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

<START> Shape <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’> Shape ID<DEFS2><DEFS><START’'> | ∑

<START’'> Shape int <START’>

<DEFS> Shape <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>

| ∑

<DT\_OT> Shapefloat | string | char | bool

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

<MST> Shape <SST><MST> | ∑

<SST>Shape<WHILE>

| <IF>

| <for>

| <DO\_WHILE>

| <ENUM>

| <PRINT>

| <INPUT>

| inc/dec <THIS>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> Shapeinc/dec <OTHER\_INC\_DEC>;

| <AO> <OE>;

<SST\_ARR\_DEC> Shape = <OE><LIST>

| [<A1>]<DIM> = {<OE><A>}<A2>

<SST2> Shape <INIT><LIST> | [<SIZE>

<SST3>Shape<D’><SST4>

| ID<SST5>

<SST4>Shape (<PARAM>)<SST4\_ALPHA>

<SST4\_ALPHA> Shape ; | .ID <I\_A> <SST\_TH>

<SST5> Shape <PC><OTHER\_OBJ>

| [<A1>]<DIM><A3>

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

<DEFS1> Shape = <OE><LIST> | [<A1>]<DIM> = {<OE><A>}<A2>

<DEFS2> Shape [<SIZE> | <INIT><LIST> | (<ARGU>)<FUNC\_DEC>

<DEFS3> Shape (<DEFS3\_ARG> | [<A1>]<DIM><A3> | <OTHER\_OBJ>;

<DEFS3\_ARG>Shape ID<ART>

| <CT>DT ID<ARR><ARGU1>)<FUNC\_DEC>

| )<FUNC\_DEC>

| <const><A>)<OTHER\_OBJ>

| (<OE>)<A>)<OTHER\_OBJ>

| !<F><A>)<OTHER\_OBJ>

| inc/dec ID<A>)<OTHER\_OBJ>

<ART> Shape ID<ARR><ARGU1>)<FUNC\_DEC> | <DOT><A>)<OTHER\_OBJ>

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

<OE> Shape <AE><OE'>

<OE'> Shape || <AE><OE'> | ∑

<AE> Shape <RE><AE'>

<AE'>Shape&& <RE><AE'> | ∑

<RE>Shape<E><RE'>

<RE'>ShapeRO <E><RE'> | ∑

<E>Shape<T><E'>

<E'> Shape PM <T><E'> | ∑

<T> Shape <F><T'>

<T'> Shape MDM <F><T'> | ∑

<F> Shape ID<DOT> | <const> | (<OE>) | !<F> | inc/dec ID

<DOT> Shape .ID<DOT> | (<PARAM>).ID<DOT> | [<OE>]<DIM>.ID<DOT> | inc/dec | ∑

<DIM> Shape [<OE>] | ∑

<PARAM> Shape <OE><PAR> | ∑

<PAR>Shape ,<OE><PAR> | ∑

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

<DEC> Shape const DT ID = <OE><LIST> | DT ID <INIT><LIST>

<INIT> Shape = <OE> | ∑

<LIST> Shape , ID <INIT><LIST> | ;

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

<VI> Shape void ID(<ARGU>)<VI1> | DT ID(<ARGU >)<VI2> | ID ID(<ARGU>)<VI2>

<VI1> Shape = 0; | ; | {<MST>}

<VI2> Shape = 0; | ; | {<MST><RET>}

<ST> Shape void ID(<ARGU>)<VOID\_DEC> | DT ID(<ARGU>)< FUNC\_DEC> | ID ID(<ARGU>)<DEC>

<VOID\_DEC> Shape ; | {<MST>}

<FUNC\_DEC>Shape; | {<MST><RET>}

<RET>Shapereturn <OE>;

