

START:

$\langle S \rangle \rightarrow$	$\langle \text{class\_dec} \rangle \langle S \rangle \mid \varepsilon$
$\text{First } S =$	$\{ \text{public, private, abstract, class, } \varepsilon \}$
$\text{Follow } S =$	$\{ \$ \}$
$\text{Selection} =$	$\{ \text{public, private, abstract, class, } \$ \}$

Expression:

$\langle OE \rangle \rightarrow$	$\langle AE \rangle \langle OE' \rangle$
$\text{First} \langle OE \rangle =$	$\{ \text{this., ID, num, deci, alpha, str, Boolean, null, ++ --, !, [} \}$
$\text{follow} \langle OE \rangle =$	$\{ ], : , , , ( \}$
$\text{selection} =$	$\{ \text{this., ID, num, deci, alpha, str, Boolean, null, ++ --, !, [} \}$

$\langle OE' \rangle \rightarrow$	$\text{OR } \langle AE \rangle \langle OE' \rangle$
$\langle OE' \rangle \rightarrow$	$E$
$\text{First} \langle OE' \rangle =$	$\{ \text{OR, } \varepsilon \}$
$\text{follow} \langle OE' \rangle =$	$\{ ], : , , , ( \}$
$\text{selection} =$	$\{ \text{OR, }, : , , , ( \}$

$\langle AE \rangle \rightarrow$	$\langle RE \rangle \langle AE' \rangle$
$\langle AE' \rangle \rightarrow$	$AND \langle RE \rangle \langle AE' \rangle$
$\langle AE' \rangle \rightarrow$	$E$
$First\langle AE' \rangle =$	$\{AND, \varepsilon\}$
$follow\langle AE' \rangle =$	$\{OR, ], : , , , ( \}$
$selection =$	$\{AND, OR, ], : , , , ( \}$

$\langle RE \rangle \rightarrow$	$\langle E \rangle \langle RE' \rangle$
$First\langle RE \rangle =$	$\{this. , ID , num, deci , alpha , str, Boolean , null, ++ --, ! , [ \}$
$follow\langle RE \rangle =$	$\{AND , OR, ], : , , , ( \}$
$selection =$	$\{this. , ID , num, deci , alpha , str, Boolean , null, ++ --, ! , [ \}$

$\langle RE' \rangle \rightarrow$	$ROP \langle E \rangle \langle RE' \rangle$
$\langle RE' \rangle \rightarrow$	$\varepsilon$
$First\langle RE' \rangle =$	$\{ROP, \varepsilon\}$
$follow\langle RE' \rangle =$	$\{AND , OR, ], : , , , ( \}$
$selection =$	$\{ROP, AND , OR, ], : , , , ( \}$

$\langle E \rangle \rightarrow$	$\langle T \rangle \langle E' \rangle$
$First\langle E \rangle =$	$\{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, ! , [ \}$
$follow\langle E \rangle =$	$\{ ROP , AND , OR, ], : , , , ( \}$
$selection =$	$\{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, ! , [ \}$

$\langle E' \rangle \rightarrow$	$PM \langle T \rangle \langle E' \rangle$
$\langle E' \rangle \rightarrow$	$\epsilon$
$First\langle E' \rangle =$	$\{ PM, \epsilon \}$
$follow\langle E' \rangle =$	$\{ ROP , AND , OR, ], : , , , ( \}$
$selection =$	$\{ PM, ROP , AND , OR, ], : , , , ( \}$

$\langle T \rangle \rightarrow$	$\langle F \rangle \langle T' \rangle$
$First\langle T \rangle =$	$\{ this., ID , num, deci , alpha , str, Boolean , null, ++ --, ! , [ \}$
$follow\langle T \rangle =$	$\{ PM, ROP , AND , OR, ], : , , , ( \}$
$selection =$	$\{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, ! , [ \}$

