Reza Naghdi (C++ Grammar)

1- Program → Headers | FunctionList

2- Headers → #Keywords1 <HeadNames>
3- Keyword1 → include | define
4- HeadNames → ID | ID.h | STR

5- FunctionList → FunctionList Function | Function
6- Function → TYPEst ID (Params) FBody

7- Params → HVParams | λ
8- HVParams → HVParams , Exp | Exp

9- FBody → {StatementList} | {StatementList return Val;}

10- StatementList → StatementList Statement | Statement
11- Statement → IFst | FORst | WHILEst | DOWHILEst | SWITCHst | ASSIGNst | CALLst | IN
| OUT | Var

12- IFst → if (Condition) Body EPart
13- EPart → ElseIfPart ElsePart
14- ElseIfPart → else if (Condition) Body ElseIfPart | λ
15- ElsePart → else Body | λ

16- FORst → for(SCN;Condition;Exp) Body
17- SCN → TYPEst ID = Val | ID = Val

18- WHILEst → while(Condition) Body
19- DOWHILEst → do {Statementlist} while(Condition);

20- SWITCHst → switch(Exp){CaseList DefCase}
21- CaseList → CASEst CaseList | CASEst
22- CASEst → case Val:StatementList break;
23- DefCase → default:StatementList | λ

24- ASSIGNst → ID SOP Exp;
25- CALLst → ID(Params);

26- Var → TYPEst VDef1;
27- VDef1 → VDef2 | VDef2 , VDef1 | VDef2 = VDef1

28- VDef2	→	ID | ID = Val

29- IN	→	cin >> CI;
30- CI	→	ID | ID >> CL

31- OUT	→	cout << CO;
32- CO	→	PR | PR << CO
33- PR	→	Exp | Val

34- Condition	→	Exp COP Exp | Condition && Exp COP Exp | Condition || Exp COP Exp
35- Body	→	Statement | {StatementList}
36- Exp	→	Exp + Term | Exp – Term | Term
37- Term	→	Term % Factor | Term * Factor | Term / Factor | Factor
38- Factor	→	ID | OOP ID | ID OOP | NO | !Facor | (Exp)

39- OOP	→	++ | --
40- COP	→	> | < | >= | <= | != | ==
41- SOP	→	= | += | -= | /= | *= | %=
42- Val	→	NO | CHR | STR |true | false
43- TYPEst	→	bool | int | char | double | float | void