

**CS 424 – Compiler Construction**

Assignment # 2 Report

Abdul Ghani Khan 2020005

## MiniLang Scanner Report

**1. Design**

This report documents the design and implementation details of a scanner for a simple language called MiniLang, built using PLY (Python Lex-Yacc). The scanner identifies and classifies tokens from the MiniLang source code.

**2. Tokens**

The scanner recognizes the following tokens:

* **Keywords:** if, else, print
* **Operators:** +, -, \*
* **Symbols:** (, )
* **Identifiers:** Variable names starting with a letter or underscore, followed by letters, numbers, or underscores.
* **Numbers:** Sequences of digits.

**3. Implementation Details**

The scanner is implemented using the lex module from PLY. Here's a breakdown of the key functions:

* **tokens**: Defines the list of recognized tokens.
* **t\_PLUS, t\_MINUS, etc.**: Regular expressions used to match operators and symbols.
* **t\_VARIABLE**: Matches variable names and converts them to strings.
* **t\_NUMBER**: Matches numbers and converts them to integers.
* **t\_ignore**: Specifies characters to be ignored (whitespace characters).
* **t\_error**: Handles errors by printing an error message and skipping the illegal character.

**4. Test Cases**

The following test cases demonstrate the scanner's functionality:

* **Valid tokens:**
  + x = 5 (identifies x as a variable, = as assignment, and 5 as a number)
  + a + b \* 2 (identifies a, +, b, \*, and 2)
  + if (x > 0): print("Positive") (identifies keywords, operators, symbols, and variable)
* **Invalid tokens:**
  + $variable (variable name starting with a special character)
  + 123abc (number followed by letters)
    - (single operator without operands)

**5. Running the Scanner**

The provided code includes a loop that prompts the user for MiniLang statements. It attempts to parse the input using the scanner and prints the identified tokens if successful. The user can exit the loop by pressing Ctrl+D.

**6. Limitations**

This scanner is a basic implementation and has limitations:

* It only recognizes a small set of tokens.
* It does not handle comments or string literals.
* Error handling is basic and could be improved.

**7. Conclusion**

This report has documented the design, implementation, and testing of a scanner for MiniLang. The scanner provides a foundation for building a more complex parser that can understand and process MiniLang programs.