$\langle T' \rangle \rightarrow$	$MDM \langle F \rangle \langle T' \rangle$
$\langle T' \rangle \rightarrow$	$\varepsilon$
$First\langle T' \rangle =$	$\{MDM, \varepsilon\}$
$follow\langle T' \rangle =$	$\{PM, ROP, AND, OR, ], : , , , (\}$
$selection =$	$\{MDM, PM, ROP, AND, OR, ], : , , , (\}$

$\langle F \rangle \rightarrow$	$\langle Th \rangle ID \langle F' \rangle$
$\langle F \rangle \rightarrow$	$\langle const \rangle$
$\langle F \rangle \rightarrow$	$[ \langle OE \rangle ]$
$\langle F \rangle \rightarrow$	$\langle inc\_dec \rangle ID$
$\langle F \rangle \rightarrow$	$! \langle F \rangle$
$First\langle F \rangle =$	$\{this. , ID , num, deci , alpha , str, Boolean , null, ++ --, ! , [ \}$
$follow\langle F \rangle =$	$\{MDM , PM, ROP , AND , OR, ], : , , , (\}$
$selection =$	$\{this. , ID , num, deci , alpha , str, Boolean , null, ++ --, ! , [ \}$

$\langle F' \rangle \rightarrow$	$[\langle PL \rangle]$
$\langle F' \rangle \rightarrow$	$\langle inc\_dec \rangle$
$\langle F' \rangle \rightarrow$	$\epsilon$
$First\langle F' \rangle =$	$\{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, !, C \}$
$follow\langle F' \rangle =$	$\{ MDM , PM, ROP , AND , ], : , , , ( \}$
$selection =$	$\{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, !,$

$\langle Th \rangle \rightarrow$	$this.$
$\langle Th \rangle \rightarrow$	$\epsilon$
$First\langle Th \rangle =$	$\{this., \epsilon\}$
$follow\langle Th \rangle =$	$\{ID\}$
$Selection =$	$\{this. , ID\}$

Single Statement(SST) & Multi Statement(MST):

<SST> →	<dec>
<SST> →	<switch_case>
<SST> →	<array>
<SST> →	<2d-array>
<SST> →	<func_dec>
<SST> →	<for_st>
<SST> →	<if_else>
<SST> →	<Array-L>
<SST> →	<While_st>
<SST> →	ID <COM>
<SST> →	<inc_dec_st>
<SST> →	end:
<SST> →	jump:
<SST> →	ID <X>:
<SST> →	this.ID:
First<SST>=	{switch, DT, public, private, void, if, ArrayList, for, while, ID, ++, --, end: jump:, this.ID:}
follow<SST>=	{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }
selection =	{ switch, DT, public, private, void, if,

	<i>ArrayList, for, while, ID, ++, --, end: jump:, this.ID:}</i>
--	---

<i>&lt;COM&gt; →</i>	<i>&lt;obj_dec_list&gt;</i>
<i>&lt;COM&gt; →</i>	<i>&lt;X&gt; &lt;D&gt;</i>
<i>First&lt;com&gt; =</i>	<i>{.ID, ID, =, +=, -=, /=, *=, %=, ++, --, } }</i>
<i>follow&lt;com&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{.ID, ID, =, +=, -=, /=, *=, %=, ++, --, } }</i>

<i>&lt;D&gt; →</i>	<i>&lt;assign_opr&gt; &lt;OE&gt;:  &lt;inc_dec&gt; :</i>
<i>First&lt;D&gt; =</i>	<i>{ =, +=, -=, *=, /=, %=, ++, -- }</i>

<i>Follow</i> <D> =	{ <i>DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, c end:, jump:, this.ID: , )</i> }
<i>Selection</i> =	{ =, +=, -=, *=, /=, %=, ++, -- }

<body> →	:
<body> →	( <MST> )
<i>First</i> <body> =	{ :, ( }
<i>follow</i> <body>=	{ <i>DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) , ]</i> }
<i>selection</i> =	{ :, ( }

<MST> →	<SST><MST>
<MST> →	ε



<i>First&lt;MST&gt;=</i>	<i>{ switch, DT, public, private, void, if, ArrayList, for,while,ID,++,--, end: jump:, this.ID: , ε }</i>
<i>follow&lt;MST&gt;=</i>	<i>{ ) }</i>
<i>selection =</i>	<i>{ switch, DT, public, private, void, if, ArrayList, for,while,ID,++,--, end: jump:, this.ID:, ) }</i>

