

# Complier Course Project

by

**Group: 10**

**Group Name: Syntax Terror**

Member Name	Roll no.	Email ID
Aakash Om Trivedi	200003	aakasht20@iitk.ac.in
Krishan Kumar	200521	krishank20@iitk.ac.in
Ujjwal Kumar	201059	ujjwalk20@iitk.ac.in

Course:	CS335
Instructor:	<i>Swarnendu Biswas</i>
TA:	<i>Abhishek Revaskar</i>
Date:	19-04-2023

## Basic Features supported by our implementation are:

- **Primitive Data Types -**
  - int, long, float, double, and boolean
  - one-dimensional and two-dimensional arrays of int
- **Basic Operators**
  - Arithmetic operators: +, -, \*, /, %, ++, --
  - Preincrement, predecrement, postincrement, and postdecrement
  - Relational operators: ==, !=, >, <, >=, <=
  - Bitwise operators: &, |, ^, ~, <<, >>, >>
  - Logical operators: &&, ||, !
  - Assignment operators: =, +=, -=, \*=, /=, &=
  - Ternary operator
- **Control flow**
  - Loops:
    - For
    - While
  - Selection Statements
    - If
    - if-else
- **Arrays:**
  - 1-D and 2-D arrays
  - Format allowed for initialization, int a[] = {1,2,3,4} and int a[][] = {{1,2},{3,4}}
- **Methods:**
  -
- **Classes**
  - Public access modifier is implemented
  - Both inner and outer class is allowed.
- **Functions**
  - Function Invocation is implemented

## Basic Features not supported by our implementation:

- private access modifier is not implemented

We were able to include most of what we have done till the final milestone. Above all, the final code generates the 3ac structured intermediate code. We faced some challenges in xv86 stack manipulation but we tried our best to overcome those.

#### Compilation commands

- `chmod +x ASTGenerator`
- `./ASTGenerator -input <input_file_name> -output <output_file_name>`

## **Contribution Table**

<b>Sr. No.</b>	<b>Member Name</b>	<b>Roll no.</b>	<b>Email ID</b>	<b>Contribution(%)</b>
1.	Aakash Om Trivedi	200003	aakasht20@iitk.ac.in	33.33
2.	Krishan Kumar	200521	krishank20@iitk.ac.in	33.33
3.	Ujjwal Kumar	201059	ujjwalk20@iitk.ac.in	33.33
Total				100