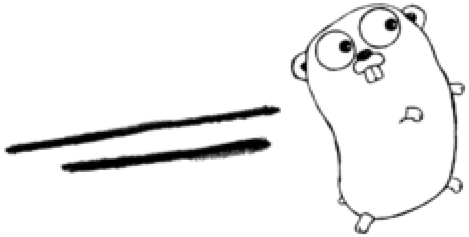
Compiler Construction

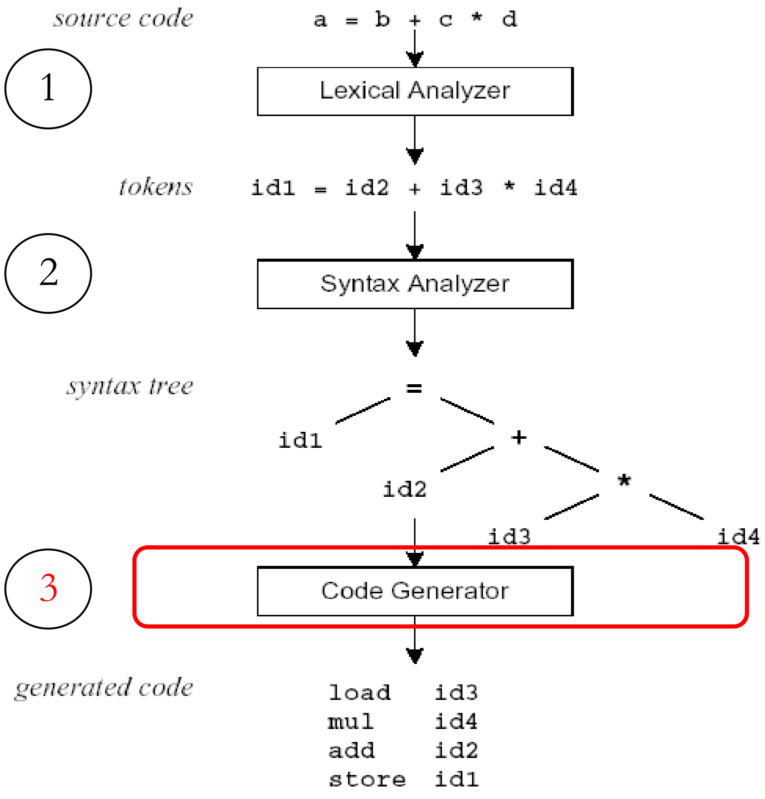
Programming Assignment 3

Generate Java Assembly Code for μGo



What to do in this Assignment?

- To accomplish the last step of building your μGO compiler,
  - which converts the μGO program into the Java assembly code.

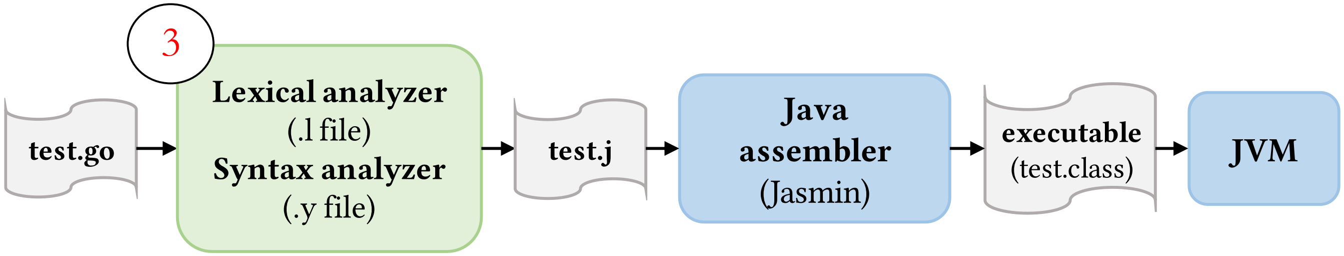


3. Code Generation
To inject the Jasmin assembly instructions into your **flex/bison** code developed in the previous assignments.

What to do in this Assignment?

- Your compiler generates the Jasmin assembly code (***test.j***) for the given input program (***test.go***).
- The generated code will then be translated to the Java bytecode (***test.class***) by the Java assembler, **Jasmin.**
- The generated Java bytecode should be run by the Java Virtual Machine (JVM).

- **In this assignment,**
  - **TAs give the score based on your .j file and the JVM execution results.**
  - **The flex/bison files need to print out the error messages.**



Basic Requirements (1/3)

- **Handle variable declarations using local variables. (20pt)**
  - When a variable is declared without initialization, your compiler should assign the initial value 0 to the variable.
  - Do not worry about the type casting in variable declaration.

Example

```
var x int  // x should be initialized to 0
```

Example

```
var x int = 2.5 // Don’t need to deal with this case
```

Basic Requirements (2/3)

- **Handle arithmetic operations for integers and float32. (30pt)**
  - I.e., implement `+`, `-`, `*`, `/`, `%`, `++`, `--`, `+=`, `-=`, `*=`, `/=` and `%=` operators. Note: `++` and `--` are postfix expressions.
  - Once float32 variable/constant involves into `mod (%)` operation, your compiler should take it as **illegal action**.

- **Handle the print and println function. (10pt)**
  - Do not worry about `println(x++)`and `print(x--)`.

Basic Requirements (3/3)

- **Handle the if…else if…else statement. (40pt)**
  - Do not consider the scoping.

