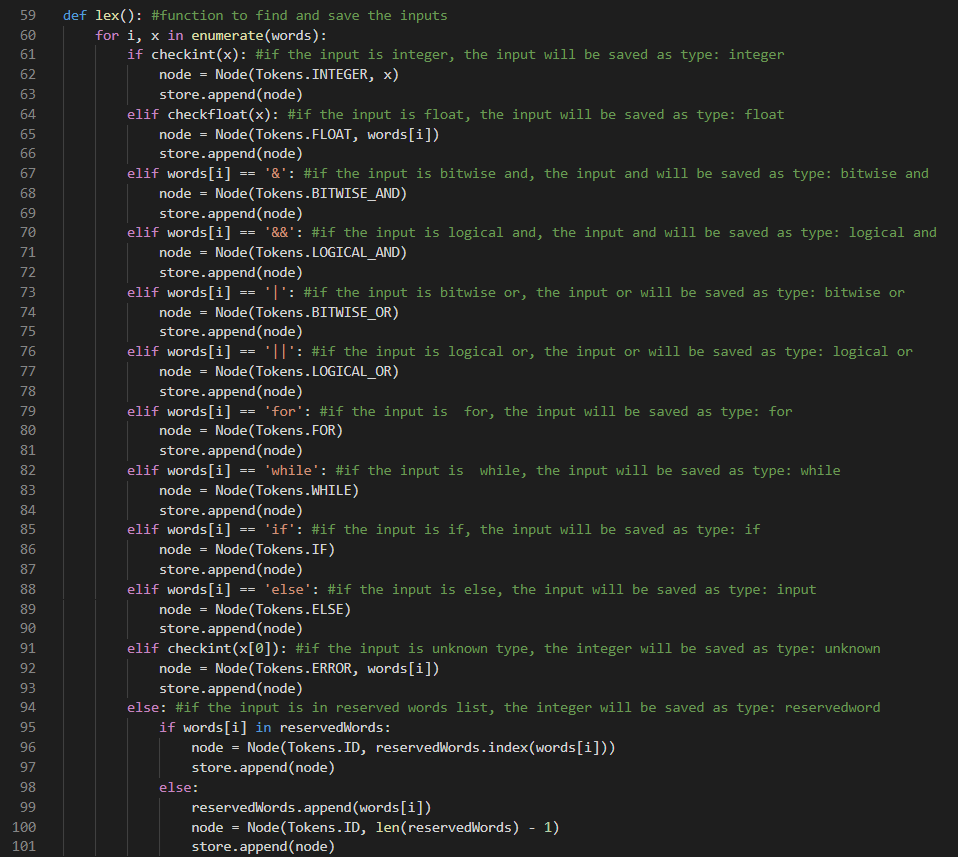
*Figure 7 demonstrates the function which checks if the data is integer or not.*

**Checking Floating Type**

Just like the checkint function, we are checking our if the data is a floating-point number or not via function called checkfloat. Try-except blocks are written to prevent the termination of the program when the time that the data is not a floating-point type.

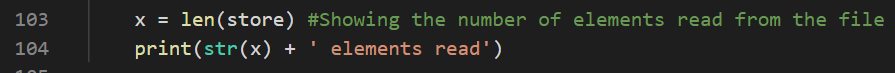
* Figure 8 demonstrates the function which checks if the data is floating-point or not.*

**Lex Function**

This is the part that where the main operations are performing. The lex function is deciding the data types of the data that are read from the text file. It is working with a simple if-else block to check the data type with “type()” method which is provided by enum library. After the decision, the data is appended to the empty set “store” which initialized above.

