<structure></structure>	> static class ID <inherit> { <s_cst></s_cst></inherit>	<s_main></s_main>	> (<pl>) <bodymst> <s_cstnm></s_cstnm></bodymst></pl>
	> condensed class ID <inherit> { <c_cst></c_cst></inherit>	<s_cstnm></s_cstnm>	> } <main_done></main_done>
	> <concrete'> class ID <inherit> { <g_cst></g_cst></inherit></concrete'>		> function <pubpriv'> static <function_sig> <s_cstnm></s_cstnm></function_sig></pubpriv'>
	> symbol ID { <int_st> } <structure></structure></int_st>		> private static <class_vars> <s_cstnm></s_cstnm></class_vars>
<eof></eof>	> ~		> <public'> static <consvar> <s_cstnm></s_cstnm></consvar></public'>
<s_cst></s_cst>	> } <structure></structure>	<consvar></consvar>	> DT <n_dec></n_dec>
	> function <pubpriv'> static <function_sig> <s_cst></s_cst></function_sig></pubpriv'>		> ID <consvar'></consvar'>
	> private static <class_vars> <s_cst></s_cst></class_vars>		> dict <multi_arr></multi_arr>
	> <public'> static <s_cst'></s_cst'></public'>	<consvar'></consvar'>	> (<al>) <body_mst></body_mst></al>
<s_cst'></s_cst'>	> DT <s_cstdt></s_cstdt>		> ID <object'></object'>
	> ID <s_cstid></s_cstid>		> [] <o_arr'></o_arr'>
	> dict <multi_arr> <s_cst></s_cst></multi_arr>	<c_cst></c_cst>	> } <structure></structure>
	> void Main <s_main></s_main>		> <privpres> <sc> <class_vars> <c_cst></c_cst></class_vars></sc></privpres>
<s_cstdt></s_cstdt>	> ID <declare'> <s_cst></s_cst></declare'>		> function <am> <types> <c_cst></c_cst></types></am>
	> [] <s_cstdt'></s_cstdt'>		> <class_vars> <c_cst></c_cst></class_vars>
	> Main <s_main></s_main>		> <public'> <c_cst'></c_cst'></public'>
<s_cstdt'></s_cstdt'>	> ID <g_arr'> <s_cst></s_cst></g_arr'>	<c_cst'></c_cst'>	> static <c_cst"></c_cst">
	> Main <s_main></s_main>		> concrete <class_vars> <c_cst></c_cst></class_vars>
<s_cstid></s_cstid>	> ID <object'> <s_cst></s_cst></object'>	<c_cst"></c_cst">	> DT <c_cstdt></c_cstdt>
	> [] <s_cstid'></s_cstid'>		> ID <c_cstid></c_cstid>
	> Main <s_main></s_main>		> dict <multi_arr> <s_cst></s_cst></multi_arr>
	> () <body_mst> <s_cst></s_cst></body_mst>		> void Main <gc_main></gc_main>
<s_cstid'></s_cstid'>	> ID <o_arr'> <s_cst></s_cst></o_arr'>	<c_cstdt></c_cstdt>	> ID <declare'> <c_cst></c_cst></declare'>
	> Main <s_main></s_main>		> [] <c_cstdt'></c_cstdt'>

	> Main <gc_main></gc_main>		> dict <multi_arr> <s_cst></s_cst></multi_arr>
<c_cstdt'></c_cstdt'>	> ID <g_arr'> <c_cst></c_cst></g_arr'>		> void Main <gc_main></gc_main>
	> Main <gc_main></gc_main>	<g_cstdt></g_cstdt>	> ID <declare'> <g_cst></g_cst></declare'>
<c_cstid></c_cstid>	> ID <object'> <c_cst></c_cst></object'>		> [] <g_cstdt'></g_cstdt'>
	> [] <c_cstid'></c_cstid'>		> Main <gc_main></gc_main>
	> Main <gc_main></gc_main>	<g_cstdt'></g_cstdt'>	> ID <g_arr'> <g_cst></g_cst></g_arr'>
	> (<al>) <body_mst> <c_cst></c_cst></body_mst></al>		> Main <gc_main></gc_main>
<c_cstid'></c_cstid'>	> ID <o_arr'> <c_cst></c_cst></o_arr'>	<g_cstid></g_cstid>	> ID <object'> <g_cst></g_cst></object'>
	> Main <gc_main></gc_main>		> [] <g_cstid'></g_cstid'>
<gc_main></gc_main>	> (<al>) <bodymst> <gc_nst></gc_nst></bodymst></al>		> Main <gc_main></gc_main>
<gc_nst></gc_nst>	> } <main_done></main_done>		> (<al>) <body_mst> <g_cst></g_cst></body_mst></al>
	> function <am> <types> <gc_nst></gc_nst></types></am>	<g_cstid'></g_cstid'>	> ID <o_arr'> <g_cst></g_cst></o_arr'>
	> <prespriv> <sc> <class_vars> <gc_nst></gc_nst></class_vars></sc></prespriv>		> Main <gc_main></gc_main>
	> <public'> <gc_consvar> <gc_nst></gc_nst></gc_consvar></public'>	<main_done></main_done>	>\$
<gc_consvar< td=""><td>>> concrete <class_vars></class_vars></td><td></td><td>> static class ID <inherit> { <s_cstnm></s_cstnm></inherit></td></gc_consvar<>	>> concrete <class_vars></class_vars>		> static class ID <inherit> { <s_cstnm></s_cstnm></inherit>
	> <static'> <consvar></consvar></static'>		> <conccond'> class ID <inherit> { <gc_nst></gc_nst></inherit></conccond'>
<g_cst></g_cst>	> } <structure></structure>		> symbol ID { <int_st> } <main_done></main_done></int_st>
	> <privpres> <sc> <class_vars> <g_cst></g_cst></class_vars></sc></privpres>	<assignop></assignop>	> =
