Mini Project Guidelines

Implement only the front end of the compiler

Can hardcode OR use tools(lex and yacc, PLY, JFLEX etc)

Lexical Analysis

1. Remove Comments
2. Generate tokens
3. Preload keywords into the symbol table
4. Make an entry for the identifiers into the symbol table(if there exists an identifier with the same name in different scopes then construct symbol table per scope)
5. Symbol table must contain entries for predefined routines like printf, scanf etc

Syntax Analysis

1. Write CFG for the entire program.
2. If implementing parser by hand:
    - Prefer RDP with backtracking.
    - Perform translation at required places in the code for each non-terminal.
3. If implementing using tool:
    - Provide appropriate semantic rules.

Semantic Analysis

1. Take care of the primitive types and array types.
2. Take care of coersions.
3. Take care of Arithemetic expressions.
4. Concetrate on the looping construct choosen.
5. Update type and storage information into the symbol table.
6. Show Abstract Syntax tree(AST).

Intermediate code generation

1. Three address code generation

Optimized Intermediate code

Optimization that are to be performed

- Constant folding
- Constant Propogation
- Common subexpression elimination (optional)
- Dead code elimination
- Reducing temporaries (optional)

- Loop optimizations

Deliverables by 11th week of the lab

1) AST (Abstarct Syntax tree)
2) Intermediate code
3) Optimized Intermediate Code