

## **Simple Compiler v3 Assignment 3 – Extended Features**

Be sure to fully review and comply with the SimpleO Language Specification document for this and all future SimpleCompilerv3 assignments.

In this assignment, the following functionality will be developed:

1. Declaration of a variable without assignment
  - a. (e.g., “int x;”)
2. Allowing your function to take a single integer argument
  - a. (e.g., “myFunction(int argA)”) )
3. Assignment of a variable after declaration
  - a. (e.g., “x = 10;”)
4. Assignment of integers, unary, and binary expressions
  - a. (e.g., “x = !y;”, “x = y + 5;”, “x = y + z;”)
5. Reassignment of a variable’s value
  - a. (e.g., “x = 20;”)
6. Variable increment
  - a. (e.g., “x++;”)
7. All functionality from the previous assignments (which is included in the test-cases directory)
8. \*\*\* Be sure you are reserving enough stack space for your local variables in the function prologue. You will lose points if you are not.

\*\*\*\*All provided test cases must work without error or warning to receive full credit. These test cases are not comprehensive and there are likely edge cases that may or may not work that are not covered by the test cases. However, the test cases provide sufficient coverage for the functionality requested (and they align with the SimpleO Language Specification document). The test cases directory is cumulative and later assignments include all test cases from the previous assignments to make sure you don’t break functionality as you develop your compiler. You may develop your own test cases, though I will only be using the supplied test cases to evaluate your submissions for each assignment.

\*\*\*\*Do not modify the MakeFile, shell scripts, file names, or directory structure. Any modification can lead to it not working in my environment when I grade it and you will not receive credit if I cannot evaluate your compiler. Only Ubuntu 22.04 LTS is supported. No other environments have been validated and are not supported.

**Be sure to follow a professional coding standard, including the naming of your variables, etc. You should use descriptive names that are appropriate. Nonsensical or whimsical names will lose points.**

Additionally, you must attest and add the following honor statement to your Canvas submission (in the Canvas submission comments):

*I, <state your name>, attest that this submission is my own work. I have not worked with anyone else on my implementation and have not used AI or other unpermitted sources to help me with my solution, in accordance with our definition of class cheating shown in Lecture 1.*

**Any submission that does not include this attestation will not receive any credit.**