Assignment 5
Translator

Note: All code must be compliable and executable on Windows systems.
. Code that does not compile will receive 0 marks. There will be no evaluations. C/C++ and JAVA will be accepted.

By now, you have developed both a lexical analyzer and a parser for the language JAVA--. Your task now is to develop the translator, which will produce the three-address code for a given program in JAVA--.

In order to do this, you must add appropriate actions to your previously defined grammar.

Like the parser, the translator shall take the file *words.txt* generated by the lex, along with the symbol table. In the case of incorrect syntax or semantics, it shall gracefully halt execution and print an appropriate error message. In the case of completely correct syntax, it shall produce the three-address code, as discussed in the class.

Requirements:

- A test file that successfully runs on your compiler. The file should contain a loop, a condition, input, output, and a mathematical expression. Name this *test.cmm*
- Complete code of your project up till this point, which will generate, in addition to the previous phases' requirements, the following text files:
  •*tac.txt*: Three address code for any JAVA-- code provided as input
  •*translator-symboltable.txt*: the names, datatypes, and relative addresses of all the variables. This file should contain data variables only, not function names.

This will be a continuation of Phase 4. The input will still be a .cmm file containing JAVA-- code. Your compiler must run the lex first, and use the freshly generated output to subsequently run the parser/translator. Do not submit an isolated translator with static *words.txt* and *symboltable.txt* files. Such submissions will not be checked.

BONUS: Nested conditionals & loops, and functions

Interface:

A main program that takes ask for file name on console. (If the file name is hardcoded in the code it will result in deduction of half marks of assignment.)
Display a proper massage on console that asks for file name, passing file name in argument of main function will also be considered wrong.
Read the instruction carefully while implementing your assignment. Unnecessary questions which have already been instructed here would not be answered. Also, don't send me queries to check if output is correct. Only the submission on slate will be considered.

Submission Instructions:
Submit files unzipped
Do not submit executables.
Submit exactly once per group.

*This deliverable will be marked without an evaluation, so make sure the code is compilable **ON Window**.*

There will be zero tolerance for plagiarism. Your assignments will be checked far more thoroughly than you are anticipating. Once detected, no appeals for removal of plagiarism will be entertained.