FIRST and FOLLOW Sets

# FIRST Sets

<program> : <dec>, <main>

<main> : main

<dec> : public, private, DT, ID, ε

<struct\_dec> : public, private

<access\_modifier> : public, private

<extends> : extends, ε

<struct\_body> : DT, ID

<struct\_body\_tail> : DT, ID, ε

<struct\_chidlren> : DT, ID

<dt\_decORfunc\_dec> : (, =, ;

<func\_dec> : (

<param\_list> : DT, ε

<param\_list\_tail> : ,, ε

<param> : DT

<Body> : {

<Constructor> : ID

<dt\_dec> : =, ;, ε

<var\_init> : =, ε

<var\_init\_tail> : ,, ε

<Const\_or\_ID> : Const, ID

<array\_dec> : DT, ID

<arr\_type> : DT, ID

<arr\_const\_or\_id> : ε, Const, ID

<SST> : while, for, if, do, ID, try, throw, return, continue, break, DT, (

<MST> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, ε

<while\_loop> : while

<cond> : Const, ID, (

<ROP> : RO1, RO2

<loop\_body> : ;, <SST>, {

<for\_loop> : for

<F1> : DT, ID, ;

<F2> : Const, ID, (, ε

<F3> : Expr, ε

<ExprList> : Expr

<ExprListTail> : Expr, ε

<inc\_dec> : ID

<assign\_st> : ID

<assign\_options> : Const, ID, exp

<if> : if

<else> : else, null

<do\_while> : do

<this> : this

<func\_call> : ID

<arg\_list> : Const, ID, ε

<arg\_list\_tail> : ,, ε

<arg> : Const, ID

<Expr> : AssignExpr

<AssignExpr> : OrExpr

<AssignExpr'> : =, +=, -=, \*=, /=, %=, ε

<OrExpr> : AndExpr

<OrExpr'> : OR, ε

<AndExpr> : EqualityExpr

<AndExpr'> : AND, ε

<EqualityExpr> : RelationalExpr

<EqualityExpr'> : ==, !=, ε

<RelationalExpr> : AdditiveExpr

<RelationalExpr'> : <, >, <=, >=, ε

<AdditiveExpr> : MultiplicativeExpr

<AdditiveExpr'> : +, -, ε

<MultiplicativeExpr> : UnaryExpr

<MultiplicativeExpr'> : \*, /, %, ε

<UnaryExpr> : inc\_dec, !, <Primary>

<Primary> : PrimaryRef, Const, (, Ref, FuncCall, ArrayCall

<return> : return

<return\_options> : ID, const, exp, null

<continue> : continue

<break> : break

<try> : try

<catch\_list> : catch

<catch\_list\_tail> : catch, ε

<throw> : throw

<throw\_options> : ID, Const, new

<const> : int, float, char, string, bool

# FOLLOW Sets

<program> : $

<main> : public, private, DT, ID, $

<dec> : main, $

<struct\_dec> : main, $

<access\_modifier> : struct

<extends> : {

<struct\_body> : }

<struct\_body\_tail> : }

<struct\_chidlren> : DT, ID, }

<dt\_decORfunc\_dec> : DT, ID, }

<func\_dec> : DT, ID, }

<param\_list> : )

<param\_list\_tail> : )

<param> : ,, )

<Body> : DT, ID, }

<Constructor> : DT, ID, }

<dt\_dec> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<var\_init> : ,, ;

<var\_init\_tail> : ;

<Const\_or\_ID> : ,, ;, ), RO1, RO2

<array\_dec> : DT, ID, }

<arr\_type> : ID

<arr\_const\_or\_id> : }

<SST> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<MST> : }

<while\_loop> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<cond> : )

<ROP> : Const, ID

<loop\_body> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<for\_loop> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<F1> : ;

<F2> : ;

<F3> : )

<ExprList> : )

<ExprListTail> : )

<inc\_dec> : Const, ID

<assign\_st> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<assign\_options> : ;

<if> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<else> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<do\_while> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<this> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<func\_call> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<arg\_list> : )

<arg\_list\_tail> : )

<arg> : ,, )

<Expr> : ), ;

<AssignExpr> : ), ;

<AssignExpr'> : ), ;

<OrExpr> : =, +=, -=, \*=, /=, %=, ), ;

<OrExpr'> : ), ;

<AndExpr> : OR, ), ;

<AndExpr'> : ), ;

<EqualityExpr> : AND, ), ;

<EqualityExpr'> : ), ;

<RelationalExpr> : ==, !=, ), ;

<RelationalExpr'> : ), ;

<AdditiveExpr> : <, >, <=, >=, ), ;

<AdditiveExpr'> : ), ;

<MultiplicativeExpr> : +, -, ), ;

<MultiplicativeExpr'> : ), ;

<UnaryExpr> : \*, /, %, ), ;

<Primary> : \*, /, %, ), ;

<return> : while, for, if, do, ID, try, throw, return, continue, break, DT, (, }

<return\_options> : ;

<continue> : ;

<break> : ;

<try> : while, for, if, do, ID, throw, return, continue, break, DT, (, }

<catch\_list> : while, for, if, do, ID, throw, return, continue, break, DT, (, }

<catch\_list\_tail> : while, for, if, do, ID, throw, return, continue, break, DT, (, }

<throw> : ;

<throw\_options> : :