	> function <am> <types> <g_cst></g_cst></types></am>		> CompAssign
	> <public'> <g_cst'></g_cst'></public'>	<ref></ref>	> . ID <ref></ref>
<g_cst'></g_cst'>	> concrete <class_vars> <g_cst></g_cst></class_vars>		> [<exp>] <ref></ref></exp>
	> <static'> <g_cst"></g_cst"></static'>		> (<pl>) . ID <ref></ref></pl>
<g_cst"></g_cst">	> DT <g_stdt></g_stdt>		>\$
	> ID <g_stid></g_stid>		

```
<SST>
               --> <For_St>
                                                                                                      -->;
               --> <lf St>
                                                                                       <SST 2>
                                                                                                      -->;
               --> <While_St>
                                                                                                      --> . ID <SST_ID'>
               --> <DoWhile St>
                                                                                       <Break>
                                                                                                      --> Break ;
                                                                                                      --> Continue ;
               --> <Return St>
                                                                                       <Continue>
                                                                                       <While_St>
                                                                                                      --> While ( <Exp> ) <Body>
               --> <Continue>
               --> <Break>
                                                                                       <Body>
                                                                                                      -->;
                                                                                                      --> <BodyMST>
               --> <Try St>
               --> inc_dec <SP'> ID <Ref>;
                                                                                       <DoWhile_St> --> do <BodyMST> while ( <Exp> )
               --> throw <Throw'>
                                                                                       <Try_St>
                                                                                                      --> try <BodyMST> <CatchFinally>
               --> <SP> ID <Ref> <AssignOP> <Exp>;
                                                                                       <CatchFinally> --> <Finally>
                                                                                                      --> <Catch> <Finally'>
               --> DT <N_Dec>
                                                                                                      --> finally <BodyMST>
               --> dict <Multi Arr>
                                                                                       <Finally>
               --> ID <SST_ID>
                                                                                       <Finally'>
                                                                                                      --> finally <BodyMST>
<SST_ID>
               --> ID <Object'>
                                                                                                      --> $
                                                                                                      --> catch ( <Exception> ID ) <BodyMST> <Catch'>
               --> <SST ID'>
                                                                                       <Catch>
<SST_ID'>
               --> . ID <SST_ID'>
                                                                                       <Catch'>
                                                                                                      --> <Catch>
                                                                                                      -->$
               --> [ <Exp> ] <SST 1>
                                                                                       <Swicth_St>
