



FEBRUARY 17, 2021

A SWIFT LIKE LANGUAGE

COMPILER CONSTRUCTION

SAMPLE CODE

```
private final class st
{
    static class bc{}
}
class abc
{
    init(){}
    let b:Int;
    virtual func fn():Int
    {
        return 56;
    }
}
class ab:abc
{
    override func fn():Int
    {
        return 56+43;
    }
}
let a=abc();
a.b=(5+4*23);
a.fn(); // :(
var array:Int=[10][20];
func meow(){}
meow();
let d:Int=67;
if(d){}
else{}
try{}
catch{}
switch d{
case: 25
{
    d=67;
}
default: {
    d=35;
}
}
```

CONTEXT FREE GRAMMAR

$\langle \text{prog} \rangle \rightarrow \langle \text{SST} \rangle \langle \text{SSTs} \rangle$

$\langle \text{SSTs} \rangle \rightarrow \langle \text{SST} \rangle \langle \text{SSTs} \rangle$

$\langle \text{SSTs} \rangle \rightarrow \epsilon$

$\langle \text{SST} \rangle \rightarrow \langle \text{SST1} \rangle$

| $\langle \text{fn_cls_dec} \rangle$

| $\langle \text{if_else_st} \rangle$

| $\langle \text{while_st} \rangle$

| $\langle \text{do_while_st} \rangle$

| $\langle \text{switch_st} \rangle$

| $\langle \text{for_st} \rangle$

| $\langle \text{try_catch_st} \rangle$

$\langle \text{SST1} \rangle \rightarrow \text{ID } \langle \text{XY} \rangle$

$\langle \text{XY} \rangle \rightarrow \langle \text{DOP4} \rangle$

| $\rightarrow \langle \text{Arr} \rangle \langle \text{IDF} \rangle$

| $\rightarrow (\langle \text{Params} \rangle) \langle \text{Fo3} \rangle$

| $\rightarrow \langle \text{OP} \rangle \langle \text{OE} \rangle ;$

$\langle \text{DOP4} \rangle \rightarrow \text{DotOp } \langle \text{Fo} \rangle$

$\langle \text{Fo} \rangle \rightarrow \text{ID } \langle \text{Fo2} \rangle \langle \text{IDF} \rangle$

$\langle \text{IDF} \rangle \rightarrow (\langle \text{Par} \rangle) \langle \text{Fo3} \rangle$

| $\rightarrow \langle \text{DOP5} \rangle$

| $\rightarrow \langle \text{OP} \rangle \langle \text{OE} \rangle ;$

$\langle \text{Fo2} \rangle \rightarrow \langle \text{Arr} \rangle$

| $\rightarrow \epsilon$

$\langle \text{Fo3} \rangle \rightarrow ;$

| $\rightarrow \langle \text{DOP5} \rangle$

$\langle \text{DOP5} \rangle \rightarrow \text{DotOp ID} \langle \text{IDF} \rangle$

$\langle \text{Params} \rangle \rightarrow \text{ID} : \text{DT} \langle \text{Param2} \rangle$

$\mid \rightarrow \epsilon$

$\langle \text{Param2} \rangle \rightarrow , \langle \text{Params} \rangle$

$\mid \rightarrow \epsilon$

$\langle \text{SST3} \rangle \rightarrow \langle \text{oth} \rangle : \text{DT} \langle \text{as} \rangle$

$\mid \text{ AS CLASS } (\langle \text{Par} \rangle) ;$

$\langle \text{X} \rangle \rightarrow \langle \text{DOP} \rangle$

$\mid \langle \text{Arr} \rangle$

$\mid \epsilon$

$\langle \text{DOP} \rangle \rightarrow \text{DotOp} \langle \text{Fol1} \rangle$

$\langle \text{Fol1} \rangle \rightarrow \text{ID} \langle \text{func_ID} \rangle$

$\langle \text{func_ID} \rangle \rightarrow \langle \text{Fol2} \rangle \langle \text{DOP2} \rangle$

$\langle \text{func_ID} \rangle \rightarrow (\langle \text{Par} \rangle) \langle \text{Fol2} \rangle \langle \text{DOP3} \rangle$

$\langle \text{Fol2} \rangle \rightarrow \langle \text{Arr} \rangle$

$\mid \epsilon$

$\langle \text{DOP2} \rangle \rightarrow \text{DotOp} \langle \text{Fol1} \rangle$

