Attack On Compilers

Python based implementation of Java Compiler

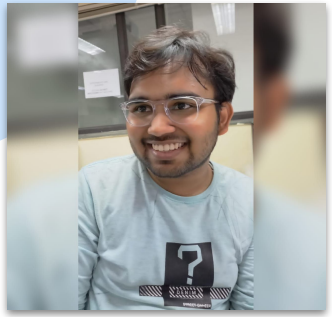
CS335A | Group 17 | Spring'23

ABOUT THE PROJECT

Course Project on implementing a compiler to compile **Java** to **x86** target.

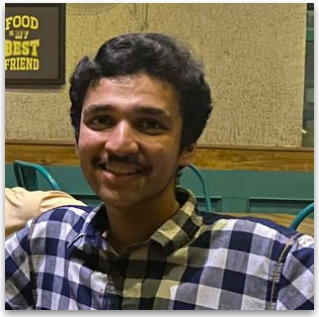
Milestones: Lexer, Parser, Symbol Table, Type checking, Three address code IR, Memory management, x86 target output

Team and Course Admins



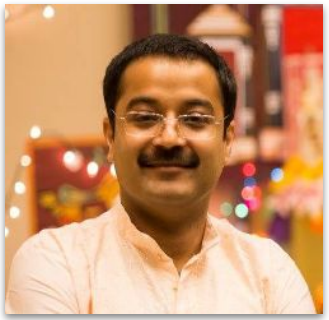
Harshit Raj

Kumar Arpit

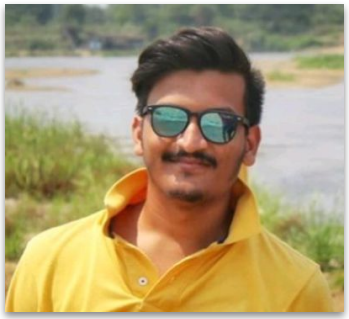


Akshan Agrawal

Team Members



Prof. Swarnendu Biswas



Mr. Dinkar Tewari

Acknowledgement

We express our gratitude towards Prof. Biswas for smooth conduction of the course and valuable guidance, and towards Mr. Dinkar for his valuable feedback throughout the project.

TABLE OF CONTENTS

DEPENDENCIES

- ply (python-lex-yacc)
- Graphviz
- Argparse
- gcc 11.3.0 +

BASIC FEATURES

- Primitive Data Types
- Multidimensional Arrays
- All basic arithmetic, logical, relational and conditional operators
- Control Flows (if else, while loops, etc.)
- Methods and Method Calls
- Support for recursion
- Support for println()
- Support for classes and objects

BONUS FEATURES

- **Invoking functions on objects**
- Do while loop statements support
- Break and Continue
- Support for printing strings
- Support for both System.out.print() and System.out.println()
- **File I/O operations (fopen, fclose, fprintf)**
- Multidimensional Arrays (any # of dimensions, instead of just 3)
- Some support for booleans
- Constructor calls

TREE

```
├── doc
│   └── Report.pdf
├── Makefile
├── README.md
├── requirements.txt
├── src
│   ├── codegen.py
│   ├── dot.py
│   ├── lexer.py
│   ├── main.py
│   ├── parse.py
│   ├── symbol_table.py
│   ├── symbol_tac.py
│   ├── tac.py
│   └── utils.py
```

DEMO

```
pip install -r requirements.txt
python3 src/main.py --input /path/to/file.java [--output
 ↪  /path/to/output] [--all] [--verbose]
dot -Tps src/output -o src/output.ps
xdg-open src/output.ps
```

Additional Remarks:

Verbose will pretty print the symbol table, TAC and x86 code on the terminal.

The last two command lines are for the the graph.

Alternatively, you can also use the Makefile!

Let's move to code

EFFORT SHEET

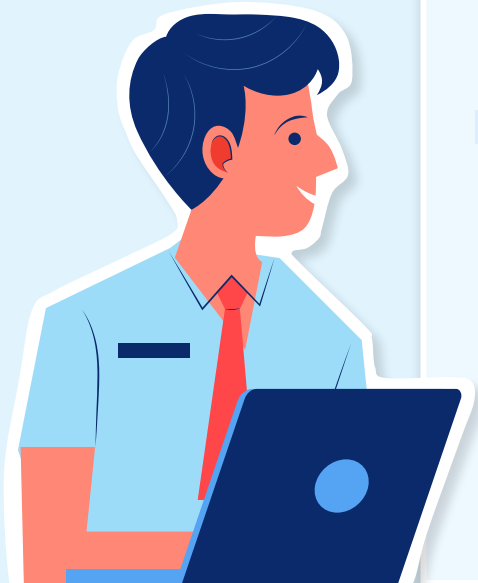
Name	Roll No	Email	Contribution
Harshit Raj	200433	harshitr20@iitk.ac.in	100%
Kumar Arpit	200532	arpit20@iitk.ac.in	100%
Akshan Agrawal	180061	akshan@iitk.ac.in	100%

FOOTNOTES

- Collaborated over git and GitHub
  https://github.com/attack-on-compilers/studious-java | Over 380 commits
- Report Available at
  https://github.com/attack-on-compilers/studious-java/blob/main/doc/Report.pdf
- Now we know where the uninitialized garbage value come from :)

REFERENCES

- Lecture Slides
- ply Lexer Manual
- Yacc manual
- Several Symbol table and type check implementations
- IR and TAC implementations
- x86 and gnu assembler tutorials
- Compiler Explorer: https://godbolt.org/



THANK YOU

Members, Group 17