

# Compiler Project

Mohammad Alomar	2	14
Mostafa Hani	2	24
Menna Ahmad	2	25
Mostafa Mohammad	2	23

This is a compiler to a language mixed from cpp and python  
using lex and yacc

How to run:

Bison parser.y -d

Flex lexer.l

gcc .\lex.yy.c .\parser.tab.c -o comp

.\comp.exe test.txt

"int" INT

"float" FLOAT

"string" STRING

"char" CHAR

"bool" BOOL

"if" IF

"else" ELSE

"for" FOR

"while" WHILE

"do" DO

"continue" CONTINUE

"true" BOOL\_VALUE

"false" BOOL\_VALUE

"TRUE" return BOOL\_VALUE

"FALSE" return BOOL\_VALUE

"and" LOGICAL\_AND

"or" LOGICAL\_OR

"not" LOGICAL\_NOT

"etba3" PRINT

"void" VOID

"dalla" FUNCTION\_KEY

"return" RETURN

"switch" SWITCH

"break" BREAK

"case" CASE

"default" DEFAULT

"==" EQUAL

"!=" NOT\_EQUAL

">" GT

"<" LT

"<=" LTE

">=" GTE

"=" EQ

"," SEMICOLON

"%" MOD

"+" ADD

"-" SUB

"\*" MUL

"/" DIV

"^" POW

"|" BITWISE\_OR

"&" BITWISE\_AND

"<<" SHL

">>" SHR

"++" { return INC; }

"--" { return DEC; }

"(" LPAREN

")" RPAREN

"{" LBRACE

"}" RBRACE

"+=" { return ASSIGN\_ADD; }

"-=" { return ASSIGN\_SUB; }

"\*=" { return ASSIGN\_MUL; }

"/=" { return ASSIGN\_DIV; }

"%=" { return ASSIGN\_MOD; }

"\*\*=" { return ASSIGN\_POW; }

"&=" { return ASSIGN\_AND; }

"|=" { return ASSIGN\_OR; }

"~=" { return ASSIGN\_NOT; }

>>=" { return ASSIGN\_SHR; }<<=" { return ASSIGN\_SHL; }

QUAD	DESCRIPTiON
JUMP -- -- L	Unconditional jump to label L
ADD x1, x2, res	Add and store in res
JUMP FALSE CONDITION, temp2, -- EXITLOOP1	Jump to exitloop if condition is false
PRINT "value"	Print
ASSIGN, temp3, --, s	S = temp3