

HOMEWORK ASSIGNMENT #1

Instructor: Mr. Usama Arshad
Subject code: CS 424

Due Date: Feb 20, 2024
No Late submission allowed

[Total 20 marks]

PLEASE NOTE: THIS is an **INDIVIDUAL** assignment and **NOT** a group assignment.

Objectives:

This assignment will help students understand the complexities of lexical analysis and prepare them for the subsequent stages of compiler design. Encourage them to be creative with their test cases and to explore the process of tokenizing different elements of the MiniLang language.

Title: Design and Implement a Scanner for "MiniLang"

Scenario:

Imagine a small, new programming language called "MiniLang." MiniLang is designed to be simple yet powerful enough to demonstrate key programming concepts. It supports basic arithmetic operations, variable assignments, if-else conditions, and print statements. Your task is to design and implement a scanner for MiniLang using C++ or python.

Requirements:

1. Language Specifications:

- Data Types: Integer, Boolean.
- Operators: + (addition), - (subtraction), * (multiplication), / (division), = (assignment), == (equality), != (inequality).
- Keywords: if, else, print, true, false.
- Identifiers: Variable names starting with a letter followed by any combination of letters and digits.
- Literals: Integer literals, Boolean literals (true, false).
- Comments: Single-line comments starting with //.

2. Scanner Implementation:

- Your scanner should read MiniLang source code from a file and tokenize it according to the language's specifications.
- Implement a finite state machine in C++ or Python that can recognize tokens defined in MiniLang's specifications.

- The output should be a list of tokens, each token should include the token type and the lexeme.

3. Error Handling:

- The scanner must be able to recognize and report lexical errors, such as invalid symbols or malformed identifiers.

4. Documentation:

- Document your design decisions, the structure of your scanner, and how to run your program.
- Include test cases that demonstrate your scanner's capabilities, including edge cases.

Deliverables:

- Source code for your scanner. (10)
- A report documenting your scanner's design, implementation details, and test cases. (Submit report of 2 pages only as hard copy in class.) (5)
- A set of MiniLang example programs and their corresponding output tokens as screenshots. (3)
- (Send as a zip file on email – usama.arshad@giki.edu.pk)
- Upload on GitHub and share link in email. (2 marks)

Evaluation Criteria:

- Correctness and completeness of the scanner implementation.
- Ability to accurately tokenize the input and handle errors.
- Code quality, including readability, comments, and adherence to C++ or Python best practices.
- Quality and thoroughness of the documentation and test cases.

===== *to err is human* =====