----------Starting----------

<Start> → <class><defs> int<Start'> main(){<MST><ret>}<defs>

| <enum><defs> int<Start'> main(){<MST><ret>}<defs>

| static<st><defs> int<Start'> main(){<MST><ret>}<defs>

| virtual<vi><defs> int<Start'> main(){<MST><ret>}<defs>

| void ID(<argu>)<void_dec><defs> int<Start'> main(){<MST><ret>}<defs>

| const DT ID <defs1><defs> int<Start'> main(){<MST><ret>}<defs>

| int<Start'> main(){<MST><ret>}<defs>

| ID ID<defs3><defs> int<Start'> main(){<MST><ret>}<defs>

| <DT_ot> ID<defs2><defs> int<Start'> main(){<MST><ret>}<defs>

<Start'> → ID<defs2><defs><Start''> | NULL

<Start''> → int <Start'>

<defs> → <class><defs>

| <enum><defs>

| static<st><defs>

| virtual<vi><defs>

| void ID(<argu>)<void_dec><defs>

| const DT ID <defs1><defs>

| <DT_ot> ID<defs2><defs>

| ID ID<defs3><defs>

| NULL

<DT_ot> → float | string | char | bool

----------UPDATED SST----------

<MST> → <SST><MST> | NULL

<SST> → <while>
| <if>
| <for>
| <do_while>
| <enum>
| inc/dec <t>ID <d'><I_A> <other_inc_dec>;
| this.ID <d'> <I_A> <SST_th>
| const DT ID <SST_Arr_Dec>
| DT ID <SST2>
| ID<SST3>

<SST_th> → inc/dec <other_inc_dec>;
| <AO> <OE>;

<SST_Arr_Dec> → = <OE><List>
| [<A1>]<Dim> = {<OE><A>}<A2>

<SST2> → <init><List> | [<SIZE>

<SST3> → <d'><SST4>
| ID<SST5>

<SST4> → (<param>)<SST4_alpha>

<SST4_alpha> → ; | .ID <I_A> <SST_th>

<SST5> → <PC><other_obj>

| [<A1>]<Dim><A3>

----------LEFT FACTORING----------

<defs1> → = <OE><List> | [<A1>]<Dim> = {<OE><A>}<A2>

<defs2> → [<SIZE> | <init><List> | (<argu>)<dec>

<defs3> → (<defs3_arg> | [<A1>]<Dim><A3> | <other_obj>;

<defs3_arg> → ID<art>

| <CT>DT ID<Arr><argu1>)<dec>

|)<dec>

| <const><A>)<other_obj>

| (<OE>)<A>)<other_obj>

| !<F><A>)<other_obj>

| inc/dec ID<A>)<other_obj>

<art> → ID<Arr><argu1>)<dec> | <dot><A>)<other_obj>

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

<OE> → <AE><OE'>

<OE'> → || <AE><OE'> | NULL

<AE> ⟶ <RE><AE'>

<AE'> ⟶ && <RE><AE'> | NULL

<RE> ⟶ <E><RE'>

<RE'> ⟶ RO <E><RE'> | NULL

<E> ⟶ <T><E'>

<E'> ⟶ PM <T><E'> | NULL

<T> ⟶ <F><T'>

<T'> ⟶ MDM <F><T'> | NULL

<F> ⟶ ID<dot> | <const> | (<OE>) | !<F> | inc/dec ID

<dot> ⟶ .ID<dot> | (<param>).ID<dot> | [<OE>]<Dim>.ID<dot> | inc/dec | NULL

<Dim> ⟶ [<OE>] | NULL

<param> ⟶ <OE><par> | NULL

<par> ⟶ ,<OE><par> | NULL

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

<Dec> ⟶ const DT ID = <OE><List> | DT ID <init><List>

<init> ⟶ = <OE> | NULL

<List> ⟶ , ID <init><List> | ;

