

Compiler Construction

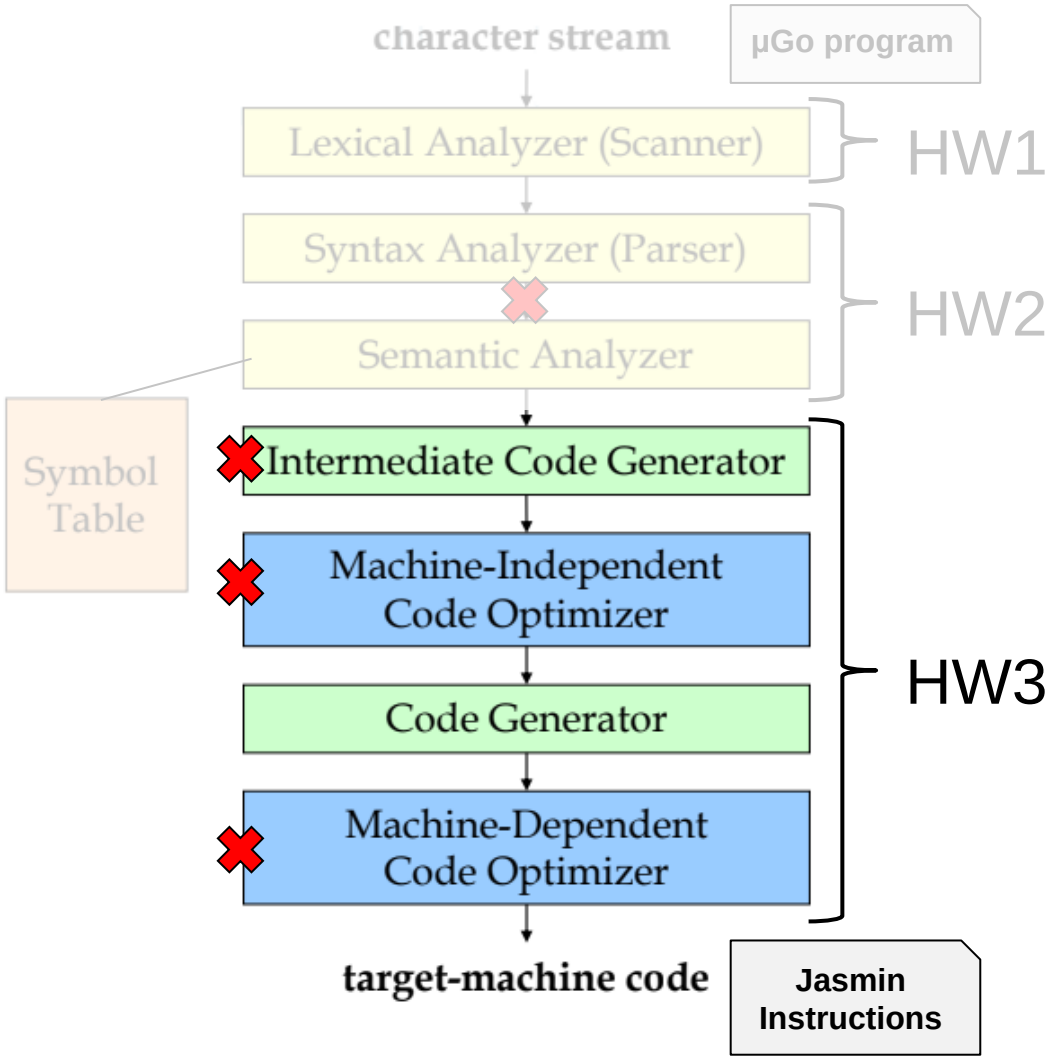
Programming Assignment 3

Generate Java Assembly Code for µGo

NCKU
National Cheng Kung University



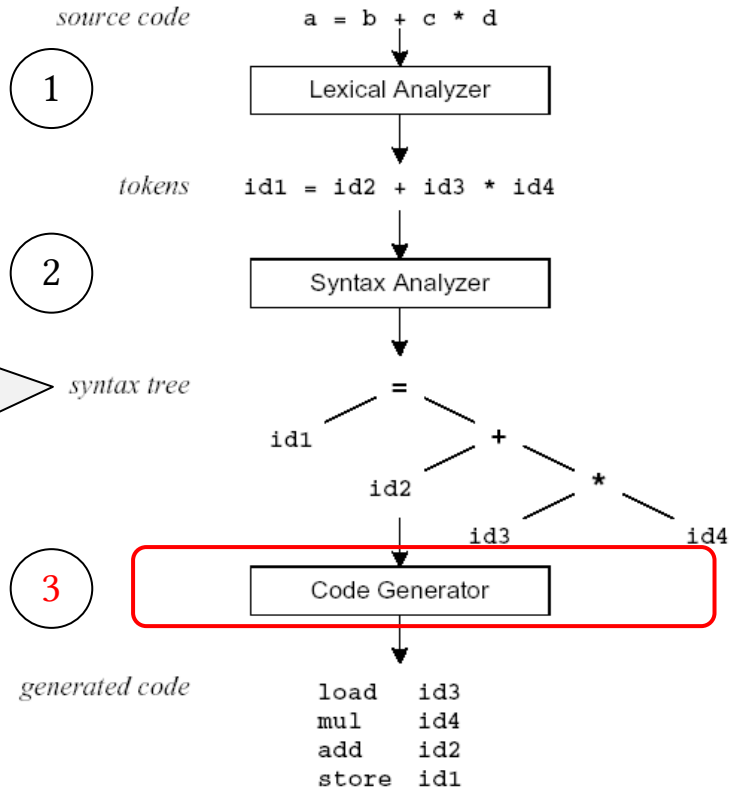
Project Outline





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.