$\langle \text{DOP2} \rangle \rightarrow \epsilon$

$\langle \text{DOP3} \rangle \rightarrow \text{DotOp ID} \langle \text{Fol2} \rangle \langle \text{DOP3} \rangle$

$\langle \text{DOP3} \rangle \rightarrow \epsilon$

$\langle \text{Arr} \rangle \rightarrow [\langle \text{IC} \rangle] \langle \text{2D} \rangle$

$\langle \text{2D} \rangle \rightarrow [\langle \text{IC} \rangle]$

$\mid \epsilon$

$\langle \text{IC} \rangle \rightarrow \text{ID}$

| int_const

<OP> → ASE

| AS

<oth> → , ID <oth>

| ε

<as> → <OP> <OE> ;

| ;

<OE> → <AE> <OE1>

<OE1> → OR <AE> <OE1>

| ε

<AE> → <RE> <AE1>

<AE1> → AND <RE> <AE1>

| ε

<RE> → <E> <RE1>

<RE1> → ROP <E> <RE1>

| ε

<E> → <T> <E1>

<E1> → PM <T> <E1>

| ε

<T> → <F> <T1>

<T1> → MDM <F> <T1>

| ε

<F> → <TH> ID <X>

| <const>

| (<E>)

<TH> → this .

$\mid \epsilon$
 $\langle \text{const} \rangle \rightarrow \text{int_const} \mid \text{float_const} \mid \text{string_const} \mid \text{char_const} \mid \text{bool_const}$
 $\langle \text{Par} \rangle \rightarrow \text{ID} : \text{DT} \langle \text{Par2} \rangle$
 $\langle \text{Par2} \rangle \rightarrow , \text{ID} : \text{DT} \langle \text{Par2} \rangle$
 $\mid \epsilon$
 $\langle \text{fn_cls_dec} \rangle \rightarrow \langle \text{AM} \rangle \langle \text{fcdg} \rangle$
 $\langle \text{fcdg} \rangle \rightarrow \langle \text{AM} \rangle \langle \text{AF} \rangle \text{CLASS ID} \langle \text{cl} \rangle \{ \langle \text{CBC} \rangle \}$
 $\langle \text{fcdg} \rangle \rightarrow \text{FUNC ID} (\langle \text{Par} \rangle) \langle \text{func_body} \rangle$
 $\langle \text{fcdg} \rangle \rightarrow \text{VAR ID} \langle \text{SST3} \rangle$
 $\langle \text{fcdg} \rangle \rightarrow \text{LET ID} \langle \text{SST3} \rangle$
 $\langle \text{CBC} \rangle \rightarrow \langle \text{CB} \rangle \langle \text{CBC} \rangle$
 $\mid \epsilon$
 $\langle \text{CB} \rangle \rightarrow \langle \text{AM} \rangle \langle \text{fcdc} \rangle \langle \text{CBC} \rangle$
 $\langle \text{CB} \rangle \rightarrow \text{VIRTUAL} \langle \text{fcdc} \rangle \langle \text{CB} \rangle$
 $\langle \text{CB} \rangle \rightarrow \text{OVERRIDE} \langle \text{fcdc} \rangle \langle \text{CB} \rangle$
 $\langle \text{CB} \rangle \rightarrow \langle \text{init_construct} \rangle$
 $\mid \epsilon$
 $\langle \text{fcdc} \rangle \rightarrow \langle \text{TM} \rangle \langle \text{func_var_cls} \rangle$
 $\langle \text{func_var_cls} \rangle \rightarrow \text{CLASS ID} \langle \text{cl} \rangle \{ \langle \text{CBC} \rangle \}$
 $\langle \text{func_var_cls} \rangle \rightarrow \text{FUNC ID} (\langle \text{Par} \rangle) \langle \text{func_body} \rangle$
 $\langle \text{func_var_cls} \rangle \rightarrow \text{VAR ID} \langle \text{SST3} \rangle$
 $\langle \text{func_var_cls} \rangle \rightarrow \text{LET ID} \langle \text{SST3} \rangle$
 $\langle \text{SST3} \rangle \rightarrow \langle \text{oth} \rangle : \text{DT} \langle \text{as} \rangle$
 $\langle \text{SST3} \rangle \rightarrow \text{AS ID} (\langle \text{Par} \rangle) ;$
 $\langle \text{init_construct} \rangle \rightarrow \text{INIT} (\langle \text{Params} \rangle) \{ \langle \text{self} \rangle \}$
 $\langle \text{self} \rangle \rightarrow \text{SELF DotOp ID AS ID} ; \langle \text{self} \rangle$

| ϵ

<as> \rightarrow <OP> <as1>;

