CS 410 **Compiler Construction** Phase 1: Scanner

EXPECTATIONS

As a team, either **select a programming language** or **design your own** in consultation with the instructor. Then…

A **Describe a minimal dialect** for your programming language using an outline in natural language (not using a BNF or EBNF grammar, which is reserved for a future assignment). Your dialect should:
- Include a control flow construct for definite iteration, like a for loop.
- Include a control flow construct for indefinite iteration, like a while loop.
- Include a control flow construct for conditional branching, like an if/else/else-if.
- Include variable declarations and assignment statements.
- Include data types for integer variables and floating point variables.
- Include arithmetic operators for addition, subtraction, multiplication, and division.
- Include numeric comparison operators for equal, unequal, lesser, lesser or equal, greater, and greater or equal.
- Support conventional precedence and associativity for arithmetic operators and grouping symbols like parentheses.
- Exclude functions, methods, and other subprograms.
- Exclude data types for structures like arrays and strings.
- Exclude boilerplate which every program in your dialect would share verbatim.

B **Illustrate a finite-state machine** for lexical analysis of your dialect in step A. Your finite-state machine should:
- Accept the tokens of your dialect, but reject any other tokens.
- Follow conventional notation for states and transitions.
- Distinguish accepting states, non-accepting states, and the starting state.
- Include transition actions only which are vital for lexical analysis.

C **Program a scanner** in Java based on your finite state machine in step B. Your scanner should:
- Define a state-transition table in the form of a two-dimensional array.
- Input a string or character sequence from a user, file, or standard channel.
- Output a stream of tokens to a user, file, or standard channel. Your stream of tokens should:
  - Represent each token's class, like the abstract concept of a keyword, operator, identifier, or literal.
  - Represent each token's value, if any, like the concrete name of an identifier or numeric value of a literal.
- Be hand-written in Java, not be made by a compiler generator (which is reserved for a future assignment).

ASSESSMENT

As a team, submit your compiler phase deliverables to the assignment submission page in eCampus.

- **Review the rubric** to ensure that your deliverables meet the grading criteria.
- **Credit the authors and reviewers** of each deliverable within the corresponding file.
  - Each deliverable has 2 or more authors and 1 or more other reviewers.
  - Each member is accountable as the author of 2 or more deliverables and as the reviewer of 1 or more other deliverables.
  - All members should contribute equitably as both authors and reviewers of their assigned deliverables.
  - If any members don't contribute equitably, please notify the instructor before or after submission.
- **Attach your minimal dialect** as a PDF file.
- **Attach your finite-state machine** as a PDF file.
- **Attach your scanner** as one or more source code files.
- **Verify your submission** with the instructor before the compiler phase is due.