- Code Generation:
  - ***Inject*** the Jasmin assembly instructions into your flex/bison code developed in the previous assignments.



The tutorial of Jasmin will be introduced in the course next week



Tentative Time Table (Version I)

3/6	1.	Course Introduction
3/13	2.	Overview & A Simple Compiler
3/20	3.	A Simple Compiler & Theory and Practice of Scanning
3/27	4.	Theory and Practice of Scanning & Grammars and Parsing
4/3	5.	Spring break! No Class!!!
4/10	6.	Lex (**HW #1**) & Quiz 1
4/17	7.	Grammars and Parsing
4/24	8.	Yacc (**HW #2**) & Quiz 2
5/1	9.	Midterm
5/8	10.	Top-Down Parsing
5/15	11.	Yacc & Jasmin (**HW #3**) & Quiz 3
5/22	12.	Intermediate Representations & Runtime Support **(Moved up to cover HW #3)**
5/29	13.	Bottom-Up Parsing
6/5	14.	Code Analyses and Optimizations & Quiz 4
6/12	**15.**	**Project demo (A simple compiler)**
6/19	16.	Final

↑ Your senior project demonstration event is taken place on 6/12.

← Compiler homework demo is taken place in the AFTERNOON on 6/12

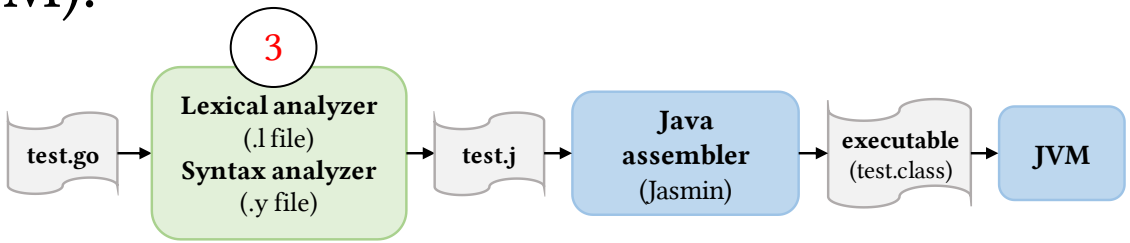
← Check Moodle carefully

1



What to do in this Assignment? (cont.)

- 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 as hw2 did.



Simple examples

µGo program

```
-5 + 3 * 2
```

Your compiler

```
ldc 5
ineg
ldc 3
ldc 2
imul
iadd
```

Jasmin Instructions

```
print("Hello")
```

Your compiler

```
ldc "Hello" ; string
getstatic java/lang/System/out Ljava/io/PrintStream;
swap
invokevirtual java/io/PrintStream/print(Ljava/lang/String;)V
```



Simple examples (cont.)

- We also give several examples in the appended document
- However, the corresponding Jasmin codes are just for reference, so you can write your own version while it can produce the same program outputs.

  - µGO Code:

```
// Precedence: ! > && > ||
true || false && !false
```

  - Jasmin Code (for reference only):

```
iconst_1    ; true (1)
iconst_0    ; false (2)
iconst_1    ; load true for "not" operator
iconst_0    ; false (3)
ixor        ; get "not" result (4) from (3)
iand        ; get "and" result (5) from (2),(4)
ior         ; get "or" result from (1),(5)
```



Assignment Requirements

- Each test case is 10pt and the total score is 130pt.
- You can judge your code locally with the attached judger.

```
$ ls
judge/  common.h  compiler_hw3.l  compiler_hw3.y  jasmin.jar  judge.conf  Makefile
$ python3 judge/judge.py
====================+=============================================================================
            Sample | Accept
====================+=============================================================================
    in01_arithmetic | ✔
====================+=============================================================================
    in02_precedence | ✔
====================+=============================================================================
         in03_scope | ✔
====================+=============================================================================
         in04_array | ✔
====================+=============================================================================
    in05_assignment | ✔
====================+=============================================================================
    in06_conversion | ✔
====================+=============================================================================
           in07_if | ✔
====================+=============================================================================
          in08_for | ✔
====================+=============================================================================
    in09_type_error | ✔
====================+=============================================================================
in10_variable_error | ✔
====================+=============================================================================
     in11_nested_if | ✔
====================+=============================================================================
    in12_nested_for | ✔
====================+=============================================================================
       in13_monster | ✔
====================+=============================================================================
Correct rate: 100.0%
Obtained/Total scores: 130.0/130
```

```
// "Hard Coding" will get 0pt.
main() {
    result = read(answer_file);
    print(result);
}
```

https://github.com/aben20807/local-judge/tree/master/examples/student



Assignment Requirements (cont.)