<as1> \rightarrow ;

<as1> \rightarrow <OE>

<AM> \rightarrow AM

<AM> $\rightarrow \epsilon$

<TM> \rightarrow AF

<TM> $\rightarrow \epsilon$

<func_body> \rightarrow <RET> | <NORET>

<RET> \rightarrow : <RETURN>

<RETURN> \rightarrow <DIR> { <MST> <return_st> }

| VOID { <MST> }

<NORET> \rightarrow { <MST> }

<DIR> \rightarrow Integer | Float | String | Char | Bool

<return_st> \rightarrow RETURN <OE>

<MST> \rightarrow <SST> <MST>

| ϵ

<cl> \rightarrow : CLASS

| ϵ

<if_else_st> \rightarrow IF (<cond>) { <if_else_body> } <oelse>

<cond> \rightarrow ID | <const> | <ID_const> ROP <ID_const>

<ID_const> \rightarrow ID | <const>

<if_else_body> \rightarrow <SST><MST>

| ϵ

$\langle \text{while_st} \rangle \rightarrow \text{WHILE} (\langle \text{cond} \rangle) \langle \text{body} \rangle$

$\langle \text{body} \rangle \rightarrow ; \mid \langle \text{SST} \rangle \mid \{ \langle \text{MST} \rangle \}$

$\langle \text{do_while_st} \rangle \rightarrow \text{DO} \{ \langle \text{MST} \rangle \} \text{WHILE} (\langle \text{cond} \rangle) ;$

$\langle \text{switch_st} \rangle \rightarrow \text{SWITCH} \langle \text{ID_const} \rangle \{ \langle \text{case} \rangle \langle \text{def} \rangle \}$

$\langle \text{case} \rangle \rightarrow \langle \text{case_st} \rangle \langle \text{cases} \rangle$

$\langle \text{cases} \rangle \rightarrow \langle \text{case_st} \rangle$

$\mid \epsilon$

$\langle \text{case_st} \rangle \rightarrow \text{CASE} \langle \text{ID_const} \rangle : \langle \text{s_body} \rangle$

$\langle \text{s_body} \rangle \rightarrow \langle \text{SST} \rangle$

$\mid \{ \langle \text{MST1} \rangle \}$

$\langle \text{MST1} \rangle \rightarrow \langle \text{SST} \rangle \langle \text{MST2} \rangle$

$\langle \text{MST2} \rangle \rightarrow \langle \text{SST} \rangle \langle \text{MST2} \rangle$

$\mid \epsilon$

$\langle \text{def} \rangle \rightarrow \text{DEFAULT} : \langle \text{body_s} \rangle$

$\langle \text{for_st} \rangle \rightarrow \text{FOR} (\langle \text{c1} \rangle \langle \text{c2} \rangle ; \langle \text{c3} \rangle) \langle \text{body} \rangle$

$\langle \text{c1} \rangle \rightarrow \text{ID AS} \langle \text{OE} \rangle \mid \text{DT ID AS} \langle \text{OE} \rangle \mid ;$

$\langle \text{c2} \rangle \rightarrow \langle \text{cond} \rangle \mid \epsilon$

$\langle \text{c3} \rangle \rightarrow \langle \text{assign_st} \rangle \mid \epsilon$

$\langle \text{try_catch_st} \rangle \rightarrow \text{TRY} \{ \langle \text{MST} \rangle \} \langle \text{OPT} \rangle$

$\langle \text{OPT} \rangle \rightarrow \text{CATCH} \{ \langle \text{MST} \rangle \} \langle \text{OPT2} \rangle$

$\langle \text{OPT2} \rangle \rightarrow \text{CATCH} \{ \langle \text{MST} \rangle \} \langle \text{OPT2} \rangle \langle \text{OPT2} \rangle$

