

因應近期國內疫情第三級警戒延長，敬請您務必落實填寫本校健康關懷問卷調查及每日體溫及症狀紀錄。



Compiler Construction

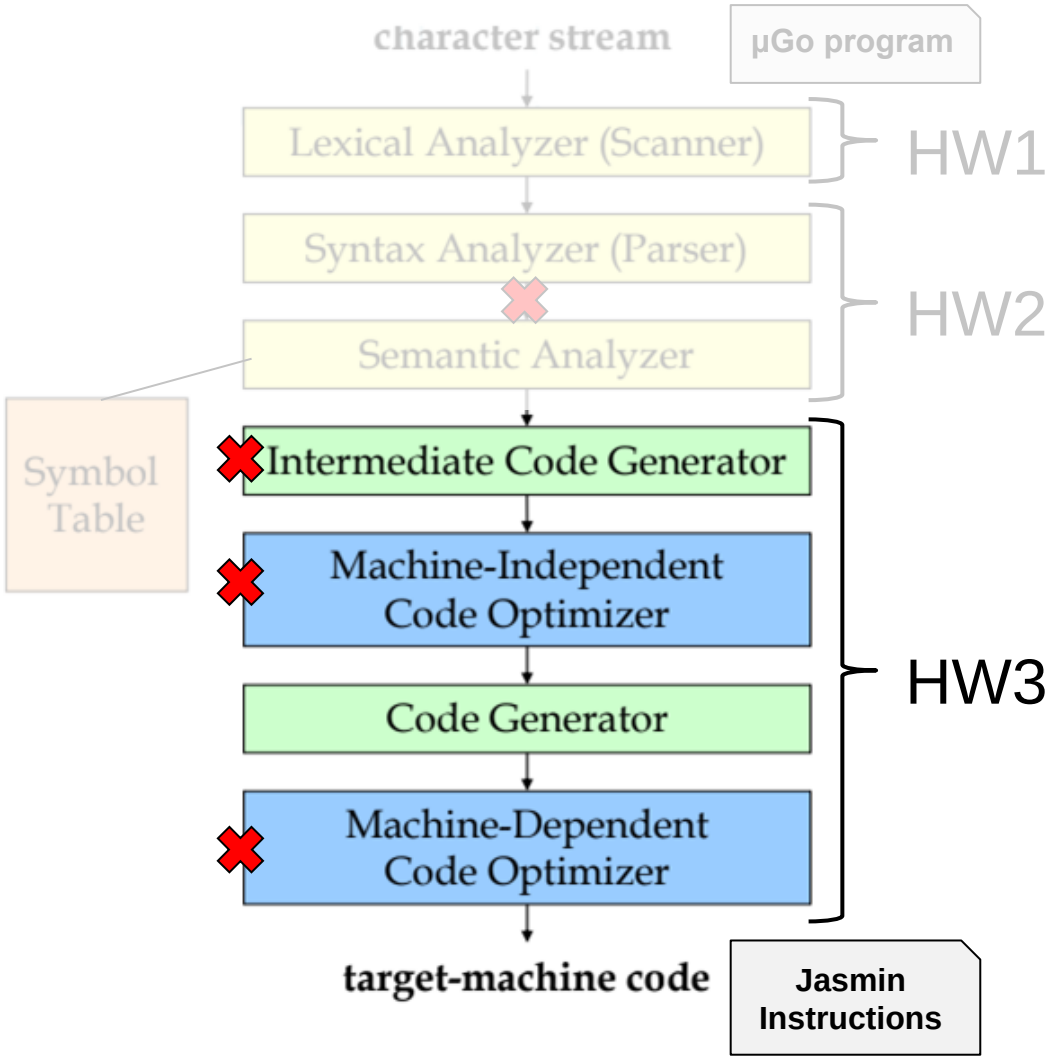
Programming Assignment 3

Generate Java Assembly Code for μC

NCKU
National Cheng Kung University



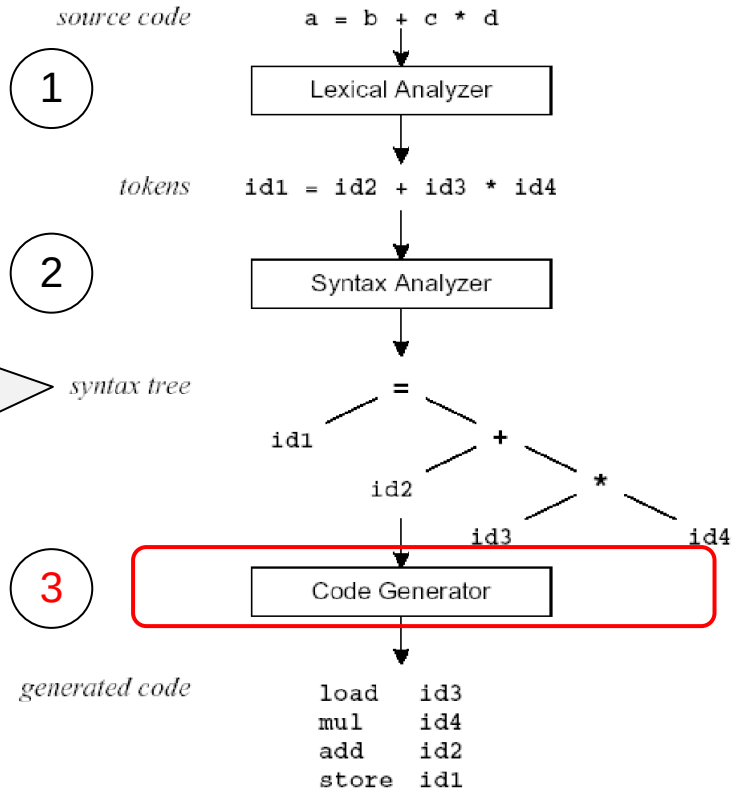
Project Outline





What to do in this Assignment?

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

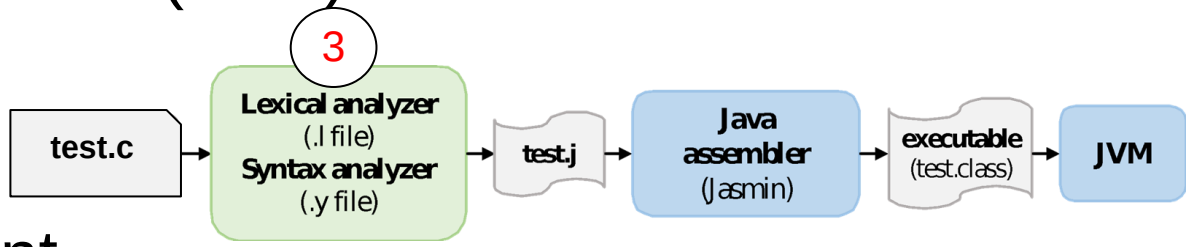


- Code Generation:
  - ***Inject*** the Jasmin assembly instructions into your flex/bison code developed in the previous assignments.



What to do in this Assignment? (cont.)

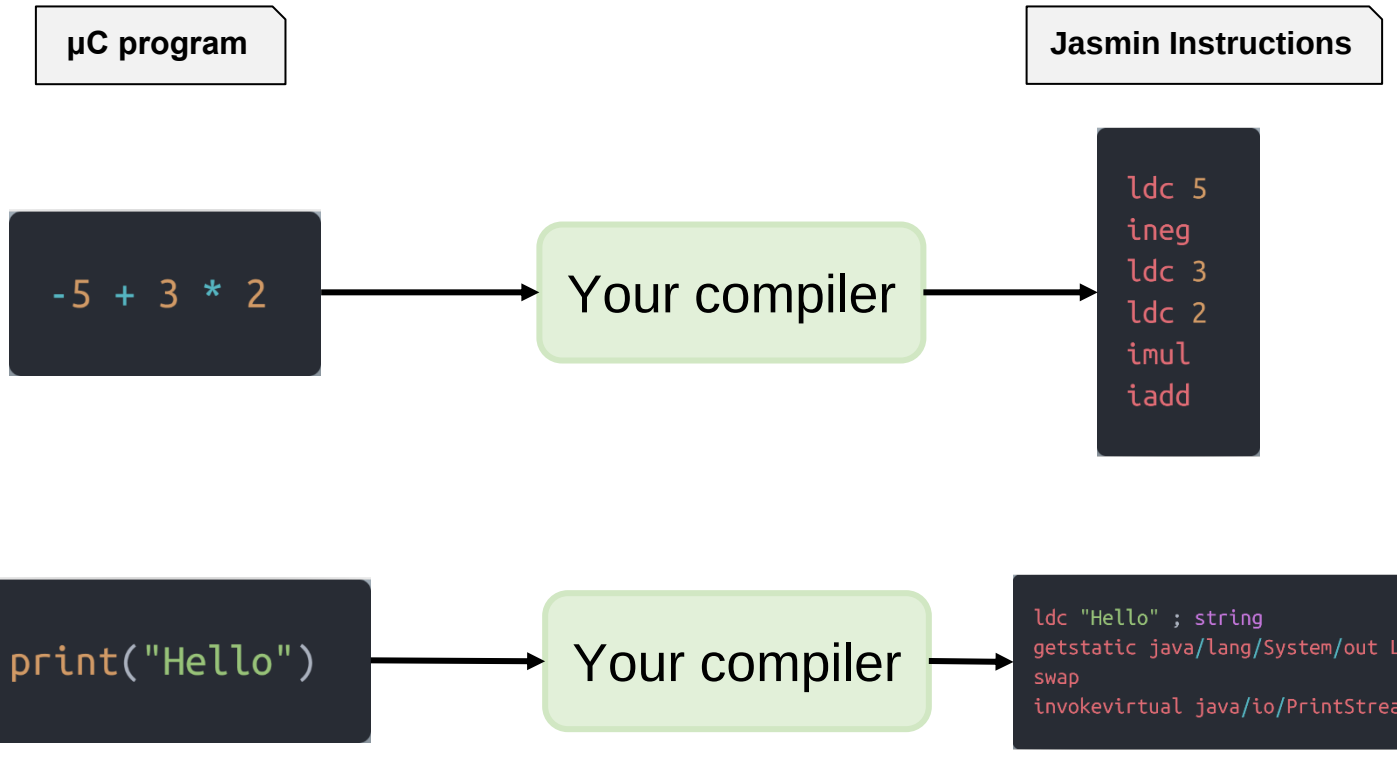
- Your compiler generates the Jasmin assembly code (test.j) for the given input program (test.c).
- The generated code will then be translated to the Java bytecode (test.class) by the Java assembler, Jasmin.
- The generated Java bytecode should be run by the Java Virtual Machine (JVM).