                                                                                                      --> swicth ( ID ) { <Swicth_Body> }
               --> ( <PL> ) <SST_2>
                                                                                       <Swicth_Body> --> <Case> <Default>
               --> inc_dec ;
               --> <AssignOp> <Exp>;
                                                                                                      --> case ( <Const> ) : <MST> <Case>
                                                                                       <Case>
<SST_1>
               --> . ID <SST ID'>
                                                                                                      -->$
               --> inc_dec;
                                                                                       <Default>
                                                                                                      --> default : <MST> <Case>
               --> <AssignOp> <Exp>;
                                                                                                      -->$
```

> <bodymst></bodymst>	<if_st></if_st>	> if (<exp>) <bodymst> <oelse></oelse></bodymst></exp>		> [<exp>] <for_opt'></for_opt'></exp>
COEIse'> → if (<exp>) <bodymst> <oeise> → <assignop> <exp> <for_opt'> → > For_St> → For (<st1> <st2> ; <st3>) <body> ←For_Opt'> → ID <for_opt> <5t1> → < Dec> → <assignop> <exp> <for_opt"> → <assignop> <exp> <for_opt"> <5t1> → < CDec> ←For_Opt"> → <assignop> <exp> <for_opt"> <0ec> → DT ID <declare'> <inherit> → <expands> <applies> <3xsignSt> → SP'> ID <ref> <assignop> <exp>; <expands> → Expands ID <5t2> → \$ → > Expands ID → > Expands ID <5t2> → \$ → > Applies ID <applies> → > Applies ID <applies> <5t3> → inc_dec <var> → \$ → > ID <applies'> → > ID <applies'> <5t3> → ID <cond'> → > \$ → > ID <applies'> → > \$ <cond> → ID <cond'> → > Concrete'> → Concrete → > Concrete → > Public'> <eetby< td=""> → ROR <exp> <</exp></eetby<></cond'></cond></applies'></cond'></applies'></applies'></var></applies></applies></expands></exp></assignop></ref></applies></expands></inherit></declare'></for_opt"></exp></assignop></for_opt"></exp></assignop></for_opt"></exp></assignop></for_opt></body></st3></st2></st1></for_opt'></exp></assignop></oeise></bodymst></exp>	<oelse></oelse>	> else <oelse'></oelse'>		> (<pl>) . ID <for_opt></for_opt></pl>
-> <bodymst> <pre> <pre< td=""><td></td><td>>\$</td><td></td><td>> inc_dec <for_opt"></for_opt"></td></pre<></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></bodymst>		>\$		> inc_dec <for_opt"></for_opt">
<for_st> -> For (<st1> <st2>; <st3>) <body> -> inc_dec <for_opt"> <st1> -> <dec> -> <assignop> <exp> <for_opt"> -> <assignst> <for_opt"> -> \$ <->; -> , ID <for_opt> -> , ID <for_opt> <dec> -> DT ID <declare'> <inherit> -> <expands> <applies> <assignst> -> <sp'> ID <ref> <assignop> <exp>; <expands> -> Expands ID <st2> -> \$ -> \$ -> \$ <st3> -> inc_dec <var> -> D <for_opt> -> \$ -> , ID <applies'> <-> ID <for_opt> <applies> -> , ID <applies'> -> , ID <applies'> <-> DI <for_opt> <applies'> -> , ID <applies'> -> , ID <applies'> <-> \$ -> DI <cond'> <concrete'> -> Concrete' -> Concrete <-> <0ont> <-> Const> <-> \$ <int_st> -> <public'> <-> \$ <-> <0ont> <-> <-> <-> <-> <-> <-> <-> <-> <-> <-</public'></int_st></concrete'></cond'></applies'></applies'></applies'></for_opt></applies'></applies'></applies></for_opt></applies'></for_opt></var></st3></st2></expands></exp></assignop></ref></sp'></assignst></applies></expands></inherit></declare'></dec></for_opt></for_opt></for_opt"></assignst></for_opt"></exp></assignop></dec></st1></for_opt"></body></st3></st2></st1></for_st>	<oelse'></oelse'>	> if (<exp>) <bodymst> <oelse></oelse></bodymst></exp>		> <assignop> <exp> <for_opt"></for_opt"></exp></assignop>