$\mid \text{FINAL} \{ \langle \text{MST} \rangle \}$

$\mid \epsilon$

LL1 PROVED GRAMMAR

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<prog> → <SST> <SSTs>	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ \$ }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY
<SSTs> → <SST> <SSTs>	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ \$ }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY
<SSTs> → ε	ε	{ \$ }	{ \$ }
<SST> → <SST1>	{ ID }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ ID }
<SST> → <fn_cls_dec>	{ AM, AF, FUNC, CLASS, LET, VAR }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ AM, AF, FUNC, CLASS, LET, VAR }
<SST> → <if_else_st>	{ IF }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ IF }
<SST> → <while_st>	{ WHILE }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ WHILE }
<SST> → <do_while_st>	{ DO }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ DO }
<SST> → <switch_st>	{ SWITCH }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ SWITCH }
<SST> → <for_st>	{ FOR }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ FOR }
<SST> → <try_catch_st>	{ TRY }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ TRY }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<SST1> → ID <XY>	{ ID }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ ID }
<XY> → <DOP4>	{ DotOp }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ DotOp }
<XY> → <Arr> <IDF>	{ [}		{ [}
<XY> → (<Params>) <Fo3>	{ (}		{ (}
<XY> → <OP> <OE> ;	{ ASE, AS }		{ ASE, AS }
<DOP4> → DotOp <Fo>	{ DotOp }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ DotOp }
<Fo> → ID <Fo2> <IDF>	{ ID }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ ID }
<IDF> → (<Par>) <Fo3>	{ (}	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ (}
<IDF> → <DOP5>	{ DotOp }		{ DotOp }
<IDF> → <OP> <OE> ;	{ ASE, AS }		{ ASE, AS }
<Fo2> → <Arr>	{ [}	{ (, DotOp, AS, AE }	{ [}
<Fo2> → ε	ε		{ (, DotOp, AS, AE }
<Fo3> → ;	{ ; }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ ; }
<Fo3> → <DOP5>	{ DotOp }		{ DotOp }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<DOP5> → DotOp ID<IDF>	{ DotOp }	{ ASE, AS }	{ DotOp }
<SST3> → <oth> : DT <as>	{ , : }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ , : }
<SST3> → AS CLASS (<Params>) ;	{ AS }		{ AS }
<Params> → ID : DT <Param2>	{ ID }	{) }	{ ID }
<Params> → ε		{) }	
<Param2> → , <Params>	{ , }	{) }	{ , }
<Param2> → ε	ε		{) }
<OP> → ASE	{ ASE }	{ this, int_const, string_const, char_const, bool_const,float_const, (}	{ ASE }
<OP> → AS	{ AS }		{ AS }
<oth> → , ID <oth>	{ , }	{ : }	{ , }
<oth> → ε	ε		{ : }
<as> → <OP> <OE> ;	{ ASE, AS }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ ASE, AS }
<as> → ;	{ ; }		{ ; }
<Par> → ID : DT <Par2>	{ ID }	{) }	{ ID }
<Par> → ε	{ ε }		{) }
<Par2> → , ID : DT <Par2>	{ , }	{) }	{ , }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
$\langle \text{Par2} \rangle \rightarrow \epsilon$	{ ϵ }	{) }	{) }
$\langle X \rangle \rightarrow \langle \text{DOP} \rangle$	{ DotOp }	{ ASE, AS }	{ DotOp }
$\langle X \rangle \rightarrow \langle \text{Arr} \rangle$	{ [}		{ [}
$\langle X \rangle \rightarrow \epsilon$	ϵ		{ ASE, AS }
$\langle \text{DOP} \rangle \rightarrow \text{DotOp}$ $\langle \text{Fol1} \rangle$	{ DotOp }	{ ASE, AS }	{ DotOp }
$\langle \text{Fol1} \rangle \rightarrow \text{ID}$ $\langle \text{func_ID} \rangle$	{ ID }	{ ASE, AS }	{ ID }
$\langle \text{func_ID} \rangle \rightarrow \langle \text{Fol2} \rangle$ $\langle \text{DOP2} \rangle$	{ [, DotOp, ASE, AS }		{ [, DotOp, ASE, AS }
$\langle \text{func_ID} \rangle \rightarrow (\langle \text{Par} \rangle)$ $\langle \text{Fol2} \rangle \langle \text{DOP3} \rangle$	{ (}		{ (}
$\langle \text{Fol2} \rangle \rightarrow \langle \text{Arr} \rangle$	{ [}	{ DotOp, ASE, AS }	{ [}
$\langle \text{Fol2} \rangle \rightarrow \epsilon$	ϵ		{ DotOp, ASE, AS }
$\langle \text{DOP2} \rangle \rightarrow \text{DotOp}$ $\langle \text{Fol1} \rangle$	{ DotOp }	{ ASE, AS }	{ DotOp }
$\langle \text{DOP2} \rangle \rightarrow \epsilon$	ϵ		{ ASE, AS }
$\langle \text{DOP3} \rangle \rightarrow \text{DotOp ID}$ $\langle \text{Fol2} \rangle \langle \text{DOP3} \rangle$	{ DotOp }	{ ASE, AS }	{ DotOp }
$\langle \text{DOP3} \rangle \rightarrow \epsilon$	ϵ		{ ASE, AS }
$\langle \text{Arr} \rangle \rightarrow [\langle \text{IC} \rangle]$ $\langle \text{2D} \rangle$	{ [}	{ DotOp, ASE, AS }	{ [}
$\langle \text{2D} \rangle \rightarrow [\langle \text{IC} \rangle]$	{ [}	{ DotOp, ASE, AS }	{ [}
$\langle \text{2D} \rangle \rightarrow \epsilon$	ϵ		{ DotOp, ASE, AS }
$\langle \text{IC} \rangle \rightarrow \text{ID}$	{ ID }	{] }	{ ID }
$\langle \text{IC} \rangle \rightarrow \text{int_const}$	{ int_const }		{ int_const }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
$\langle Y \rangle \rightarrow \langle DOP \rangle$	{DotOp}	{ (}	{DotOp}
$\langle Y \rangle \rightarrow \langle Arr \rangle$	{ [}		{ [}
$\langle Y \rangle \rightarrow \epsilon$	ϵ		{ (}
$\langle DOP \rangle \rightarrow \text{DotOp}$ $\langle \text{Fol3} \rangle$	{DotOp}	{ (}	{DotOp}
$\langle \text{Fol3} \rangle \rightarrow \text{ID } \langle \text{Fol2} \rangle$ $\langle \text{DOP2} \rangle$	{ID}	{ (}	{ID}
$\langle \text{Fol2} \rangle \rightarrow \langle Arr \rangle$	{ [}	{ DotOp, (}	{ [}
$\langle \text{Fol2} \rangle \rightarrow \epsilon$	ϵ		{ DotOp, (}
$\langle \text{DOP2} \rangle \rightarrow \text{DotOp}$ $\langle \text{Fol3} \rangle$	{DotOp}	{ (}	{DotOp}
$\langle \text{DOP2} \rangle \rightarrow \epsilon$	ϵ		{ (}
$\langle Arr \rangle \rightarrow [\langle IC \rangle] \langle 2D \rangle$	{ [}	{ DotOp, (}	{ [}
$\langle 2D \rangle \rightarrow [\langle IC \rangle]$	{ [}	{ DotOp, (}	{ [}
$\langle 2D \rangle \rightarrow \epsilon$	ϵ		{ DotOp, (}
$\langle IC \rangle \rightarrow \text{ID}$	{ID}	{] }	{ID}
$\langle IC \rangle \rightarrow \text{int_const}$	{int_const}		{int_const}