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

<vi> ⟶ void ID(<argu>)<vi1> | DT ID(<argu>)<vi2> | ID ID(<argu>)<vi2>

<vi1> ⟶ = 0; | ; | {<MST>}

<vi2> ⟶ = 0; | ; | {<MST><ret>}

<st> ⟶ void ID(<argu>)<void_dec> | DT ID(<argu>)<dec> | ID ID(<argu>)<dec>

<void_dec> ⟶ ; | {<MST>}

<dec> ⟶ ; | {<MST><ret>}

<ret> ⟶ return <OE>;

<argu> ⟶ <CT>DT ID<Arr><argu1> | ID ID<Arr><argu1> | NULL

<CT> ⟶ const | NULL

<argu1> ⟶ , <argu2> | NULL

<argu2> ⟶ DT ID<Arr><argu1> | ID ID<Arr><argu1>

<Arr> ⟶ [<A1>]<dim> | NULL

<dim> ⟶ [<A1>] | NULL

<A1> ⟶ <OE> | NULL

----------Jump----------

<jump> ⟶ Jump statements; | NULL

----------While----------

<while> ⟶ while (<OE>)<Body_wh>

<Body_wh> ⟶ ; | {<MST><jump>}

----------Func Call----------

<d'> ⟶ .ID<d'> | [<OE>]<Dim>.ID<d'> | NULL

<Dim> ⟶ [<OE>] | NULL

<param> ⟶ <OE><par> | NULL

<par> ⟶ ,<OE><par> | NULL

-------Constructor-----

<con'> ⟶ ; | {<MST>} | NULL

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

<class> ⟶ class ID<seal>;

<seal> ⟶ final {<Body>} | : Access_Modifier ID <class'> {<Body>} | {<Body>} | NULL

<class'> ⟶ , Access_Modifier ID <class'> | NULL

<Body> ⟶ <class><Body>
| <enum><Body>
| static<st><Body>
| virtual<vi><Body>
| void ID(<argu>)<void_dec><Body>
| const DT ID <defs1><Body>
| DT ID<defs2><Body>
| ID<Bd><Body>
| ~ID()<con'><Body>
| Access_Modifier : <Body>
| NULL

<Bd> ⟶ ID<defs3> | (<argu>)<con'>

----------Object----------

<PC> ⟶ (<OE><A>) | NULL

<other_obj> ⟶ ; | ,ID<PC><other_obj>

----------enum-----

<enum> ⟶ enum ID{<values>};

<values> ⟶ ID<val> | NULL

<val> ⟶ ,ID<val> | =<const><val'> | NULL

<val'> ⟶ ,ID<val> | NULL

----------inc/dec----

<t> ⟶ this. | NULL

<inc_dec> ⟶ <t>ID <d'><I_A> inc/dec <other_inc_dec>; | inc/dec <t>ID <I_A> <other_inc_dec>;

<I_A> ⟶ (<param>).ID <I_A> | NULL

<other_inc_dec> ⟶ ,<inc_dec> | NULL

----------For------

<for> ⟶ for(<F1> <F2> ; <F3>) <for'>

<F1> ⟶ <Dec> | <assignment>

<F2> ⟶ <OE> | NULL

<F3> ⟶ this.ID <d'><I_A> <SST_th>
| ID <d'><I_A> <SST_th>
| inc/dec <t>ID <I_A> <other_inc_dec>;
| NULL

<for'> → ; | {<MST><jump>}

----------Do While----------

<do_while> → do{<MST><jump>}while(<OE>);

----------If---------

<if> → if(<OE>)<if'><else>

<if'> → ; | {<MST><jump>}

<else> → else<if'> | NULL

----------Array------

<A> → ,<OE><A> | NULL

<SIZE> → <OE>]<Dim> <A7> |]<Dim> = {<OE><A>}<A8>

<A2> → ; | ,ID[<A1>] = {<OE><A>}<A2>

<A3> → ; | = {<A4>}

<A4> → ID<A5><A6>

<A5> → (<param>) | [<OE>] | NULL

<A6> → ,ID<A5> | NULL

<A7> → ; | = {<OE><A>}<A2>

<A8> → ; | ,ID[<SIZE>

----------Assignment-------

<assignment> → <t> ID <d'><I_A> <AO> <OE>;

<AO> → = | PMMDM