

CSL302: Compiler Design

Assignment-3

Due date: November 21, 2025 11:59 PM

Instructions

- Implement your solution using bison and flex tools.
- Prepare your submission in a zip file and name it as <ROLL_NO1_ROLL_NO2>.zip. If the team size is other than two, name your submission accordingly.
- Your submission should include a README file, containing the instructions to execute your program(s). README file carries 5 Marks.
- Upload your assignment in the canvas portal.
- Note that weightage of each assignment will be different.
- It is sufficient if one member of the team submits the file.
- **Any form of plagiarism is strictly prohibited; all the text must be your own, and all the references must be cited.**

1 Intermediate Code Generation

In this assignment, you are required to implement the intermediate code representation (TAC) for the same language as used in syntax analyzer. You can reuse the lexical and syntax analyzer developed in Assignment-1 and 2 for solving this assignment.

Your implementation should take an input program and produce an output containing the three address code. If the input program is invalid, your solution should print the associated error(s) with a meaningful description(s).

Your implementation should support the following semantics.

Semantics:

1. All the identifiers must be declared before they are accessed. A variable should not be declared twice.
2. Access to variables in the nested block structures NOT REQUIRED.
3. You must implement a symbol table to store the information associated with the identifiers along with the scope. Refer to the Lecture on symbol tables.
4. In if statements the else is associated with the most recent if.

5. Logical and/or operators do not follow short circuit evaluation
6. Functions other than main (or any equivalent function, depending on the programming language) are **NOT** supported.
7. Operators follow the same precedence and associativity as that of the C language.
8. You should generate the three-address code for all the program statements within the main function.
9. Arrays declaration and reference should be supported.

Note that some features are yet to be covered in class, but this should not stop you from solving the assignment.

[**Points 100**]