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
$\langle OE \rangle \rightarrow \langle AE \rangle \langle OE1 \rangle$	{ this, int_const, string_const, char_const, bool_const, float_const, (}	{ ; }	{ this, int_const, string_const, char_const, bool_const, float_const, (, ID }
$\langle OE1 \rangle \rightarrow OR \langle AE \rangle \langle OE1 \rangle$	{ OR }	{ ; }	{ OR }
$\langle OE1 \rangle \rightarrow \epsilon$	ϵ		{ ; }
$\langle AE \rangle \rightarrow \langle RE \rangle \langle AE1 \rangle$	{ this, int_const, string_const, char_const, bool_const, float_const, (}	{ OR, ; }	{ this, int_const, string_const, char_const, bool_const, float_const, (, ID }
$\langle AE1 \rangle \rightarrow AND \langle RE \rangle \langle AE1 \rangle$	{ AND }	{ OR, ; }	{ AND }
$\langle AE1 \rangle \rightarrow \epsilon$	ϵ		{ OR, ; }
$\langle RE \rangle \rightarrow \langle E \rangle \langle RE1 \rangle$	{ this, int_const, string_const, char_const, bool_const, float_const, (}	{ AND, OR, , ; }	{ this, int_const, string_const, char_const, bool_const, float_const, (, ID }
$\langle RE1 \rangle \rightarrow ROP \langle E \rangle \langle RE1 \rangle$	{ ROP }	{ AND, OR, , ; }	{ ROP }
$\langle RE1 \rangle \rightarrow \epsilon$	ϵ		{ AND, OR, , ; }
$\langle E \rangle \rightarrow \langle T \rangle \langle E1 \rangle$	{ this, int_const, string_const, char_const, bool_const, float_const, (}	{ ROP, AND, OR, , , ; }	{ this, int_const, string_const, char_const, bool_const, float_const, (, ID }
$\langle E1 \rangle \rightarrow PM \langle T \rangle \langle E1 \rangle$	{ PM }	{ ROP, AND, OR, , ; }	{ PM }
$\langle E1 \rangle \rightarrow \epsilon$	ϵ		{ ROP, AND, OR, , ; }
$\langle T \rangle \rightarrow \langle F \rangle \langle T1 \rangle$	{ this, int_const, string_const, char_const, bool_const, float_const, (}	{ PM, ROP, AND, OR, , ; }	{ this, int_const, string_const, char_const, bool_const, float_const, (, ID }
$\langle T1 \rangle \rightarrow MDM \langle F \rangle \langle T1 \rangle$	{ MDM }	{ PM, ROP, AND, OR, , ; }	{ MDM }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
$\langle T1 \rangle \rightarrow \epsilon$	ϵ		{ PM, ROP, AND, OR , ; }
$\langle F \rangle \rightarrow \langle TH \rangle ID \langle X \rangle$	{ this, ID }	{ MDM,), PM, ROP, AND, OR , ; }	{ this, ID }
$\langle F \rangle \rightarrow \langle const \rangle$	{ int_const, string_const, char_const, bool_const_float_const }		{ int_const, string_const, char_const, bool_const_float_const }
$\langle F \rangle \rightarrow (\langle E \rangle)$	{ (}		{ (}
$\langle TH \rangle \rightarrow this .$	{ this }	{ ID }	{ this }
$\langle TH \rangle \rightarrow \epsilon$	ϵ		{ ID }
$\langle const \rangle \rightarrow int_const$	{ int_const }	{ MDM,), PM, ROP, AND, OR , ; }	{ int_const }
$\langle const \rangle \rightarrow float_const$	{ float_const }		{ float_const }
$\langle const \rangle \rightarrow char_const$	{ char_const }		{ char_const }
$\langle const \rangle \rightarrow string_const$	{ string_const }		{ string_const }
$\langle const \rangle \rightarrow bool_const$	{ bool_const }		{ bool_const }
$\langle X \rangle \rightarrow \langle DOP \rangle$	{ DotOp }	{ MDM,), PM, ROP, AND, OR , ; }	{ DotOp }
$\langle X \rangle \rightarrow \langle Arr \rangle$	{ [}		{ [}
$\langle X \rangle \rightarrow \epsilon$	ϵ		{ MDM,), PM, ROP, AND, OR , ; }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<code><fn_cls_dec> → <AM><fcdg></code>	{AM, AF, CLASS, FUNC , LET, VAR}	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{AM, AF, CLASS, FUNC, LET, VAR}
<code><fcdg> → <AM> <AF> CLASS ID <c > { <CBC> }</code>	{AF, CLASS}		{AM, AF, CLASS}
<code><AM> → AM</code>	{AM}	{AF, CLASS, FUNC, VAR, LET}	{AM}
<code><AM> → ε</code>	ε		{AF, CLASS, FUNC, VAR, LET}
<code><AF> → AF</code>	{AF}		{AF}
<code><AF> → ε</code>	ε		{CLASS}
<code><fcdg> → FUNC ID (<Par>) <func_body></code>	{FUNC}	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{FUNC}
<code><fcdg> → VAR ID <SST3></code>	{VAR}		{VAR}
<code><fcdg> → LET ID <SST3></code>	{LET}		{LET}
<code><CBC> → <CB><CBC></code>	{AM, AF, TM, FUNC, VAR, CLASS, VIRTUAL, LET, OVERRIDE, INIT, } }	{ } }	{AM, AF, TM, FUNC, VAR, CLASS, VIRTUAL, LET, OVERRIDE, INIT, } }
<code><CBC> → ε</code>	ε	{ } }	{ } }
<code><CB> → <AM> <fcdc><CBC></code>	{AM, AF }	{AM, AF, TM, FUNC, VAR, CLASS, VIRTUAL, LET, OVERRIDE, INIT, } }	{AM, AF, TM, FUNC, VAR, CLASS, VIRTUAL, OVERRIDE, INIT, } }
<code><CB> → VIRTUAL <fcdc> <CB></code>	{VIRTUAL}		{VIRTUAL}
<code><CB> → OVERRIDE <fcdc> <CB></code>	{ OVERRIDE }		{OVERRIDE}

