

# Building a C++ Compiler

<b>Name</b>	Pratik Byathnal	Lamya Bhasin	Sujeeth AV
<b>SRN</b>	PES1201700272	PES1201701244	PES1201700958

6th Sem; G section;

## CFG

**S** -> B | Func | DeclExp | ClassDef

**B** -> int main ob cb ofb Start | Func | P return Id; cfb

**P**-> cout << Str | << var\_list

**ClassDef** -> Class ClassName ofb AccessSpecifier : | Expression | Func cfb;

**ClassName** -> Id

**AccessSpecifier** -> AccessType Id;

**AccessType** -> private | public | protected

**Func** -> type name ob Arg | lambda cb ofb Stmt; cfb Func | B | lambda

**Arg** -> type Id | type Id Arg|Arg |, Arg | lambda

**Stmt** -> assignExp; Stmt | Array; Stmt | matched | Unmatched |  
ForExpr | WhileExpr | obj

**Array** -> type Id V | keyword Id V = ofb range cfb| lambda

**Start** -> Stmt; Start | ForExpr | WhileExpr | ifExpr | DeclExp Start | P | lambda

**Matched** -> if ob Condition cb ofb Matched cfb else ofb Matched cfb | Stmt

**Unmatched** -> if ob Condition cb ofb Stmt cfb| if ob Condition cbMatched else  
Unmatched

**ForExpr** -> for ob DeclExp | Expression ; Condition; Un | assignExp cb ofb Stmt; Start  
cfb

**Condition** -> Expression Rel Expression | Expression Log Expression |  
Bool | lambda

**WhileExpr** - while ob Condition cb ofb Stmt; Start cfb

**Obj** -> ClassName Id

**DeclExp** -> type assignExp | type Id;

**AssignExpr** -> Id = Expression;

**Expression** -> Expression + Expression1 | Expression - Expression1 | Expression

**Expression1** -> Expression1\*Expression2 | Expression1/Expression2 | Expression2

**Expression2** -> Id | Num

**Relic\_expr** -> return Bool;

**Type** -> int | char | float | short | short int | long int

**Beta** -> Letter | \_ | Digit B | lambda

**var\_list** -> id >> var\_list | id

## Token

<Assignment, line\_no>

<Operators, type, line\_no>

<Identifier, data\_type, value,scope, line\_no>

## Operators

< Unary\_op , ofb ++ , -- cfb>

< arith\_op , ofb + , - , \* , / , % cfb>

< Rel , ofb > , < , <= , >= , != , == cfb>

< Log , ofb && , || , ! cfb>.

< const , ofb [0-9] + , [a-z A-Z] cfb>

< Identifiers , ofb [ a-z A-Z ] ob [ a-z A-Z ] \_ [ 0 -9 ] cb\* , \_ ob [ a-z A-Z ] \_ [ 0 - 9 ] cfb\* cfb>