

C-Compiler

Team 9



*Team Members: -*

*Ahmed Mohamed Salah 1152114*

*Ahmed Shams 1124313*

*Mohamed Bassel 1152253*

*Mustafa Mufeed 1162249*

**Project Description**

Simple C language Compiler that is designed to compile main functions of C language components.  
the compiler is design is divided into two phases

**Phase-1**

1. Lex file that contain main tokens used in the project.
2. YACC file that contain the Production Rules used in the language as writes an indictor to file either the compiling is success or fauilor.
3. Simple C# windows application GUI used to generate Y-Files and EXE files as well as Compiling Test Cases.

**Phase-2**

1. Symbol table
2. Semantic Analyzer
3. Enhanced GUI
4. Arrays Production Rules

**Technologies Used**

1. Flex: Compile LEX files
2. Bison: Compile YACC file
3. DEV-C: Generate Executable Files

**List of Tokens**

|  |  |
| --- | --- |
| OCBRACKET | { |
| CCBRACKET | } |
| ORBRACKET | ( |
| CRBRACKET | ) |
| SEMICOLON | ; |
| COLON | : |
| COMMA | , |
| PLUSEQUAL | += |
| MINUSEQUAL | -= |
| MULTIPLYEQUAL | \*= |
| DIVIDEEQUAL | /= |
| GREATERTHAN | > |
| LESSTHAN | < |
| GREATERTHANOREQUAL | >= |
| LESSTHANOREQUAL | =< |
| EQUALEQUAL | == |
| NOTEQUAL | != |
| PLUS | + |
| MINUS | - |
| MULTIPLY | \* |
| DIVIDE | / |
| EXPONENT | ^ |
| ASSIGN | = |
| REM | % |
| AND | && |
| OR | || |
| NOT | ! |
| WHILE | WHILE |
| FOR | FOR |
| IF | IF |
| ELSE | ELSE |
| PRINT | PRINT |
| BOOL | Data Type |
| INT | Data Type |
| CHAR | Data Type |
| STRING | Data Type |
| CONST | Data Type |
| DO | DO |
| BREAK | BREAK |
| SWITCH | SWITCH |
| CASE | CASE |
| FALSE | Boolean Value |
| TRUE | Boolean Value |
| RET | Return |

**Associativity List**

|  |  |
| --- | --- |
| ASSIGN | Left |
| GREATERTHAN | Left |
| LESSTHAN | Left |
| GREATERTHANOREQUAL | Left |
| LESSTHANOREQUAL | Left |
| EQUALEQUAL | Left |
| NOTEQUAL | Left |
| AND | Left |
| OR | Left |
| NOT | Left |
| PLUS | Left |
| MINUS | Left |
| DIVIDE | Left |
| MULTIPLY | Left |
| REM | Left |

**Production Rules**

Variables and Constants declaration.

Mathematical and logical expressions.

Assignment statement.

If-then-else statement, while loops, repeat-until loops, for loops, switch statement.

Block structure (nested scopes where variables may be declared at the beginning of blocks).

Functions.