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<CB> → ID <fcdc> <CB>	{ ID }		{ID}
<CB> → <init_construct>	{ INIT }		{INIT}
<CB> → ε	ε	{AM, AF, TM, FUNC, VAR, CLASS, VIRTUAL, LET, OVERRIDE, INIT, } }	{AM, AF, TM, FUNC, VAR, CLASS, VIRTUAL, LET, OVERRIDE, INIT, } }
<fcdc> → <TM><func_var_cls>	{TM, FUNC, VAR}	{AM, AF, TM, FUNC, VAR, CLASS, VIRTUAL, LET, OVERRIDE, INIT, } }	{AM, TM, FUNC, VAR, LET, CLASS}
<TM> → TM	{TM}	{CLASS, FUNC, VAR, LET, INIT}	
<TM> → ε	ε		{CLASS, FUNC, VAR, LET}
<func_var_cls> → CLASS ID <cl> { <CBC> }	{ CLASS }	{AM, AF, TM, FUNC, VAR, CLASS, VIRTUAL, LET, OVERRIDE, INIT, } }	{ CLASS }
<func_var_cls> → FUNC ID (<Par>) <func_body>	{FUNC}		{FUNC}
<func_var_cls> → VAR ID <SST3>	{VAR}		{VAR}
<func_var_cls> → LET ID <SST3>	{LET}		{LET}
<SST3> → <oth> : DT <as>	{ , : }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ , : }
<SST3> → AS ID (<Par>) ;	{ AS }		{ AS }
<init_construct> → INIT (<Params>) { <self> }	{ INIT }	{AM, AF, TM, FUNC, VAR, CLASS, VIRTUAL, LET, OVERRIDE, INIT, } }	{ INIT }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<self> → SELF DotOp ID AS ID ; <self>	{ SELF }	{ }	{ SELF }
<self> → ε	ε	{ }	{ }
<as> → <OP> <as1>;	{ ASE, AS }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ ASE, AS }
<as1> → ;			
<as1> → <OE>			{ this, int_const, string_const, char_const, bool_const,float_const, (, ID }
<AM> → AM	{ AM }	{ FUNC, CLASS, LET, VAR, TM }	{ AM }
<AM> → ε	ε		{ FUNC, CLASS, LET, VAR, TM }
<TM> → AF	{ AF }	{ FUNC, CLASS, LET, VAR }	
<TM> → ε	ε		{ FUNC, CLASS, LET, VAR }
<func_body> → <RET> <NORET>	{ :, { }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ :, { }
<RET> → : <RETURN>	{ : }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ : }
<RETURN> → <DIR> { <MST> <return_st> }	{ Integer, String, Bool, Char, Float }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ Integer, String, Bool, Char, Float }
<RETURN> → VOID { <MST> }	{ VOID }		{ VOID }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<NORET> → { <MST> }	{{}}	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{{}}
<DIR> → DT	{ Integer }	{{}}	{ DT }
<return_st> → RETURN <OE>;	{ RETURN }	{{}}	{ RETURN }
<MST> → ϵ	ϵ		{ RETURN. } }
<cl> → : ID	{ : }	{{}}	{ : }
<cl> → ϵ	ϵ		{{}}