SST1> -> <dec> -> <assignop> <exp> <for_opt"> -> <assignst> <for_opt"> -> \$ -> ; -> ,ID <for_opt> <dec> -> DT ID <declare'> <inherit> -> <expands> <applies> <assignst> -> <sp'> ID <ref> <assignop> <exp>; <expands> -> Expands ID <st2> -> \$ -> \$ -> \$ <ncond> <applies> -> Applies ID <applies'> -> ID <for_opt> <applies> -> ,ID <applies'> <ncond> -> ID <cond'> <applies'> -> Concrete <ncond'> -> \$ <int_st> -> Public'> <int_st> <return> -> return <ret'>; <public'> -> Public -> \$ <ret'> -> <exp> <rt'> -> []</rt'></exp></ret'></public'></ret'></return></int_st></int_st></ncond'></applies'></cond'></ncond></applies'></applies></for_opt></applies'></applies></ncond></st2></expands></exp></assignop></ref></sp'></assignst></applies></expands></inherit></declare'></dec></for_opt></for_opt"></assignst></for_opt"></exp></assignop></dec>		> <bodymst></bodymst>	<for_opt'></for_opt'>	> . ID <for_opt></for_opt>
-> <assignst></assignst>	<for_st></for_st>	> For (<st1> <st2> ; <st3>) <body></body></st3></st2></st1>		> inc_dec <for_opt"></for_opt">
->; <dec> -> DT ID <declare'> <assignst> -> <\$P'> ID <ref> <assignop> <exp>; -> \$ -> <\$ -> <applies> -> \$ -> <applies <applies'="" id=""> -> \$ -> <applies <applies'="" id=""> -> \$ -> <applies'> -> Applies ID <applies'> -> \$ -> ID <for_opt> -> \$ -> ID <for_opt> -> \$ -> ID <for_opt> -> \$ <cond> -> \$ <cond> -> \$ -> ID <cond'> -> \$ -> Concrete'> -> <concrete' -=""> <concrete' -=""> <cond> -> \$ <cond'> -> \$ -> IN _ST = S = S = S = S = S = S = S = S = S =</cond'></cond></concrete'></concrete'></cond'></cond></cond></for_opt></for_opt></for_opt></applies'></applies'></applies></applies></applies></exp></assignop></ref></assignst></declare'></dec>	<st1></st1>	> <dec></dec>		> <assignop> <exp> <for_opt"></for_opt"></exp></assignop>
CDec> -> DT ID < Declare'> <inherit> -> < Expands> < Applies> <assignst> -> \$P'> ID < Ref> < AssignOP> < Exp>; < Expands> -> Expands ID <st2> -> \$ -> \$ -> \$ <st3> -> inc_dec < Var> -> inc_dec < Var> <-> D < For_Opt> < Applies> -> \$ < Cond> -> ID < Cond'> < Concrete'> -> Concrete < Cond> -> ID < Cond'> < Concrete'> -> Concrete < Cond'> -> \$ < Int_St> -> Public'> < RT> ID (< AL>) {}; < Return> -> return < Ret'>; < Public'> -> Public -> Public < Ret'> -> \$ < RT'> -> []</st3></st2></assignst></inherit>		> <assignst></assignst>	<for_opt"></for_opt">	>\$
<assignst> -> <sp'> ID <ref> <assignop> <exp>; <expands> -> Expands ID <st2> -> \$ > \$ > \$ -> <cond> <applies> -> Applies ID <applies'> <>St3> -> inc_dec <var> > iD <for_opt> <applies'> > ,ID <applies'> <-> \$ > ID <cond'> > Concrete'> > Concrete <</cond'></applies'></applies'></for_opt></var></applies'></applies></cond></st2></expands></exp></assignop></ref></sp'></assignst>		>;		> , ID <for_opt></for_opt>
<5t2> > \$ > \$ > <applies> > Applies ID <applies'> <5t3> > inc_dec <var> > \$ > \$ > ID <for_opt> <applies'> > \$ <cond> > ID <cond'> > \$ > Concrete' > Concrete <cond> > \$ <int_st> > Public'> <int_st> ID (<al>) {}; <return> > ROR <exp> <public'> > Public <ret'> > \$ <int_st> > Public <> \$ <int_st< td=""> > Public <> \$ <int_st< td=""> > \$ <> \$ <int_st< td=""> > Public <> \$ <int_st< td=""> > \$ <> \$ <int_st< td=""> <td< td=""><td><dec></dec></td><td>> DT ID <declare'></declare'></td><td><inherit></inherit></td><td>> <expands> <applies></applies></expands></td></td<></int_st<></int_st<></int_st<></int_st<></int_st<></int_st></int_st></int_st></int_st></int_st></ret'></int_st></ret'></int_st></ret'></int_st></ret'></public'></exp></return></al></int_st></int_st></cond></cond'></cond></applies'></for_opt></var></applies'></applies>	<dec></dec>	> DT ID <declare'></declare'>	<inherit></inherit>	> <expands> <applies></applies></expands>
-> <cond></cond>	<assignst></assignst>	> <sp'> ID <ref> <assignop> <exp> ;</exp></assignop></ref></sp'>	<expands></expands>	> Expands ID
<5t3> > inc_dec <var> > \$ > \$ <-> ID <for_opt> <applies'> > \$ <-> \$ > \$ > \$ <cond> > ID <cond'> > Concrete'> > Concrete > <const> <cond'> > \$ <int_st> > Public'> <rt> ID (<al>) { } ; <cond'> > ROR <exp> > \$ > \$ <return> > return <ret'>; <-> \$ > \$ <ret'> > \$ > \$ > \$ <-> <</ret'></ret'></return></exp></cond'></al></rt></int_st></cond'></const></cond'></cond></applies'></for_opt></var>	<st2></st2>	> \$		>\$
-> ID <for_opt> -> \$> \$> Cond>> ID <cond'>> Const> <cond'>> \$> \$> Concrete'>> Concrete'> Cond'>> \$> ROR <exp>> return <ret'>;> \$> \$</ret'></exp></cond'></cond'></for_opt>		> <cond></cond>	<applies></applies>	> Applies ID <applies'></applies'>
