For More Info :

<http://www.parashift.com/c++-faq/yaccable-grammar.html>

<http://gccxml.github.io/HTML/Index.html>

<http://www.codesynthesis.com/~boris/blog/2010/05/03/parsing-cxx-with-gcc-plugin-part-1/>

<http://digitocero.com/en/blog/exporting-and-visualizing-gccs-abstract-syntax-tree-ast>

Primary not supported syntax and Limitation

* Initialing in declaration : int a = 4;
* Just *Primitive Type Declaration* within *BLOCK*
* Class has specific syntax for its member.
* Can not Return Pointer types
* Can not define var with void type

Program -> DeclarationList

DeclarationList -> Declaration DeclarationList | ε

Declaration -> FunctionDeclaration | ClassDeclaration

------------------------------ + Class ----------------------------

ClassDeclaration -> **class** **ID** OptionalSuperclass **{** MemberDeclarations **} ;**

OptionalSuperclass -> **:** **public** **ID** | ε

MemberDeclarations -> FieldDeclarationList **public:** MethodDeclarationList | ε

FieldDeclarationList -> FieldDeclaration FieldDeclarationList | ε

FieldDeclaration -> OptionalStatic VariableDeclaration

OptionalStatic -> **static**| ε

MethodDeclarationList -> MethodDeclaration MethodDeclarationList | ε

MethodDeclaration -> OptionalStatic FunctionDeclaration

------------------------------ - Class ----------------------------

------------------------------ + Function -------------------------

FunctionDeclaration -> FunctionType **ID** (OptionalParameterList ) Block

FunctionType -> **void**| TypeName

OptionalParameterList -> Parameter MoreParameters | ε

Parameter -> TypeName Variable

MoreParameters -> **,**Parameter MoreParameters | ε

Block -> **{** DeclarationOrStatementList **}**

DeclarationOrStatementList -> DeclarationOrStatement DeclarationOrStatementList | ε

------------------------------ - Function -------------------------

------------------------------ + Variable -------------------------

TypeName -> PrimitiveType | **ID**

PrimitiveType -> **int**| **bool** | **float**

VariableDeclaration -> TypeName RestOfVariableDeclaration

RestOfVariableDeclaration -> Variable MoreVariables ;

OptionalPointer -> **\*** | ε

Dimensions -> **[ NUM ]** Dimensions | ε

Variable -> OptionalPointer **ID** Dimensions

MoreVariables -> ,Variable MoreVariables | ε

------------------------------ - Variable -------------------------

------------------------------ + Expression -------------------------

MoreExpressions -> , Expression MoreExpressions | ε

OptionalExpression -> Expression | ε

Expression -> Disjunction

Disjunction -> Conjunction MoreConjunctions

MoreConjunctions -> **||** Conjunction MoreConjunctions | ε **{||**=**Disjunction}**

Conjunction -> Equality MoreEqualities

MoreEqualities -> **&&** Equality MoreEqualities | ε **{&&**=**Conjunction}**

Equality -> Inequality RestOfEquality

RestOfEquality -> **==** Inequality | **!=** Inequality | ε

Inequality -> Sum RestOfInequality

RestOfInequality -> **<** Sum | **>** Sum | **<=** Sum | **>=** Sum | ε

Sum -> Product MoreProducts

MoreProducts -> **+** Product MoreProducts | **-** Product MoreProducts | ε

Product -> Factor MoreFactors

MoreFactors -> **\*** Factor MoreFactors | **/** Factor MoreFactors | **%** Factor MoreFactors | ε

Factor -> Primary | **+**Primary | **-**Primary | **!**Primary

**{ Primary =** **foo**[3] Or **foo**.boo Or **foo**(3,4) Or **foo** Or **45** Or **(**...**) }**

Primary -> **ID** MorePrimary | **NUM** | **true**| **false**| (Expression )

**{ \* MorePrimary =** foo**[3]** Or foo**.boo Or** foo**(3,4)** Or (4+foo[1])**.boo** **}**

**\* Each MorePrimary proceed with another MorePrimary }**

MorePrimary -> ArrayAccess | FieldAccess | FunctionCall | ε

ArrayAccess -> **[**Expression**]** MorePrimary

FieldAccess -> **.ID** MorePrimary

FunctionCall -> **(**OptionalArgumentList**)** MorePrimary

OptionalArgumentList -> Expression MoreExpressions | ε

------------------------------ - Expression -------------------------

------------------------------ + Statement -------------------------

**{** \* **PrimitiveDeclaration** : int a,b;

\* **KeywordStatement**: if/while/for/return/{...}/

\* **OtherDeclarationOrStatement** :

\*\* **RestOfVariableDeclaration:** Foo **obj1, obj2;**

\*\* **RestOfExpressionStatement:** obj1**[3]=..;** Or obj1**.boo=..;** Or obj1**(3,4)=..;**

DeclarationOrStatement -> PrimitiveDeclaration | KeywordStatement | OtherDeclarationOrStatement

**}**

PrimitiveDeclaration -> PrimitiveType RestOfVariableDeclaration

**{ \* Statement** is used in **Selection** ana **Iteration**

**\* ExpressionStatement** is equal to 2nd part of **OtherDeclarationOrStatement.**

**Don’t Confuse.**

Statement -> KeywordStatement | ExpressionStatement

ExpressionStatement -> **ID** RestOfExpressionStatement

**}**

KeywordStatement -> Selection | Iteration | Jump | Block

Selection -> **if** **(** Expression **)** Statement ~~OptionalElse~~

~~OptionalElse ->~~ **~~else~~** ~~Statement | ε~~

Iteration -> **while** **(**Expression**)**Statement | **do** Statement **while** **(**Expression**);**

Jump -> **return** OptionalExpression **;**

OtherDeclarationOrStatement -> **ID** RestOfDeclarationOrStatement

{

}

RestOfDeclarationOrStatement-> RestOfVariableDeclaration | RestOfExpressionStatement

RestOfExpressionStatement -> MorePrimary OptionalAssignment ';'

OptionalAssignment -> **=** Expression | ε

------------------------------ - Statement -------------------------