

## Simple Programming Language using Lex and Yacc

### Number of Students per Group

3-4 Students

### Overview

It is required to design and implement a programming language using the Lex and Yacc compiler generating packages.

### Requirements

- Design a suitable programming language; you may use an existing one. The important constructs to be considered are:
  - Variables and Constants declaration. ✓
  - Mathematical and logical expressions. ✓
  - Assignment statements. ✓
  - If-then-else statement, while loops, repeat-until loops, for loops, switch statement. ✓
  - Block structure (nested scopes where variables may be declared at the beginning of blocks). ✓
  - Functions. ✓
- Design a suitable and extensible format for the symbol table. ✓
- Implement the lexical analyzer using Lex. ✓
- Design suitable action rules to produce the output quadruples and implement your parser using YACC. ✓
- Implement a proper syntax error handler. ✓
- Build a simple semantic analyzer that checks for the following:
  - Variable declaration conflicts. i.e. multiple declarations of the same variable. ✓
  - Improper usage of variables regarding their type. ✓
  - Variables used before being initialised and unused variables. ✓
  - [Optional] The addition of type conversion quadruples to match operators' semantic requirements, i.e. converting integer to real, etc. ✓
- [Optional] Implement a simple GUI. ✓

### Project Phases

- **Phase I:** In this phase, you're required to deliver your lex and yacc files i.e, your lexer and parser.

- **Phase II:** In this phase, you're required to modify your implementations to include the following:
  - Design a suitable and extensible format for the symbol table. ✓
  - Design suitable action rules to produce the output quadruples. ✓
  - Implement a proper syntax error handler. ✓
  - Build a simple semantic analyzer. ✓

## Deliverables

- Source code of your project.
- A Document that contains the following:

- Project Overview
- Tools and Technologies used
- A list of tokens and a description of each
- A list of the quadruples and a short description of each e.g:

Quadruple	Description
JMP L	Unconditional jump to label l
NEG V1, V2	V2 = -V1

## Program evaluation

- The program is to be fed by a source code file containing your language and do the following:
  - Produce the corresponding quadruples.
  - Display syntax errors that exist in your program.
  - Display the semantic errors that exist in your program.
  - Display the symbol table.

## Evaluation Criteria

- The correctness of your quadruples.
- Handling syntax and semantic errors.
- Teamwork and good documentation.

## Notes

- Anything listed as optional will be considered **a bonus**.
- Any semantic checks implemented other than the ones mentioned above will be considered **a bonus**.

## Due Dates

- Phase I delivery: week 11 | 20 April 2024 (11:59 PM)
- Phase II delivery: week 14 | 11 May 2024 (11:59 PM)