*Declaration:*

<i>&lt;dec&gt; →</i>	<i>DT ID &lt;init&gt; &lt;list&gt;</i>
<i>First&lt;dec&gt;=</i>	<i>{ DT }</i>
<i>Follow&lt;dec&gt;=</i>	<i>{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, !, [ , : , DT, switch , public ,private, void, if, ArrayList, for, while, end:., jump:., this.ID: , ) }</i>
<i>selection =</i>	<i>{ DT }</i>

<i>&lt;init&gt; →</i>	<i>&lt;OE&gt;</i>
<i>&lt;init&gt; →</i>	<i>ε</i>
<i>First&lt;init&gt;=</i>	<i>{=, ε}</i>

<i>follow</i> <init> =	{:, , }
<i>selection</i> =	{=, :, }

<list> →	:
<list> →	, ID <init> <list>
<i>First</i> <list>=	{:, , }
<i>follow</i> <list>=	{ <i>this.</i> , ID , num, deci , alpha , str, Boolean , null, ++ --, !, [ , : , DT, switch , public ,private, void, if, ArrayList, for, while, end:, jump:, <i>this.ID:</i> , ) }
<i>selection</i> =	{:, , }

Parameter List:

<PL> →	<OE> <PL2>
<i>First</i> <PL>=	{ <i>this.</i> , ID , num, deci , alpha , str, Boolean ,

	<i>null, ++ --, !, [ }</i>
<i>follow&lt;PL&gt;=</i>	<i>{ }</i>
<i>selection=</i>	<i>{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, !, [ }</i>

<i>&lt;PL2&gt; →</i>	<i>, &lt;OE&gt;&lt;PL2&gt;</i>
<i>&lt;PL2&gt; →</i>	<i>ε</i>
<i>First&lt;PL2&gt;=</i>	<i>{ , ,ε }</i>
<i>follow&lt;PL2&gt;=</i>	<i>{ }</i>
<i>selection =</i>	<i>{ , [ }</i>

<i>&lt;PL3&gt; →</i>	<i>DT ID &lt;f-list&gt;</i>
<i>First&lt;PL3&gt;=</i>	<i>{ DT }</i>
<i>follow&lt;PL3&gt; =</i>	<i>{ }</i>
<i>selection =</i>	<i>{ DT }</i>

$\langle f\text{-list} \rangle \rightarrow$	$\varepsilon \mid , DT ID \langle f\text{-list} \rangle$
$First\langle f\text{-list} \rangle =$	$\{ , , \varepsilon \}$
$follow\langle f\text{-list} \rangle =$	$\{ \}$
$selection =$	$\{ , \}$

IF-Else:

$\langle if\_else \rangle \rightarrow$	$if \langle OE \rangle \langle body \rangle \langle o\_else \rangle$
$First\langle if\_else \rangle =$	$\{ if \}$
$follow\langle if\_else \rangle =$	$\{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) \}$
$selection =$	$\{ if \}$

$\langle o\_else \rangle \rightarrow$	$else \langle body \rangle$
$\langle o\_else \rangle \rightarrow$	$\varepsilon$
$First\langle o\_else \rangle =$	$\{ else, \varepsilon \}$
$Follow\langle o\_else \rangle =$	$\{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) \}$
$Selection =$	$\{ else, DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) \}$

Object Oriented Concept:

<i>&lt;obj_dec_list&gt; →</i>	<i>ID&lt;list1&gt;</i>
<i>&lt;obj_dec_list&gt; →</i>	<i>&lt;list2&gt; :</i>
<i>First&lt;obj_dec_list&gt;=</i>	<i>{ID, =}</i>
<i>follow&lt;obj_dec_list&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{ID, =}</i>

