




-----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>  $\longrightarrow$  <SST><MST> | NULL

<SST>  $\longrightarrow$  <while>  
| <if>  
| <for>  
| <do\_while>  
| <enum>  
| inc/dec <t>ID <d'><l\_A> <other\_inc\_dec>;  
| this.ID <d'> <l\_A> <SST\_th>  
| const DT ID <SST\_Arr\_Dec>  
| DT ID <SST2>  
| ID<SST3>

<SST\_th>  $\longrightarrow$  inc/dec <other\_inc\_dec>;  
| <AO> <OE>;

<SST\_Arr\_Dec>  $\longrightarrow$  = <OE><List>  
| [<A1>]<Dim> = {<OE><A>}<A2>

<SST2>  $\longrightarrow$  <init><List> | [<SIZE>

<SST3>  $\longrightarrow$  <d'><SST4>  
| ID<SST5>

<SST4>  $\longrightarrow$  (<param>)<SST4\_alpha>

<SST4\_alpha>  $\longrightarrow$  ; | .ID <I\_A> <SST\_th>

<SST5>  $\longrightarrow$  <PC><other\_obj>  
| [<A1>]<Dim><A3>

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

<defs1>  $\longrightarrow$  = <OE><List> | [<A1>]<Dim> = {<OE><A>}<A2>

<defs2>  $\longrightarrow$  [<SIZE> | <init><List> | (<argu>)<dec>

<defs3>  $\longrightarrow$  (<defs3\_arg> | [<A1>]<Dim><A3> | <other\_obj>;

<defs3\_arg>  $\longrightarrow$  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>  $\longrightarrow$  ID<Arr><argu1>)<dec> | <dot><A>)<other\_obj>

-----Expression-----

<OE>  $\longrightarrow$  <AE><OE'>

<OE'>  $\longrightarrow$  | | <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>  $\longrightarrow$  (<OE><A>) | NULL

<other\_obj>  $\longrightarrow$  ; | ,ID<PC><other\_obj>

-----enum-----

<enum>  $\longrightarrow$  enum ID{<values>;

<values>  $\longrightarrow$  ID<val> | NULL

<val>  $\longrightarrow$  ,ID<val> | =<const><val'> | NULL

<val'>  $\longrightarrow$  ,ID<val> | NULL

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

<t>  $\longrightarrow$  this. | NULL

<inc\_dec>  $\longrightarrow$  <t>ID <d'><l\_A> inc/dec <other\_inc\_dec>; | inc/dec <t>ID <l\_A> <other\_inc\_dec>;

<l\_A>  $\longrightarrow$  (<param>).ID <l\_A> | NULL

<other\_inc\_dec>  $\longrightarrow$  ,<inc\_dec> | NULL

-----For-----

<for>  $\longrightarrow$  for(<F1> <F2> ; <F3>) <for'>

<F1>  $\longrightarrow$  <Dec> | <assignment>

<F2>  $\longrightarrow$  <OE> | NULL

<F3>  $\longrightarrow$  this.ID <d'><l\_A> <SST\_th>  
| ID <d'><l\_A> <SST\_th>  
| inc/dec <t>ID <l\_A> <other\_inc\_dec>;  
| NULL

<for'>  $\longrightarrow$  ; | {<MST><jump>}

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

<do\_while>  $\longrightarrow$  do{<MST><jump>}while(<OE>);

-----If-----

<if>  $\longrightarrow$  if(<OE>)<if'><else>

<if'>  $\longrightarrow$  ; | {<MST><jump>}

<else>  $\longrightarrow$  else<if'> | NULL

-----Array-----

<A>  $\longrightarrow$  ,<OE><A> | NULL

<SIZE>  $\longrightarrow$  <OE>]<Dim> <A7> | ]<Dim> = {<OE><A>}<A8>

<A2>  $\longrightarrow$  ; | ,ID[<A1>] = {<OE><A>}<A2>

<A3>  $\longrightarrow$  ; | = {<A4>}

<A4>  $\longrightarrow$  ID<A5><A6>

<A5>  $\longrightarrow$  (<param>) | [<OE>] | NULL

<A6>  $\longrightarrow$  ,ID<A5> | NULL


<A7>  $\longrightarrow$  ; | = {<OE><A>}<A2>

<A8>  $\longrightarrow$  ; | ,ID[<SIZE>



-----Assignment-----

<assignment>  <t> ID <d'><l\_A> <AO> <OE>;

<AO>  = | PMMDM