CS335: Compiler Design
Project Milestone 3

Group 6

Group 6 members:
Shivang Pandey (200941)
Devesh Shukla (200322)
Yash Gupta (190997)

1 Git tag:

The git tag for this submission is:

`milestone3`

2 Tools used:

Flex, Bison, gcc/g++, C/C++

3 Compilation instructions

In the `milestone3/src/` directory, run the following command to generate the executable `main`:

```
make
```

4 Execution instructions

Let's say the input file is `input.py` and the output directory is `output_dir/`. The `-verbose` flag is optional. In the `milestone3/src/` directory, run `main` using the command:

```
./main -verbose -input input.py -output output_dir/
```

This will generate the following files in `output_dir/`:

- `x86.s`: x86_64 code

- CSV files: Symbol tables
- `3ac.log`: 3AC code
- `lexer.log` and `parser.log`: Debugging information

To run `x86.s` with gcc, run:

```
gcc output_dir/x86.s -o output_dir/test1 -no-pie
output_dir/test1
```

5 Command line options

Usage: `./main [-verbose] -input <input_file> -output <output_dir>`
-input: Input file to parse.
-output: Output directory to dump x86_64 code, symbol tables and 3AC.
-verbose: Show verbose output.

6 Features supported

- Primitive data types (e.g., int, float, str, and bool)
- 1D lists
- Basic operators:
  - Arithmetic operators: +,-, *, /, //, %, **
  - Relational operators: ==, !=, >, <, >=, <=
  - Logical operators: and, or, not
  - Bitwise operators: &, |, ˜, ˆ, «, »
  - Assignment operators: =, +=, -=, *=, /=, //=, %=, **=, &=, |=, ˆ=, «=, »=
- Control flow via if-elif-else, for, while, break, and continue
  - Support iterating over ranges specified using the `range()` function.
- Support for recursion
- Support the library function `print()` for only printing the primitive Python types, one at a time
- Support for classes and objects, including multilevel inheritance and constructors.
- Methods and method calls, non-static methods

7 Run test cases

In the `milestone3/src/` directory, run:

```
make
./run-compiler.sh
./run-x86.sh
```

This will run the 5 test cases in `milestone3/tests` and save the output in `milestone3/tests/test<number>/out.log`.

8 Changes in 3AC from milestone 2 to milestone 3

Further refined and improved the 3AC from milestone 2 to remove bugs like parameters included in global symbol table instead of local symbol table, and type error on passing a float to index lists. Other minor improvements were also made.

9 Manual changes to generated assembly file

No manual changes are required.

10 Effort Sheet

Devesh Shukla (200322) - 34%
Yash Gupta (190997) - 33%
Shivang Pandey (200941) - 33%