  Example

```
if (x == 0) {
   var y  = 3
   print(y)
}
```

- **When ERRORs occur during the parsing phase,**
  - **TAs expect your compiler to print out ALL error messages, as Assignment 2 did, and**
  - **DO NOT generate the Java assembly code (.j file).**

Advanced Features

- Handle the for statement. (10pt)
- Handle the scoping for JVM. (10pt)
- Handle user defined function. (10pt)
- Provide a README explaining WHAT and HOW advanced functions you implemented if you attempt to implement these features.

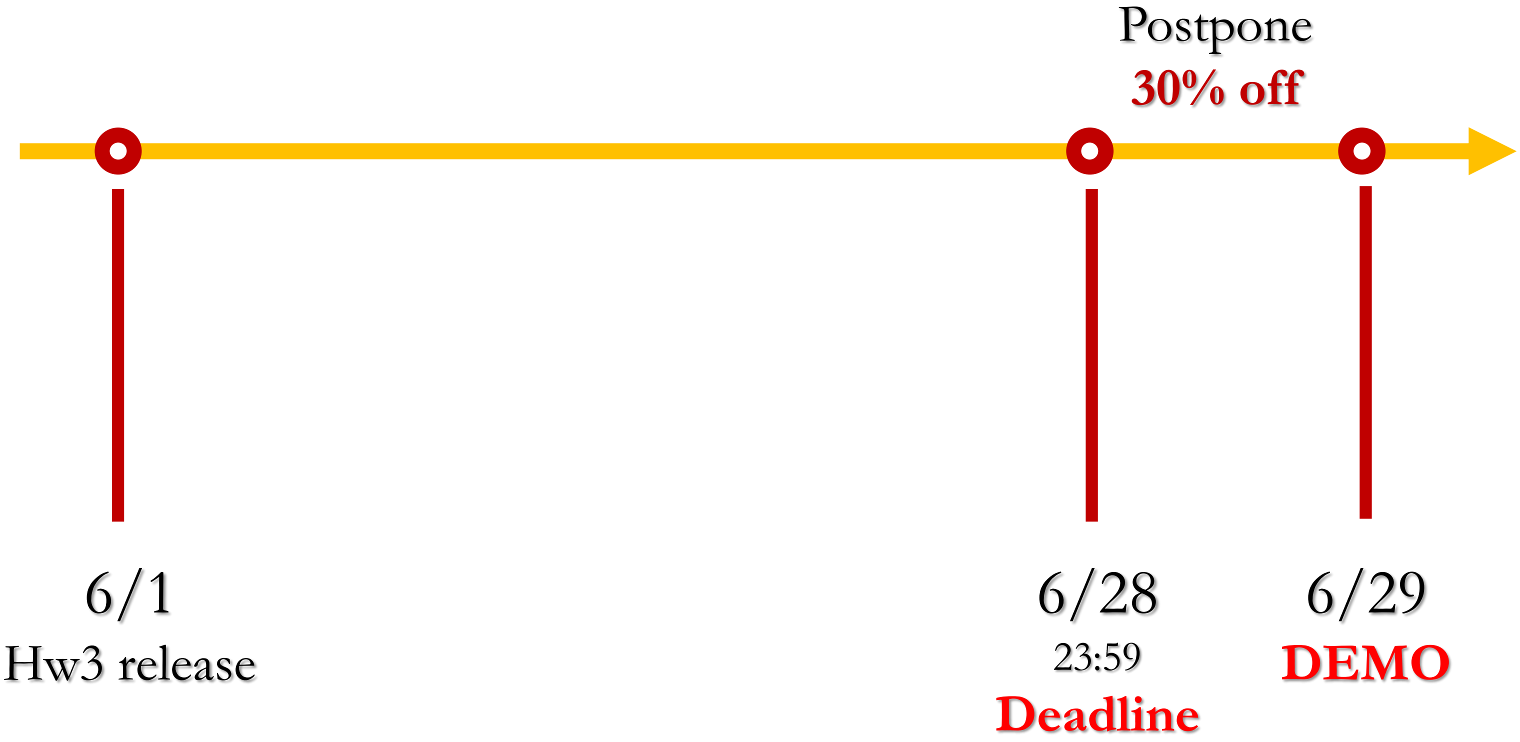
Requirements of Your Uploaded Codes

- Upload your homework to Moodle.
- Only .zip and .rar types of compression are allowed.
- The directory should be organize as:

```
Compiler_StudentID_HW3.zip
 ↳ Compiler_StudentID_HW3/
       ↳ input/
          jasmin_example/ (Java assembly code example)
          Makefile
          compiler_hw3.l
          compiler_hw3.y
          jasmin.jar (Java assembler)
          README (If any)
          xxx.h (Header file if required)
```

- **You will get 10% discount if your programs were uploaded in incorrect format!!!**

Deadline



Postpone

30% off

6/1

Hw3 release

6/28

23:59

Deadline

6/29

DEMO

About DEMO

- Demo time is 9:00-14:00, 6/29
  - The demo is partitioned into several time periods.
- We will open a Google Form for you to register your demo time slot.
- Each time period allows less than 26 person to demo.
- You are responsible for your code.
  - If you cannot explain your code clearly, you score will be low.

- **Please come to demo ON TIME.**
- Bring your own code and development environment to demo site just in case.

Questions ?

Information of the Quiz

- 請依座位表到考試教室
- 並依座位入座
- 考試時間一個小時
- 寫完即可交卷