

# CS335 Project Milestone 4

Rishabh Kothary	Rikesh Sharma	Kartavya
180608	180606	180343
rishk@iitk.ac.in	rikesh@iitk.ac.in	kartavya@iitk.ac.in

## 1 Tools Used

We did lexical analysis using *flex* and parsing using *bison*. To automate compiling we used *Makefile* and to make graphs we used *graphviz*. Our codebase is written in C++ programming language.

## 2 Flags Available

The flags available are :

- `--input=` : Used to specify path to input file
- `--output=` : Used to specify path to output file
- `--verbose` : Used to switch on the debugger of bison
- `--help` : Used to load the help page

## 3 Compilation

To compile the compiler on the Milestone\_4 directory run the following command on the command line.

```
make parse
```

If you want to clean all the binaries and outputs then run the following command from the Milestone\_4 directory on the command line:

```
make clean
```

If you want to generate the assembly of our test suites in an automated manner run the following command from the Milestone\_4 directory on the command line:

```
make test
```

then you would be asked for the test suite number. Is suppose you picked test suit x, then the files test\_x.s is created in the output folder. In order to create assemblies of multiple test suites at the same time run the following command on the command line from the Milestone\_4 directory :

```
make test_many
```

and then you will be asked number  $n$ , which will run *test<sub>1</sub>.java* to *test<sub>n</sub>.java* and return the output in the output folder.

If you want to generate the assembly of our test suites and even run them, then run the following command from the Milestone\_4 directory on the command line:

```
make run
```

then you would be asked for the test suite number. If suppose you picked test suit  $x$ , then the files test\_x.s is created in the output folder. The output and errors of the code will be present in test\_x.txt file. In order to create assemblies of multiple test suites at the same time and even run them, then run the following command on the command line from the Milestone\_4 directory :

```
make run_many
```

and then you will be asked number  $n$ , which will run *test<sub>1</sub>.java* to *test<sub>n</sub>.java* and return the output in the output folder.

If you want to run your own test cases then use the following commands from the Milestone\_4 folder on the command line :

```
cd bin
./parser --input = < Path to Input File > --output = < Path to output file >
```

This would generate the assembly output file.

## 4 Features we support

We only support static function calls. You can't create objects and invoke their functions. We assume the output of any static method is of integral type. Apart from that we support all the necessary requirements of the project.

## 5 Effort Sheet

Name	Roll No.	Email Id	Effort
Rishabh Kothary	180608	rishk@iitk.ac.in	46%
Rikesh Sharma	180606	rikesh@iitk.ac.in	46%
Kartavya	180343	kartavya@iitk.ac.in	8%

Table 1: Effort Sheet