*Figure 9 demonstrates the lex function.*

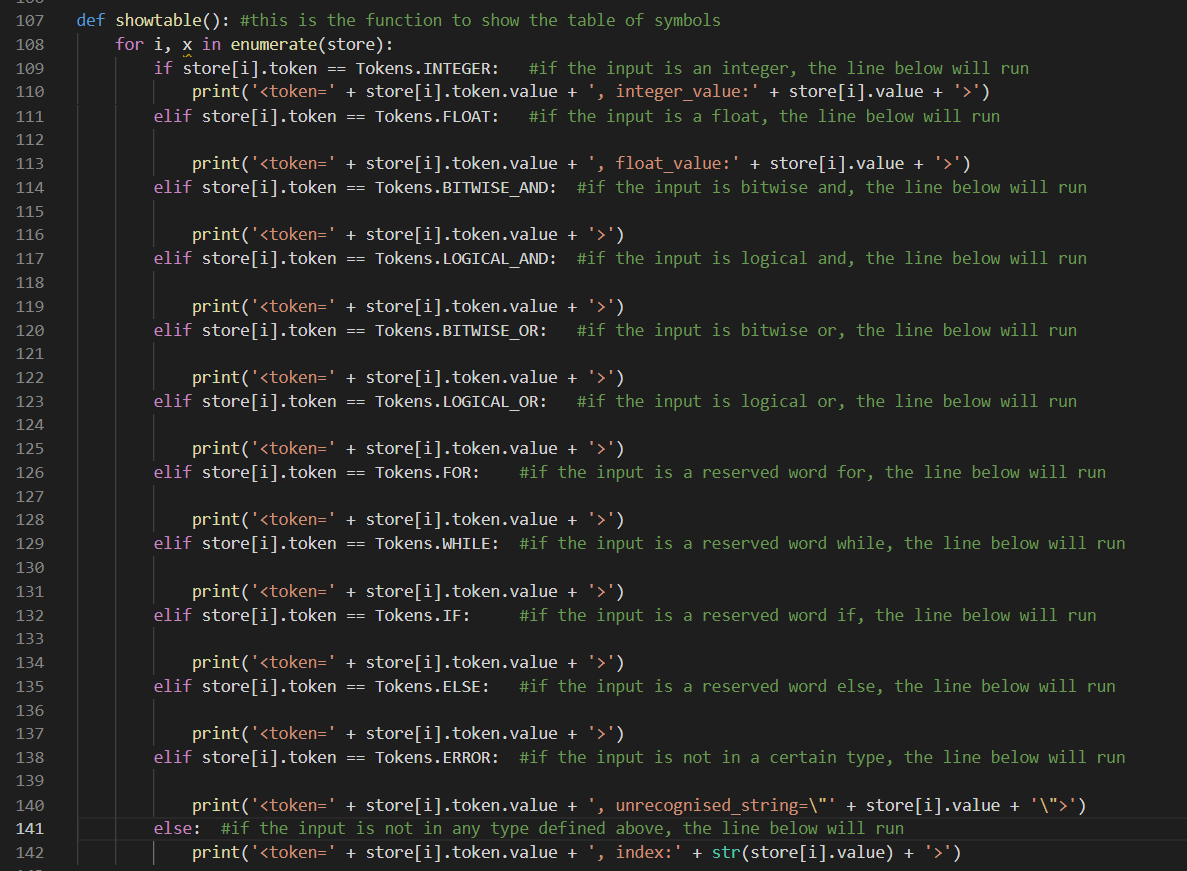
**Calculating Storage**

After the analysis is done by the lex function, we are calculating the number of data which went through the decision process with easily using the method “len(store)” and initializing it to a variable called x to print it to the users.