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

```
if (HAS_ERROR) {
    remove("hw3.j");
}
```



Submission

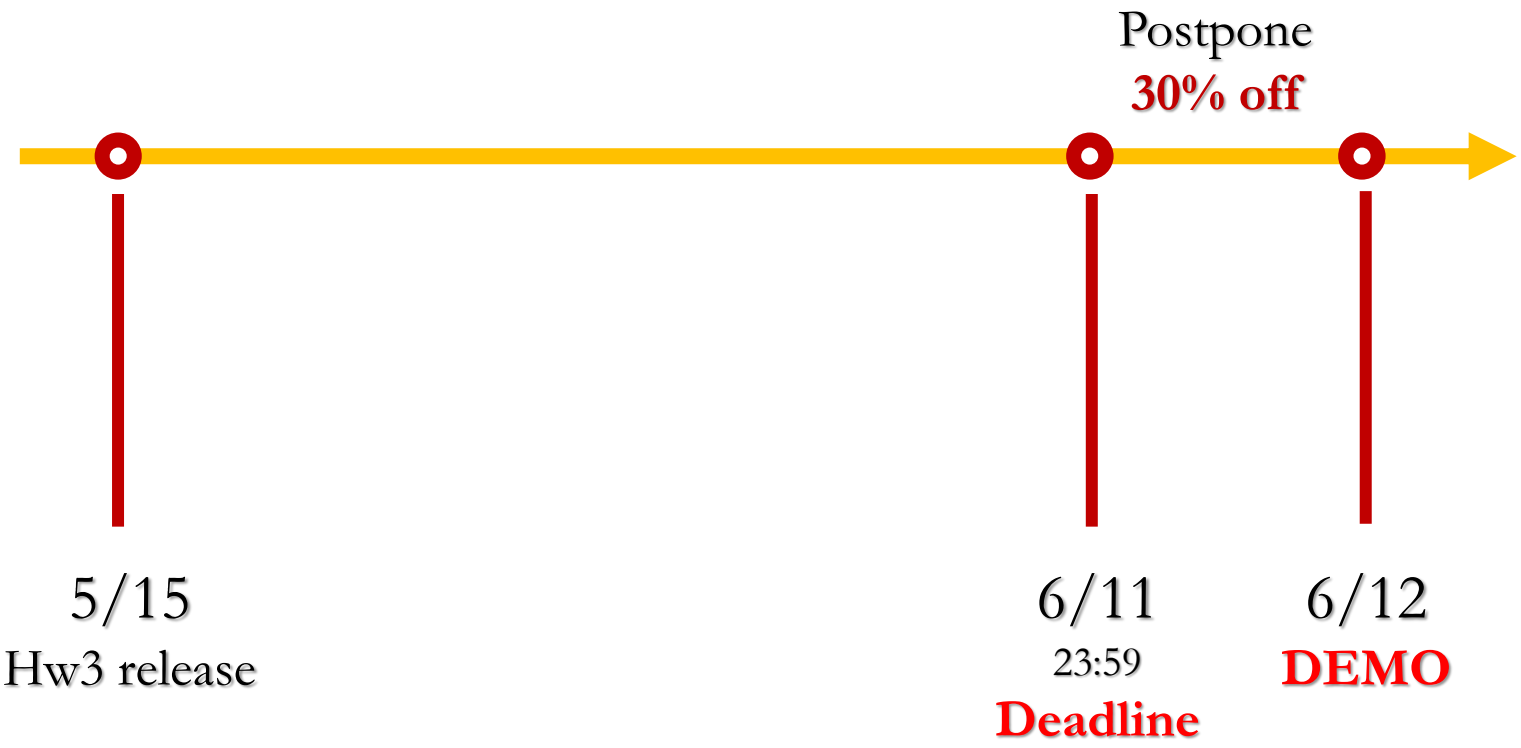
- Upload your homework to Moodle.
- The expected arrangement of your codes:
  - Only **.zip** and **.rar** types of compression are allowed.
  - The directory should be organized as:

```
Compiler_StudentID_HW3.zip/
└── Compiler_StudentID_HW3/
    ├── compiler_hw3.l
    ├── compiler_hw3.y
    ├── common.h
    ├── jasmin.jar
    └── Makefile
```

  - You will lose 10pt if your programs were uploaded in incorrect format!!!



Deadline





About DEMO

- Demo time is 10:00-17:00, 6/12
  - 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 people to demo.
- Demo place is at Room 65704
- You are responsible for your code.
  - If you cannot explain your code clearly, you score will be discounted.

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



How to Mail TAs

- Send mail to asrlab@csie.ncku.edu.tw, not any TA's mail!!
- Email subject starts with "[Compiler2020]"



QUESTIONS ?