

CS335 Project Milestone 1 Report

Group Members: Divyansh Kankariya (200352), Nishi Mehta (200645), Mohd. Umam (200595)

March 23, 2023

1 Tools Used

In this project, we used the following tools:

1.1 Flex

Flex is a tool for generating scanners (also known as lexers or tokenizers) that can recognize regular expressions in the text. We used Flex to generate a lexer for the Java 17 language.

1.2 Bison

Bison is a tool for generating parsers that can recognize context-free grammar. We used Bison to generate a parser for the Java 17 language.

2 Compilation and Execution Instructions

To compile and execute our lexer and parser, follow the steps below:

1. Unzip the file `cs335_project_200645_200595_200352`
2. Navigate to the project directory
3. Compile the lexer and parser using the makefile provided.

```
1 // Your code goes here
2 make
3 make st
```

On executing the 'make st' command, we will have to enter the filename of the test file. After the execution of the parser, all the symbol tables are stored in the folder 'outputSymTabs' with the filename as 'Scope.SCN.csv' where SCN is the scope number. In our representation, the scope number is represented as, eg:- '1.2a.3a', for three scopes.

3 Language Features

Our symbol-table plus type-checking for Java-17 supports the following language features: **milestone 2**

- Support for type-checking for all constructs and creating symbol tables according to their scopes.
- For arrays, we have represented the array dimensions by the type and number of brackets. For eg, for a 2-D array of 'int', we have `int[][]` as the dimension.

- For class objects, we have stored their type as the class name itself. For eg, the object of the class name says, 'Class1' has its type stored as 'Class1'.
- We have implemented support for public, private and static modifiers over all three types of declarations ie. classDeclaration, fieldDeclaration and methodDeclaration.
- In the case of method declaration, we have stored its type as 'method', and as 'constructor' for constructor declaration.
- We have also implemented support for widening of data-type for a variable, for eg, say $a = 5 + 9.8$, we have the type of variable 'a' as a float.
- For classes, we have stored them with the type 'class' at the time of their declaration.

The things we have assumed:

- Function definition occurs at the same place as its declaration
- If-else statements don't have single-line declarations without curly-braces
- Assumed that no-recursion calls occur
- Library functions such as 'System.out.println()' are directly added to the global symbol table.
-

Note that this list is not exhaustive and may be updated as we continue to develop our implementation. We were able to figure out the functions required for the 3-AC generator but were not able to finish it off.

In addition, we have created 14 test cases to ensure the correctness of our lexer plus parser. These test cases are located in the `tests` folder.