<program> --> <var\_dec> <program>

| <func\_dec> <program>

| EOF

<var\_dec> --> <params> SM;  
<func\_dec> --> <type\_stmt> ID LP <param\_list> RP LB <func\_body> RB

<cm\_id> --> CM ID <cm\_id>

| empty;

<params> --> <type\_stmt> ID <cm\_id>;  
<sm\_param> --> SM <params> <sm\_param>

| empty <param\_list> --> empty

| <params> <sm\_param> ;

<func\_body> --> <var\_dec> <func\_body>

| <stmt> <func\_body>

| empty; <stmt> --> assignment SM

| call SM

| return\_stmt SM;

<assignment>--> ID EQ <exp>;

<cm\_exp> --> CM <exp> <cm\_exp>

| empty;

<call> --> ID LP RP

| ID LP <exp> <cm\_exp> RP

<return\_stmt> --> RETURN <exp>

<exp> -->

<exp1> ADD <exp>

| <exp1>

;

<exp1> -->

: <exp2> SUB <exp2>

| <exp2>  
;

<exp2> -->

: <exp2> MUL <exp3>

|  <exp2> DIV <exp3>

|  <exp3> ;

<exp3> -->

: INTLIT

|  FLOATLIT

|  ID

|  <call>

|  LP <exp> RP ;

INTLIT: NUMBER+;

FLOATLIT: NUMBER+ ((DOT NUMBER+ [eE][+-]?)

| (DOT | ([eE][+-]?))) NUMBER+;

INT: 'int';  
FLOAT: 'float'; RETURN: 'return';

LB: '{'; RB: '}'; SM: ';'; CM: ','; EQ: '='; LP: '('; RP: ')'; ADD: '+'; SUB: '-'; MUL: '\*'; DIV: '/';

COLON: ':'; VAR: 'Var';

ID: LETTER (LETTER | NUMBER)\*;  
WS: [ \t\r\n\f]+ -> skip; // skip spaces, tabs, newlines