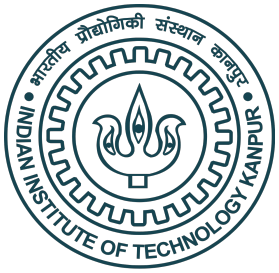

CS335 Compiler

MileStone 4

Swarnendu Biswas

April 22, 2023

Jaya Gupta (200471)

Harshit Bansal (200428)

Mohit Gupta (200597)

---

1 Requirements

- g++ $\geq$ 12, gcc $\geq$ 12 : The system should also have g++ installed. Its version should be greater than 12.
- flex: The environment should have flex installed.
  - If its Linux then it can be installed from the command

```
sudo apt install flex
```

- Bison : The environment should have bison installed.
  - If its Linux then it can be installed from the command

```
sudo apt install bison
```

2 Execution Instructions

Compilation and Execution Instructions

```
make clean
make
./build/milestone/java2x86 --input <input-file-name> --output asm.s
gcc -c asm.s -o asm.o
gcc -o asm asm.o
./asm
```

To parse all the testcases in tests folder, execute run.sh script.

Command Line Options

```
Usage: javair [-h] --input VAR --output VAR [--verbose]

Optional arguments:
  -h, --help    shows help message and exits
  -v, --version prints version information and exits
  --input       java file to parse [required]
  --output      output asm file name [required]
  --verbose     increase output verbosity for parser
```

3 Basic Features Implemented

- Primitive data types (e.g., int, long, short, byte and boolean)
- Multidimensional (max 3D) arrays.
- Basic operators:
  - Arithmetic operators: +, -, *, /, %, ++, –
  - Preincrement, predecrement, postincrement, and postdecrement
  - Relational operators: ==, $!=, >, <, >=, <=$
  - Bitwise operators: $\&, |, \hat{}, <<, >>, >>>$
  - Logical operators: $\&\&, ||, !$
  - Assignment operators: $=, +=, -=, *=, /=, \&=$
  - Ternary operator
- Control flow vila if-else, for, and while,
- Methods and method calls, including non-static methods
- Support for recursion
- Support the library function println() for only printing the primitive types listed earlier
- Support for classes and objects. For class definitions, public and private access modifiers are supported.

4 Optional Features Supported

- **this** keyword is supported. Any instance variable or function of the class can be referenced using `this` keyword.
- **Explicit Constructor Invocation** → A constructor within another constructor can be invoked using `this()`. It is a feature of Java.

- **Static polymorphism via method overloading** → Multiple functions with same name but with different arguments can be made and called in our Java compiler.
- **Multidimensional Array** → Arrays with more than three dimensions are also supported. Moreover Java-type array declarations are also supported.
- **Do_While** → Do While Loop Support is also given.
- **Unordered Function/Class Declaration** → As opposed to the C compiler where function should be declared before use, for our compiler in accordance with Java17, function/class declaration order does not matter. If a function is invoked above and declared below, the code works fine. Same is the case for classes.

```
public class GFG {
    static void main(){
       add(10,20);
    }

    static void add(int a, int b){
        int sum = a + b;
    }
}

// The above code works fine.
```

```
public class GFG {
    static void main(){
       DownClass down = new DownClass();
       add(10,20);
    }
}

public class DownClass{
    int x;
}

// The above code also works fine.
```

5 Assumptions

- Expressions in array dimensions is not allowed. Array dimension while initialising should be integer literal.
- Function arguments should only be integer type. Array arguments are not supported.
- Static variable and functions are not supported. For method invocation from Main(), first create a object and then use the object to call different methods and access instance variables.

6 Contribution

S.NO	MEMBER_NAME	ROLL_NO	EMAIL	CONTRIBUTION
1.	Jaya Gupra	200471	jayagupta20@iitk.ac.in	33.33%
2.	Harshit Bansal	200428	harshitb20@iitk.ac.in	33.33%
3.	Mohit Gupta	200597	mohtig20@iitk.ac.in	33.33%