<i>&lt; list1&gt; →</i>	<i>:</i>
<i>&lt;list1&gt; → =</i>	<i>&lt;list2&gt; :</i>
<i>&lt;list1&gt; →</i>	<i>, ID &lt;list1&gt;</i>
<i>First&lt;list1&gt;=</i>	<i>{: , =, ,}</i>
<i>follow&lt;list1&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{: , , , =}</i>

<i>&lt;list2&gt; →</i>	<i>new ID [&lt;PL&gt;]</i>
<i>&lt;list2&gt; →</i>	<i>ID</i>
<i>First&lt;list2&gt;=</i>	<i>{new , ID}</i>
<i>follow&lt;list2&gt;=</i>	<i>{ : }</i>
<i>selection =</i>	<i>{ new , ID }</i>

*Class Declaration:*

<i>&lt;Class_dec&gt; →</i>	<i>&lt;AM&gt; &lt;Abs&gt; class ID &lt;extends&gt; [&lt;class_body&gt;]</i>
<i>First&lt;class_dec&gt;=</i>	<i>{public,private,abstract,class}</i>
<i>follow&lt;class_dec&gt;=</i>	<i>{ \$,public, private, abstract, class}</i>
<i>selection =</i>	<i>{public,private,abstract,class}</i>

<i>&lt;Abs&gt; →</i>	<i>abstract</i>
<i>&lt;Abs&gt; →</i>	<i>ε</i>
<i>First&lt;Abs&gt;=</i>	<i>{abstract , ε}</i>
<i>follow&lt;Abs&gt;=</i>	<i>{class}</i>
<i>selection =</i>	<i>{abstract , class}</i>

<i>&lt;AM&gt; →</i>	<i>public</i>
<i>&lt;AM&gt; →</i>	<i>private</i>
<i>&lt;AM&gt; →</i>	$\epsilon$
<i>First&lt;AM&gt;=</i>	<i>{ public , private , <math>\epsilon</math> }</i>
<i>follow&lt;AM&gt;=</i>	<i>{ abstract, class, void, DT, ID }</i>
<i>selection =</i>	<i>{ public , private , abstract, class, void, DT, ID }</i>

<i>&lt; extends &gt; →</i>	<i>inherit ID</i>
<i>&lt; extends &gt; →</i>	$\epsilon$
<i>First&lt;extends&gt;=</i>	<i>{ inherit , <math>\epsilon</math> }</i>
<i>Follow &lt;extends&gt; =</i>	<i>{ [ ] }</i>
<i>Selection =</i>	<i>{ inherit , [ ] }</i>

<i>&lt;class_body&gt; →</i>	<i>&lt;main-method&gt;   &lt;body&gt;</i>
<i>First&lt;class_body&gt;=</i>	<i>{ main , : , ( }</i>
<i>Follow&lt;class_body&gt;=</i>	<i>{ ] }</i>
<i>Selection =</i>	<i>{ main , : , ( }</i>

<i>&lt;main-method&gt; →</i>	<i>main method [ ] &lt;body&gt;</i>
<i>first&lt;main-method&gt; =</i>	<i>{main}</i>
<i>follow&lt;main-method&gt;=</i>	<i>{ } }</i>
<i>selection =</i>	<i>{main}</i>

*Function Declaration:*

<i>&lt;func_dec&gt; →</i>	<i>&lt;AM&gt; &lt;return_type&gt; ID [&lt;PL3&gt;] (&lt;func_body&gt;)</i>
<i>First&lt;func_dec&gt;=</i>	<i>{public,private,void,DT,ID}</i>
<i>follow&lt;func_dec&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection set=</i>	<i>{public,private,void,DT,ID}</i>

<i>&lt;return_type&gt; →</i>	<i>Void</i>
<i>&lt;return_type&gt; →</i>	<i>DT</i>
<i>&lt;return_type&gt; →</i>	<i>ε</i>
<i>First&lt;return_type&gt;=</i>	<i>{ void , DT , ε }</i>
<i>follow&lt;return_type&gt;=</i>	<i>{ID}</i>
<i>selection =</i>	<i>{void , DT, ID}</i>