< ARGU > Shape <CT>DT ID<ARR><ARGU1> | ID ID<ARR><ARGU1> | ∑

<CT> Shape const | ∑

<ARGU1>Shape, <ARGU2> | ∑

<ARGU2> Shape DT ID<ARR><ARGU1> | ID ID<ARR><ARGU1>

<ARR> Shape [<A1>]<ARR\_DIM> | ∑

<ARR\_DIM> Shape [<A1>] | ∑

<A1> Shape <OE> | ∑

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

<ELSE> Shape Jump statements; | ∑

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

<WHILE> Shape while (<OE>)<BODY\_WH>

<BODY\_WH> Shape ; | {<MST><ELSE>}

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

<D’> Shape .ID<D’> | [<OE>]<DIM>.ID<D’> | ∑

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

<CON'> Shape ; | {<MST>} | ∑

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

<CLASS> Shape class ID<SEAL>;

<SEAL> Shape final {<BODY>} | : Access\_Modifier ID <CLASS'> {<BODY>} | {<BODY>} | ∑

<CLASS'> Shape , Access\_Modifier ID <CLASS'> | ∑

<BODY> Shape <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>

| ∑

<BD> Shape ID<DEFS3> | (<ARGU>)<CON'>

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

<PC> Shape (<OE><A>) | ∑

<OTHER\_OBJ> Shape ; | ,ID<PC><OTHER\_OBJ>

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

<ENUM> Shape enum ID{<VALUES>};

<VALUES> Shape ID<VAL> | ∑

<VAL> Shape ,ID<VAL> | =<const><VAL'> | ∑

<VAL'> Shape ,ID<VAL> | ∑

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

<THIS> Shape this. | ∑

<INC\_DEC>Shape<THIS>ID <D’><I\_A> inc/dec <OTHER\_INC\_DEC>; | inc/dec <THIS>ID <I\_A> <OTHER\_INC\_DEC>;

<I\_A> Shape (<PARAM>).ID <I\_A> | ∑

<OTHER\_INC\_DEC> Shape ,<INC\_DEC> | ∑

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

<FOR> Shape for(<F1> <F2> ; <F3>) <FOR'>

<F1> Shape <DEC> | <ASSIGNMENT>

<F2> Shape <OE> | ∑

<F3> Shape this.ID <D’><I\_A> <SST\_TH>

| ID <D’><I\_A> <SST\_TH>

| inc/dec <THIS>ID <I\_A> <OTHER\_INC\_DEC>;

| ∑

<FOR'> Shape ; | {<MST><ELSE>}

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

<DO\_WHILE> Shape do{<MST><ELSE>}while(<OE>);

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

<IF> Shape if(<OE>)<IF’><IF\_ELSE>

<IF’> Shape ; | {<MST><ELSE>}

<IF\_ELSE> Shape else<IF’> | ∑

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

<A> Shape ,<OE><A> | ∑

<SIZE> Shape <OE>]<DIM> <A7> | ]<DIM> = {<OE><A>}<A8>

<A2> Shape ; | ,ID[<A1>] = {<OE><A>}<A2>

<A3> Shape ; | = {<A4>}

<A4> Shape ID<A5><A6>

<A5> Shape (<PARAM>) | [<OE>] | ∑

<A6> Shape ,ID<A5> | ∑

<A7> Shape ; | = {<OE><A>}<A2>

<A8>Shape; | ,ID[<SIZE>

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

<ASSIGNMENT> Shape <THIS> ID <D’><I\_A> <AO> <OE>;

<AO> Shape = | PMMDM

-------------------------------Print------------------------

<PRINT> Shape cout <PRINT’> ;

<PRINT’> Shape << <PRINT\_END> | ∑

<PRINT\_END> Shape <OE> <PRINT’> | endl <PRINT’>

----------------------------------Input---------------

<INPUT> Shape cin <INPUT’> ;

<INPUT’> Shape >> ID <INPUT\_END> | ∑

<INPUT\_END> Shape . ID <INPUT\_END> | [<OE>] <DIM> <INPUT\_END> | ∑