*Figure 10 demonstrates the calculation of the storage.*

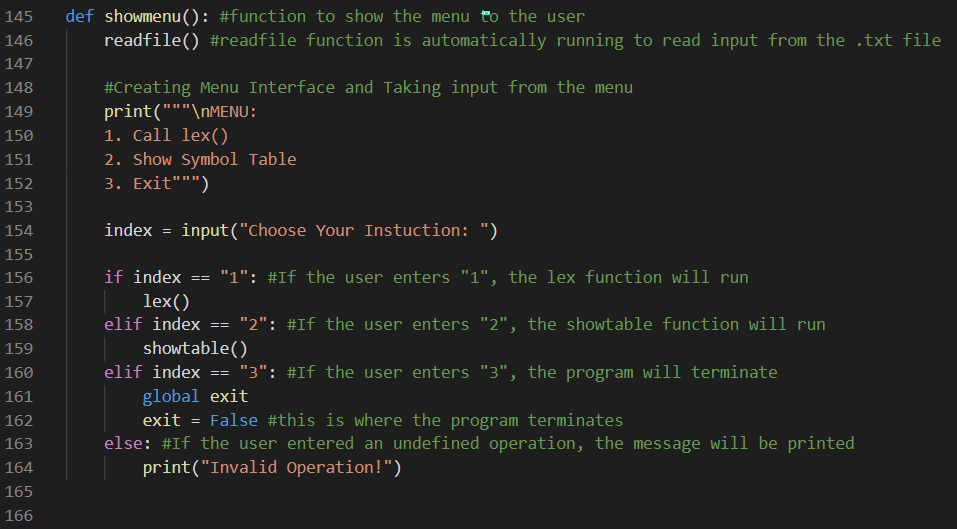
**Show Table**

This is the part of the system where the symbol table is shown to the user when it is selected from the menu. This part’s work process is as follows: The function is reading the data from the “store” list which determined and filled from the lex function, and then “showtable” function is reading the values from the “store” list and printing them to the screen with the view that is asked for in the assignment file.

 *Figure 11 demonstrates the showtable function.*

**Showing Menu & Input Operation**

We simply created a function called “showmenu” with the way that we are asked for. This function is printing the string “MENU:” to the users, on the following lines, the functions are written with the relevant operation number. Afterwards, we are taking the input from the user with the string “Choose Your Instruction: “, and assigning it to the variable called “index”. The “index” variable is going to an if-else block to determine the instruction entered by the user. In a condition of invalid operation input, the system is not terminating and informing the user that the value entered is wrong.

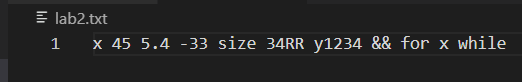
* Figure 12 demonstrates the menu for our program.*

**Running the Program**

This part is just a while loop that took the conditional variable “exit” that we defined very above until the instruction exit is selected. When exit is selected, the value of exit is going to turn to False and the program will terminate.

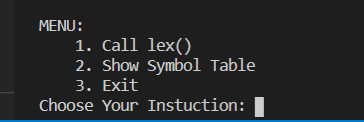
*Figure 13 demonstrates the exiting from the program.*

**4. TESTING**

In this part, we are going to test our program. We are going to share the output for each function and in the end, we will summarize it. Before the testing stage, we wanted to show you the prepared text file called “lab2.txt”.

*Figure 14 demonstrates the .txt file.*

**Menu**

The figures below will show what the menu looks like when the program started to run.

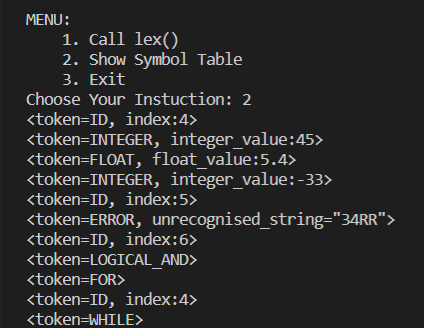
*Figure 15 demonstrates the menu-test.*

**Lex Function**

We have already defined what lex function does. When we chose our instruction as “1” from the menu, the lex function automatically runs. After the run is completed, we just see the number of data analyzed.

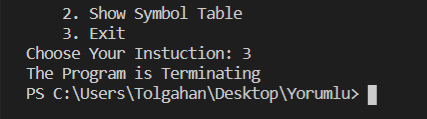
*Figure 16 demonstrates the output of lex function.*

**Show Table**

After the lex function run, now we can use show table function. The output of the show table operation is as follows.

*Figure 17 demonstrates the output of show table function.*

**Exit**

This the last part of the code where the program is terminating by the purpose of the user.

*Figure 18 demonstrates the termination process of the program.*

**5. CONCLUSION**

Today, we created a menu-based console application written in Python to lexically analyze the data that will be driven from a text file. We coded and tested our program. This assignment helped us to understand the basic principles of data analyzes.

**6.REFERENCES**

1. *Definition of enum: https://www.tutorialspoint.com/enum-in-python*