Birzeit University logo

College of Engineering & Technology

***Computer Science Department***

# COMP439 Final Project Fall 24/25 Due midnight : 10-01-2025

# ======================================================================

***Givezn the grammar defined by the following set of production rules which represent some programming language :***

program 🡪 lib-decl declarations ( function-decl )\* block **exit**

lib-decl 🡪 # **include** **<** file-name **>** **;** lib-decl | λ

declarations 🡪 const-decl var-decl

const-decl 🡪 **const** data-type const-name **=** value **;** const-decl | λ

var-decl 🡪 **var** data-type name-list  **;** var-decl | λ

name-list 🡪 var-name more-names

more-names 🡪  **,** name-list | λ

data-type 🡪 **int** | **float** | **char**

function-decl 🡪 function-heading declarations block ;

function-heading 🡪 **function** function-name ;

block 🡪 **newb** stmt-list **endb**

stmt-list 🡪 statement **;** stmt-list | λ

statement 🡪 ass-stmt | inout-stmt | if-stmt | while-stmt | block | repeat-stmt

| function-call-stmt

ass-stmt 🡪 var-name **:=** exp

exp 🡪 term exp-prime

exp-prime 🡪 add-oper term exp-prime | λ

term 🡪 factor term-prime

term-prime 🡪 mul-oper factor term-prime | λ

factor 🡪 ( exp ) | “var-name” | “const-name” | value

add-oper 🡪 **+** | **-**

mul-oper 🡪 **\*** | **/** | **mod | div**

inout-stmt 🡪 **cin** **>>** var-name | **cout** **<<** name-value

if-stmt 🡪 **if** ( condition ) statement else-part

else-part 🡪 **else** statement | λ

while-stmt 🡪 **while** ( condition ) **newb** stmt-list **endb**

repeat-stmt 🡪 **repeat** stmt-list **until** condition

condition 🡪 name-value relational-oper name-value

name-value 🡪 “var-name” | “const-name” | value

relational-oper 🡪 = | =! | < | =< | > | =>

name 🡪 letter ( letter | digit )\*

value 🡪 integer-value | real-value

integer-value 🡪 digit ( digit )\*

real-value 🡪 digit ( digit )\*. digit ( digit )\*

function-call-stmt 🡪 **call** function-name

**Note:** The tokens in **bold** are reserved words or standard identifiers (library functions or procedures).

Write a **recursive descent** parser for the above grammar.

\* Use only **Java** as a host language.

\* You should work **individually only**, any signs of cheating will be penalized severely,

- more than 70% copy will not be graded and receive zero.

- 20-70%, will be graded and receive the grade equals the complement of the copy percentage.

\* Your program will be tested with my text files (programs), either it works correctly or doesn’t.

\* No programs will be accepted after the deadline for any reason whatsoever.

\* In the ERROR function, report the error clearly and precisely showing the **line** and **token** where

the Error occurs and exit the program (panic mode) error handling.

\* Submit only the source code by replying to the message “**439-Project-F24**” on Ritaj web page.