->\$ ->\$ ->\$ ->\$ ->\$ ->\$ ->\$ ->\$ ->\$ ->\$	<st3></st3>	> inc_dec <var></var>		>\$
<cond> > ID <cond'> > Concrete > Concrete > <const> <cond'> > \$ <int_st> > <public'> <rt> ID (<al>) {}; <cond'> > ROR <exp> > \$ > \$ <return> > return <ret'>; <public'> > Public <ret'> > \$ > \$ > <exp> <rt'> > []</rt'></exp></ret'></public'></ret'></return></exp></cond'></al></rt></public'></int_st></cond'></const></cond'></cond>		> ID <for_opt></for_opt>	<applies'></applies'>	> , ID <applies'></applies'>
-> <cond'> -> \$ <cond'>> \$ > ROR <exp>> return <ret'>; > \$ > \$ > \$ > Public'> <rt> ID (<al>){}; > Public'>> \$ > Public'>> Public > \$ </al></rt></ret'></exp></cond'></cond'>		> \$		>\$
<cond'> > \$ > <public'> <rt> ID (<al>) {}; <return> > return <ret'>; <public'> > Public <ret'> > \$ > \$ > <exp> <rt'> > []</rt'></exp></ret'></public'></ret'></return></al></rt></public'></cond'>	<cond></cond>	> ID <cond'></cond'>	<concrete'></concrete'>	> Concrete
> ROR <exp>> \$ <return>> return <ret'>;</ret'></return></exp>		> <const> <cond'></cond'></const>		>\$
<return>> return <ret'>;</ret'></return>	<cond'></cond'>	>\$	<int_st></int_st>	> <public'> <rt> ID (<al>) { } ;</al></rt></public'>
<ret'>> \$> []</ret'>		> ROR <exp></exp>		>\$
> <exp> <rt'>> []</rt'></exp>	<return></return>	> return <ret'> ;</ret'>	<public'></public'>	> Public
	<ret'></ret'>	> \$		>\$
<for_opt>> . ID <for_opt>> \$</for_opt></for_opt>		> <exp></exp>	<rt'></rt'>	->[]
	<for_opt></for_opt>	> . ID <for_opt></for_opt>		>\$

```
<RT>
               --> DT <RT'>
                                                                                        <Init_Multidim>--> new dict [ <Init_Multidim'> ]
               --> ID <RT'>
                                                                                                       --> ID
               --> Void
                                                                                        <Init_Multidim'>--> <Exp>]
               --> dict [ ]
                                                                                                       --> ] { <Val_Multidim> }
<AL>
               -->$
                                                                                        <Val_Multidim>--> ID <Val_Multidim'>
               --> <AL'>
                                                                                                       -->$
<AL'>
               --> ID <AL"'> ID <AL">
                                                                                        <Val Multidim'>-->$
               --> DT <AL"'> ID <AL">
                                                                                                       --> . ID
               --> dict [] <AL">
                                                                                        <N_Dec>
                                                                                                       --> ID <Declare'>
<Al''>
               -->$
                                                                                                       -->[] ID <G_Arr'>
               --> , <AL'>
                                                                                                       --> ID <Object'>
                                                                                        <O Dec>
<AL'''>
               -->$
                                                                                                       -->[] ID <O_Arr'>
               -->[]
                                                                                        <Declare'>
                                                                                                       -->;
<< PubPriv'>
               --> Public
                                                                                                       --> , ID <Declare'>
                                                                                                       --> = <Init_List>
               --> Private
               -->$
                                                                                        <G_Arr'>
