

# CS335A

## Project

### Milestone 3

Ashok Vishwakarma(180151)  
Viraj Ravjibhai Limbasiya (180870)

April 4, 2023

## Tools used

We implemented the 3AC using an in-memory data structure (List) in python. The software/utilities used are:

- **Ply (Python Lex-Yacc):** It is a Python implementation of the popular Lex and Yacc tools for parsing and analyzing complex languages. It provides a set of tools for generating lexer and parser code in Python from a set of grammar rules specified by the user.
- **ply.lex** is a lexer generator that takes in a set of regular expression patterns and generates a lexer in Python code.
- **ply.yacc** is a parser generator that takes in a set of grammar rules and generates a parser in Python code.

## Build/Execute

The Lexical grammar for Java8 is present in JavaLexer.py which has been obtained from [Java 17 Lexical Structure](#).

The Syntactic actions for grammar are present in IRGen.py which has been obtained from [Java 17 Syntax](#).

To run the source code, use the following command:

```
> make
```

which internally calls 'python3 ./IRGen.py ../tests/test\_1.java'

It will generate the 3AC in the txt file (.3ac extension) of the input Java file.