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<if_else_st> → IF (<cond>) { <if_else_body> } <oelse>	{ IF }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ IF }
<cond> → <ID_const> <cond2>	{ ID, int_const, string_const, char_const, bool_const,float_const }		{ ID, int_const, string_const, char_const, bool_const,float_const }
<ID_const> → ID	{ ID }	{ ROP }	{ ID }
<ID_const> → <const>	{ int_const, string_const, char_const, bool_const,float_const }		{ int_const, string_const, char_const, bool_const,float_const }
<cond2> → ROP <ID_const>	{ROP}	())	{ROP}
<cond2> → ε			())
<if_else_body> → <MST>	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }
<if_else_body> → ε			{ }
<oelse> → { <MST> }			
<while_st> → WHILE (<cond>) <body>	{ WHILE }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ WHILE }
<body> → ;	{ ; }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ ; }
<body> → <SST>	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }		{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }
<body> → { <MST> }	{ }		{ }
<do_while_st> → DO { <MST> } WHILE (<cond>) ;	{ DO }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ DO }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<MST> → <SST> <MST>	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{ }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY
MST> → ε			{ }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<for_st> → FOR (<c1> <c2> ; <c3>) <body>	{ FOR }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ FOR }
<c1> → <in1> ID AS int_const;	{ INT, ID }		{ INT, ID }
<c1> → ;	{ ; }		{ ; }
<in1> → INT	{ INT }		{ INT }
<in1> → ε	{ ID }		{ ID }
<c2> → <ID_const> ROP <ID_const>	{ ID, int_const, string_const, char_const, bool_const_float_const }	{ ; }	{ ID, int_const, string_const, char_const, bool_const_float_const }
<c2> → ε	ε		{ ; }
<c3> → <assign_st>	{ ID }	{) }	{ ID }
<c3> → ε	ε		{) }
<assign_st> → ID <X> <OP> <OE>	{ ID }	{ ID, int_const, string_const, char_const, bool_const_float_const, ;,) }	{ ID }
<X> →		{ ASE, AE }	
<X> → ε			
<try_catch_st> → TRY { <MST> } <OPT>	{ TRY }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ TRY }
<OPT> → CATCH { <MST> } <OPT2>	{ CATCH }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ CATCH }
<OPT2> → CATCH { <MST> } <OPT2>	{ CATCH }	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{ CATCH }