                                                                                                       -->;
<Function_Sig> --> <RT> ID ( <AL> ) <Body_MST>
                                                                                                       --> , ID <G_Arr'>
<Class Vars> --> DT <N Dec>
                                                                                                       --> = <Init GArr> <G Arr'>
                                                                                       <Init_GArr>
               --> ID <O_Dec>
                                                                                                       --> ID
               --> dict <Multi_Arr>
                                                                                                       --> new DT [ <Init_GArr'>
<Multi_Arr>
               -->[] ID <<Multi_Arr'>
                                                                                        <Init_GArr'>
                                                                                                       --> <Exp> ]
<Multi Arr'> -->;
                                                                                                       --> ] { <Val_GArr> }
               -->, ID <Multi Arr'>
                                                                                       <Val_GArr>
                                                                                                       --> <Const> <Val_GArr'>
               --> = <Init_Multidim> <Multi_Arr'>
                                                                                                       -->$
```

<val_garr'></val_garr'>	> \$	<b'></b'>	> && <c> <b'></b'></c>
	> , <const> <val_garr'></val_garr'></const>		>\$
<init_list'></init_list'>	> ;	<c></c>	> <e> <c'></c'></e>
	> , ID <declare></declare>	<c'></c'>	> ROR <e> <c'></c'></e>
<init_list></init_list>	> <sp'> ID <list1></list1></sp'>		>\$
	> <const> <list2></list2></const>	<e></e>	> <t> <e'></e'></t>
	> (<exp>) <list2></list2></exp>	<e'></e'>	> PM <t> <e'></e'></t>
	>! <f> <init_list'></init_list'></f>		>\$
<list1></list1>	> = <init_list></init_list>	<t></t>	> <f> <t'></t'></f>
	> . ID <list1></list1>	<t'></t'>	> MDM <f> <t'></t'></f>
	> [<exp>] <list3></list3></exp>		>\$
	> (<pl>) <list2></list2></pl>	<f></f>	> (<exp>)</exp>
	> inc_dec <list2></list2>		> <const></const>
	> <list2></list2>		>! <f></f>
<list2></list2>	> <t'> <e'> <c'> <b'> <a> <init_list'></init_list'></b'></c'></e'></t'>		> <sp'> ID <optf></optf></sp'>
		<sp></sp>	> Self .
<list3></list3>	> = <init_list></init_list>		> Parent .
	> . ID <list1></list1>	<sp'></sp'>	> Self .
	> inc_dec <list2></list2>		> Parent .
	> <list2></list2>		>\$
<exp></exp>	> <a>	<sc></sc>	> Static
<a>	> <a>		> Concrete
	> \$		>\$
	> <c> <b'></b'></c>	<optf></optf>	> . ID <optf></optf>

	> [<exp>] <optf></optf></exp>		>\$
	> (<pl>) <optf2></optf2></pl>	<val_oarr'></val_oarr'>	> , new ID (<pl>) <val_oarr'></val_oarr'></pl>
	> inc_dec		>\$
	>\$	<pl></pl>	>\$
<optf1></optf1>	> inc_dec		> <exp> <pl'></pl'></exp>
	> . ID <optf></optf>	<pl'></pl'>	> , <exp> <pl'></pl'></exp>
<optf2></optf2>	> . ID <optf></optf>		>\$
	>\$	<function_sig></function_sig>	> <rt> ID (<al>) <bodymst></bodymst></al></rt>
<bodymst></bodymst>	> { <mst> }</mst>	<am></am>	> Public
<mst></mst>	> <sst> <mst></mst></sst>		> Private
	>\$		> Preserved
<object'></object'>	>;		>\$
	> , ID <object'></object'>	<types></types>	> Condensed <rt> ID (<al>);</al></rt>
	> = <init_obj> <object'></object'></init_obj>		> <sc> <function_sig></function_sig></sc>
<init_obj></init_obj>	> new ID (<pl>)</pl>	<prespriv></prespriv>	> Preserved
	> ID		> Private
<0_Arr'>	>;	<conccond'></conccond'>	> Concrete
	> , ID <o_arr'></o_arr'>		> Condensed
	> = <init_oarr> <o_arr'></o_arr'></init_oarr>		>\$
<init_oarr'></init_oarr'>	> ID	<const></const>	> Int_Const
	> new ID [<init_oarr'></init_oarr'>		> Float_Const
<init_oarr'></init_oarr'>	>] { <val_oarr> }</val_oarr>		> String_Const
	> <exp>]</exp>		> Char_Const
<val_oarr></val_oarr>	> new ID <pl>) <val_oarr'></val_oarr'></pl>		> Bool_Const