

Compiler-Part-1

1- تعليمة تعريف تابع (مع او بدون وسطاء) ويقوم بارجاع قيمة او عدم ارجاع قيمة :

```
include:headers program
;

program: function main '(' ')' '{' body return '}'

;

function: main '(' parameter ')' '{' body return '}' function
|
;

parameter: datatype ID init
|parameter COMMA parameter
|
;
```

2- تعليمة تعرف متغير ونقوم بإسناد اما قيمة (ثابت) او متغير او تعبير رياضي :

```
declaration: datatype ID '=' value
| datatype ID '=' expression
| datatype ID
;

init: '=' value
|
;

expression: expression arithmetic expression
| value
;
```

3- تعليمة تعريف حلقة فور مضمنة (Nested for) :

```
55
56 for:FOR '(' statement ';' condition ';' statement ')' '{' body '}'
57 | FOR '(' statement ';' condition ';' statement ')' statement ';'
58 ;
59
```

4- تعليمة تعريف تعبير (تطبيق عمليات منطقية او رياضية) :

```
expression: expression arithmetic expression
| value
;

arithmetic: ADD
| SUBTRACT
| MULTIPLY
| DIVIDE
;

relog: OR
| AND
;

logical: logical relog logical
| value
| expression
;

relop: LT
| GT
| LE
| GE
| EQ
| NE
;
```

5- تعليمة return مثال return y; return; :

```
127
128  return: RETURN value ';'
129  | RETURN expression ';'
130  | RETURN ';'
131  |
132  ;
133
```

6- تعليمة if else مضمنة :

```
3
4  if: IF '(' condition ')' '{' body '}' else
5  | IF '(' condition ')' '{' body '}'
6  | IF '(' condition ')' statement ';'
7  ;
8
9  else: ELSE '{' body '}'
10 |
11
```

شرح بسيط عن الملفين (lexer.l & parser.y)

تم بناء ال rules الاساسية للغة C++ في ملف ال lexer.l حيث تم تعريفها ك Token في ملف ال parser.y وبناء جميع التعليمات مع مراعاة جميع الحالات الممكنة التي قد تواجه البرنامج . بالإضافة الى انه تم كتابة بعض التعليمات الاضافية لتحقيق المتطلبات المذكورة مثل ال expression وغيرها .