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
<OPT2> → FINAL { <MST> }	{FINAL}		{FINAL}
<OPT2> → ϵ	ϵ		{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }
<OP>	{ASE,AS}	{ this, ID, int_const, string_const, char_const, bool_const_float_const, (}	{ASE,AS}
<OE>	{ this, ID, int_const, string_const, char_const, bool_const_float_const, (}	{) }	{ this, ID, int_const, string_const, char_const, bool_const_float_const, (}

PRODUCTION RULE	FIRST SET	FOLLOW SET	SELECTION SET
$\langle \text{switch_st} \rangle \rightarrow$ SWITCH $\langle \text{ID_const} \rangle \{$ $\langle \text{case} \rangle \langle \text{def} \rangle \}$	{SWITCH}	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY SWITCH, \$ }	{SWITCH}
$\langle \text{case} \rangle \rightarrow \langle \text{cases} \rangle$	{CASE}	{DEFAULT}	{CASE, DEFAULT}
$\langle \text{cases} \rangle \rightarrow \langle \text{case_st} \rangle$	{CASE}	{DEFAULT}	{CASE}
$\langle \text{cases} \rangle \rightarrow \epsilon$	ϵ		{DEFAULT}
$\langle \text{case_st} \rangle \rightarrow \text{CASE}$ $\langle \text{ID_const} \rangle :$	{CASE}	{DEFAULT}	{CASE}
$\langle \text{c_body} \rangle \rightarrow \langle \text{SST} \rangle$	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY	{DEFAULT}	{ LET, VAR, ID, AF, AM, CLASS, FUNC, DO, WHILE, IF, FOR, TRY
$\langle \text{c_body} \rangle \rightarrow$ $\{ \langle \text{SST} \rangle \langle \text{MST} \rangle \}$	{ {} }		{ {} }
$\langle \text{MST2} \rangle \rightarrow \epsilon$	ϵ		{ {} }
$\langle \text{def} \rangle \rightarrow \text{DEFAULT} :$ $\langle \text{c_body} \rangle$	{DEFAULT}	{ {} }	{DEFAULT}