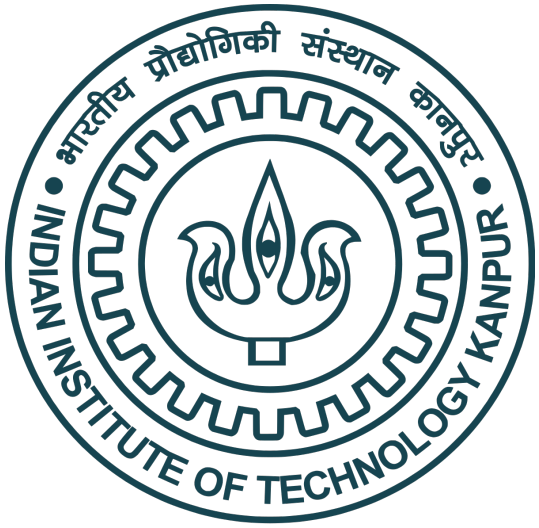

CS 335A

Project : Milestone 4

Samarth Arora (200849)
Sarthak Kohli (200886)
Abhimanyu Sethia (190023)

Instructor: **Swarnendu Biswas**

Date: April 19, 2023

1 Folder Content:

The submitted folder consists of the milestone4 directory which includes the following sub-directories and files:

- **milestone4/src** sub-directory that contains all the source files
- **milestone4/tests** sub-directory that contains all the test files
- **milestone4/readme.pdf** which contains this PDF.

2 Requirements and Tools Used:

We have the following tools to build a x86 assembly file given a java file as input.

- **Flex**: For Lexical Analysis
- **Bison**: For Syntactical Analysis
- To install the above requirements use the following commands:
  - sudo apt-get update
  - sudo apt-get install flex
  - sudo apt-get install bison

3 Compilation Instructions:

Follow the given steps to compile and execute the files:

- **Step1:** Open the terminal in the *milestone4* folder and go to the *src* directory. The following Command will get you there:

```
cd src
```

- **Step2:** To compile the files enter the following command:

```
make
```

- **Note** The compiled files will create the following extra files in the *src* folder. the **output** file is the executable. The files are:

```
lex.yy.c
parser.output
parser.tab.h
parser.tab.c
output
convert
```

To remove these files to clean the *src* repository enter the following:

```
make clean
```

4 Execution Instructions:

4.1 Individual file from the Command Line

- One can enter the input file as follows. This will create a corresponding a directory with the name of the java file in the location of the input folder. For example, the following command will create a a directory test_1 in ../tests/. This will contain all the symbol tables for all functions and all classes and one 3ac file called $\text{test}_1.3ac$.

```
./output --input=../tests/test_1.java
```

Use the verbose option to list all the shift, reduce and lexer actions. For example:

```
./output --input=../tests/test_1.java --verbose
```

Enter the following command to view about all the available options:

```
./output --help
```

If a file which does not have the extension ".java" is entered or a invalid option is entered then an appropriate error will be thrown.

Run the following command to make a x86 assembly file from .3ac file

```
./convert --input=../tests/test_1.3ac
```

Here similar to above you can mention the output file destination

Run the subsequent commands to execute this assembly file

```
gcc -c ../tests/test_1.s -o ../tests/test_1.o
gcc -o ../tests/test1 ../tests/test_1.o
./../tests/test1
```

5 Basic features of Our implementation:

We have provided support for the following basic features:

- Support for all the arithmetic operations, relational operations, bitwise operations
- Support for strings
- Support for if,for,while loops etc.
- Support for Methods
- Support for Constructor
- Support for println statement
- Support for array initialization and reference

6 Extension to Milestone 3 for Milestone 4:

We have changed our 3ac so that the conversion of 3ac to assembly file becomes easy. We made the following changes:

- We have used only temporary variables and base pointer and stack pointer and no variable names
- We have made it very similar to the x86 asssembly calling convention so that it is very easy for us to convert it into .s file

7 No manual changes to be done to the .s file. Just follow the execution given above

8 Contribution:

- Samarth Arora; Email id:samartha20@iitk.ac.in; Roll number:200849 ; **Contribution 55 percent**
- Sarthak Kohli Email id:sarthakk20@iitk.ac.in ;Roll number:200886; **Contribution 35 percent**
- Abhimanyu Sethia; Email id:sethia@iitk.ac.in Roll number:190023 ; **Contribution 10 percent**