

# Compiler

## – Assignment 3 –

JIEUNG KIM

[jieungkim@yonsei.ac.kr](mailto:jieungkim@yonsei.ac.kr)



# Assignment 3 Instruction

- Goal
  - Modify provided python files (`codegen.py` and `resolver.py`) to pass all test cases in `codegen_test.py` and `resolver_test.py`
- Due and how to submit
  - Dec. 8, 2024, 11:59pm
  - Submit your `codegen.py` and `resolver.py` via LearnUs without changing the name.
  - You do not need to submit other files.
    - However, test files (`codegen_test.py` and `resolver_test.py`) are to check your grade by yourself before submission.



# Assignment 3 Instruction

- How to
  - Please read the written instructions as comments when you implement your code to avoid disadvantages of your grades.
  - Modify definitions in the given Python files (`codegen.py` and `resolver.py`)
  - The parts that you have to modify are specified with comments.
  - You may need to refer to the PLY manual.
    - <https://www.dabeaz.com/ply/ply.html>
  - You can get help from the following auto code generation tools, but you should not share your answers with your classmates.
    - Copilot (<https://github.com/features/copilot>)
    - ChatGPT (<https://openai.com/blog/chatgpt>)



# Assignment 3 Instruction

- Directory structure (1/3)
  - Files are provided as a single tar.gz file, and it includes the following items.
  - `ply/*`
    - PLY files from PLY repository (<https://github.com/dabeaz/ply>).
    - You do not need to touch or look at those files.
  - `README.md`
    - A definition of ToyPL and a few instructions of our assignment.
  - `lexer.py` and `lexer_test.py`
    - A lexer its test file.
  - `parser.py` and `parser_test.py`
    - A parser its test file.



# Assignment 3 Instruction

- Directory structure (2/3)
  - `resolver.py`
    - A symbol table generator that you have to work on
  - `resolver_test.py`
    - A test file for our symbol table generator that uses multiple pre-defined test cases in the assessment directory
  - `codegen.py`
    - A codegen file that you have to work on
  - `codegen_test.py`
    - A test file for our codegen that uses multiple pre-defined test cases in the assessment directory



# Assignment 3 Instruction

- Directory structure (3/3)
  - `examples/`
    - 7 sample ToyPL programs.
  - `assessment/`
    - Sample programs (9 programs) written in ToyPL along with their corresponding lexing results (9 result files.)
    - `*.topyl`: ToyPL programs
    - `*_lex_result`: (desired) lexing results of ToyPL programs
    - `*_yacc_result`: (desired) parsing results of ToyPL programs
    - `*.ir`: compile results (IR programs)
    - `*_resolver_result`: generated symbol table



# Assignment 3 Instruction

- Grade
  - Run “codegen\_test.py” and “resolver\_test.py”, and it will automatically provide your grade
    - Type “python codegen\_test.py” and “python resolver\_test.py”
    - They will check the generated three address code with your code generator and compare it with the expected output.
  - Test cases may not cover all desired functionalities, but passing them should be considered the minimum requirement.

