# CS335 Milestone 1- Creating AST for java program

Tejas Ramakrishnan-201050, Ujwal Jyot Panda-201060, Uttam Kumar-201071

January 2023

To construct a scanner and a parser for the Java language. The output of your compiler should be the abstract syntax tree (AST) in a graphical form.
We have used flex and bison for lexical scanning and parsing respectively. We have also used C++ to implement and create the AST .dot file. **Compilation instructions-** To compile, first navigate to the directory 'src' in the terminal. The files in this directory are: scanner.l, parser.y, tree.h and tree.cpp. To compile the program, run the following command in the terminal:

```
make
```

You may get an warning stating 'clang: warning: treating 'c' input as 'c++' when in C++ mode, this behavior is deprecated', please ignore that.
**Execution instructions-** An executable 'ans' would have been created in the directory 'milestone1'. To execute the program with a test case file 'test1.java' in the "tests" directory, run the following commands from the src directory:

```
./ans -input ../tests/test1.java -output graph.dot
```

The output will be created in a dot file called 'graph.dot' in the same directory, in the required format. Please open to check the output. You can run

```
dot -Tpng graph.dot -o graph.png
```

to create an image for it. Options supported in the parser are:

```
--help: For usage guidelines
-input <file>: Passes the input java file to the parser to be read
-output <file>: Creates the dot script containing the AST in <file>
-verbose: Prints entire line of error
```