***Program*** 🡪 **int main () { Declarations Statements }**

***Declarations*** 🡪 **{ Declaration }**

***Declaration*** 🡪 **Type Identifier [ [ Integer ] ] { , Identifier [ [ Integer ] ] }**

***Type*** 🡪 **int | bool | float | char**

***Statements*** 🡪 **{ Statement }**

***Statement*** 🡪 **; | Block | Assignment | IfStatement | WhileStatement**

***Block*** 🡪 **{ Statements }**

***Assignment*** 🡪 **Identifier [ [ Expression ] ] = Expression;**

***IfStatement*** 🡪 **if ( Expression ) Statement [ else Statement ]**

***WhileStatement*** 🡪 **while ( Expression ) Statement**

***Expression*** 🡪 **Conjunction { || Conjunction }**

***Conjunction*** 🡪 **Equality { && Equality }**

***Equality*** 🡪 **Relation [ EquOp Relation ]**

***EquOp*** 🡪 **== | !=**

***Relation*** 🡪 **Addition [ RelOp Addition]**

***RelOp*** 🡪 **< | <= | > | >=**

***Addition*** 🡪**Term { AddOp Term }**

***AddOp*** 🡪 **+ | -**

***Term*** 🡪 **Factor { MulOp Factor }**

***MulOp*** 🡪 **\* | / | %**

***Factor*** 🡪 **[ UnaryOp ] Primary**

***UnaryOP*** 🡪 **- | !**

***Primary*** 🡪 **Identifier [ [Expression] ] | Literal | ( Expression ) | Type ( Expression)**

***Identifier*** 🡪 **Letter { Letter | Digit }**

***Letter* 🡪 a | b | … | z | A | B | … | Z**

***Digit*** 🡪 **0 | 1 | … | 9**

***Literal*** 🡪 **Integer | Boolean | Float | Char**

***Integer*** 🡪 **Digit { Digit }**

***Boolean* 🡪** **true | false**

***Float*** 🡪 **Integer.Integer**

***Char*** 🡪 **' ASCIIChar '**

**Note : The Grammar is in EBNF form**