- 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





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.

- µC 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

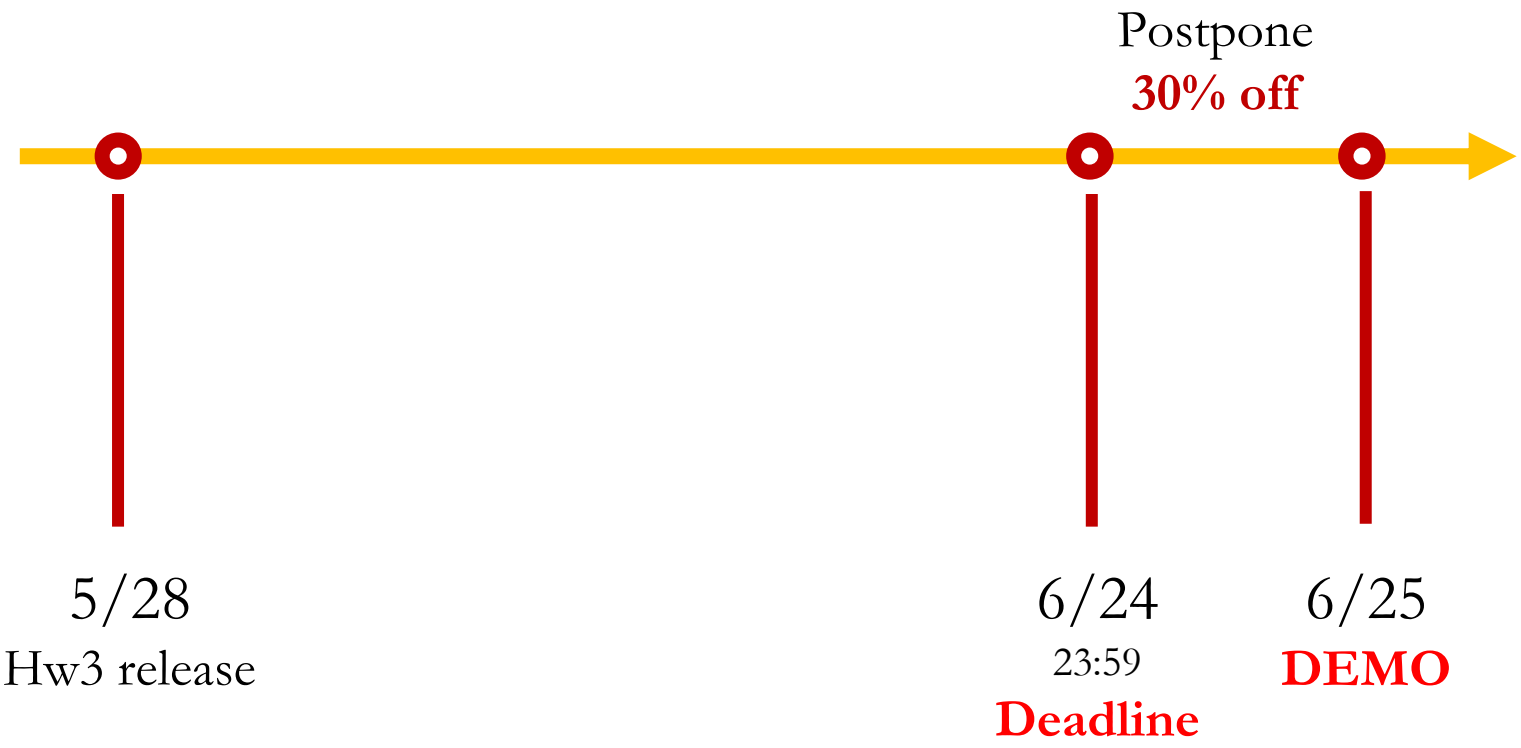
- Upload your homework to Moodle.
- The expected arrangement of your codes:
  - Only .zip and .rar types of compression are allowed.
  - The directory should be organized as:

```
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





How to Mail TAs

- Send mail to asrlab@csie.ncku.edu.tw, not any TA's mail!!
- Email subject starts with "[Compiler2021]"



QUESTIONS ?