



Delhi Technical Campus Greater Noida

Group Assignment/Mini Project

Instructions:

- This mini project is designed to cover various compiler phases from lexical analyzer to code generation. The project aims to cover all the COs.
- Students should work collaboratively in groups, allocate tasks for implementation and present the solutions in a final report in lab file.
- Submit the assignment on or before (Mentioned date).

Subject: Compiler Design

Subject Code: CIC-303/CIC-351

Class: B.Tech CSE 5th

**Faculty Name: Dr.Seema Verma/
Sweta Rai**

Date of Issue:

Date of Submission: 31 OCT 2023

Mini Project Title: "Mini Compiler: High-Level Language Conversion"

Project Overview:

The project aims to design a compiler software capable of translating code from one high-level programming language to another. It will encompass the following core phases:

1. Lexical Analysis:

- Develop a lexer to tokenize the input source code.
- Define rules for recognizing keywords, identifiers, constants, and operators.
- Implement basic error handling for invalid tokens.

2. Syntactic Analysis (Parsing):

- Create a parser to generate a simplified abstract syntax tree (AST) from tokens.
- Handle syntax errors and provide basic error messages.

3. Semantic Analysis:

- Perform basic checks on the AST for type compatibility and variable scope.
- Include a minimal symbol table to store variable information.
- Report basic semantic errors.

4. Code Optimization (Optional)

- Basic optimization constructs to be added

5. Code Generation:

- Translate the AST into the target high-level language.
- Generate code adhering to the target language's syntax and basic semantics.
- Ignore platform-specific considerations.

Conversion Between High-Level Languages:

- Define mappings between source and target languages for basic constructs like data types, control structures, and functions.

Development Tools:

- Choose any programming language (C/C++/others) and development environment for implementation. You can use available parser/lexer generators or build them manually.

Testing and Validation:

- Create a limited set of test cases with source code in the source language and expected output in the target language. Test the compiler to ensure basic correctness.

Documentation and User Interface:

- Provide minimal documentation explaining how to use the compiler and how language conversion works. Create a basic user interface for input and output.

Remarks: This simplified project is to be focused on the essential components of a compiler while omitting some of the complexity found in real-world compilers. It should be more manageable for smaller teams or educational purposes.