<i>&lt;func_body&gt; →</i>	<i>&lt;MST&gt; &lt;ret_line&gt;</i>
<i>&lt;func_body&gt; →</i>	<i>:</i>
<i>First&lt;func_body&gt;=</i>	<i>{ switch, DT, public, private, void, if, ArrayList, for,while,ID,++,--, end: jump:, this.ID:, : , ε }</i>
<i>follow&lt;func_body&gt;=</i>	<i>{ ) }</i>
<i>selection =</i>	<i>{ switch, DT, public, private, void, if, ArrayList, for,while,ID,++,--, end: jump:, this.ID:, : , ) }</i>

<i>&lt;ret_line&gt; →</i>	<i>return &lt;ret&gt;:</i>
<i>First&lt;ret_line&gt;=</i>	<i>{return}</i>
<i>follow&lt;ret_line&gt;=</i>	<i>{ ) }</i>
<i>selection =</i>	<i>{ return }</i>

<i>&lt;ret&gt; →</i>	<i>&lt;OE&gt;</i>
<i>&lt;ret&gt; →</i>	<i>E</i>
<i>First&lt;ret&gt;=</i>	<i>{this. , ID , num, deci , alpha , str, Boolean , null, ++ --, ! , ε }</i>
<i>follow&lt;ret&gt;=</i>	<i>{ : }</i>
<i>selection =</i>	<i>{ First&lt;ret&gt;={this. , ID , num, deci , alpha , str, Boolean , null, ++ --, ! , ε }</i>

X:

$\langle X \rangle \rightarrow$	$.ID\langle X \rangle \mid [\langle index \rangle] \langle X3 \rangle \mid [\langle PL \rangle] \langle X2 \rangle$
$\langle X \rangle \rightarrow$	$\varepsilon$
$First\langle X \rangle =$	$\{ .ID, [, \varepsilon \}$
$Follow\langle X \rangle =$	$\{ ++, --, :, =, +=, -=, *=, /=, \%=, ] \}$
$Selection =$	$\{ .ID, [ ++, --, :, =, +=, -=, *=, /=, \%=, ] \}$

$\langle X2 \rangle \rightarrow$	$.ID\langle X \rangle$
$First\langle X2 \rangle =$	$\{ .ID \}$
$Follow\langle X2 \rangle =$	$\{ ++, --, :, =, +=, -=, *=, /=, \%=, ] \}$
$Selection =$	$\{ .ID \}$

$\langle X3 \rangle \rightarrow$	$.ID\langle X \rangle \mid \varepsilon$
$First\langle X3 \rangle =$	$\{ .ID, \varepsilon \}$
$Follow\langle X3 \rangle =$	$\{ ++, --, :, =, +=, -=, *=, /=, \%=, ] \}$
$Selection =$	$\{ .ID, ++, --, :, =, +=, -=, *=, /=, \%=, ] \}$

<i>&lt;index&gt; →</i>	<i>int_const</i>
<i>First&lt;index&gt; =</i>	<i>{int_const}</i>
<i>Follow&lt;index&gt;=</i>	<i>{ }</i>
<i>Selection =</i>	<i>{int_const}</i>

Assign Value:

<i>&lt;assign_st&gt; →</i>	<i>ID &lt;X&gt; &lt;assign_opr&gt; &lt;OE&gt;:</i>
<i>First&lt;assign_st&gt;=</i>	<i>{ID}</i>
<i>Follow&lt;assign_st&gt;=</i>	<i>{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, !, [, : }</i>
<i>Selection =</i>	<i>{ ID }</i>

<i>&lt;assign_opr&gt; →</i>	<i>=</i>
<i>&lt;assign_opr&gt; →</i>	<i>+=</i>
<i>&lt;assign_opr&gt; →</i>	<i>-=</i>
<i>&lt;assign_opr&gt; →</i>	<i>*=</i>
<i>&lt;assign_opr&gt; →</i>	<i>/=</i>
<i>&lt;assign_opr&gt; →</i>	<i>%=</i>
<i>&lt;assign_opr&gt; →</i>	<i>=</i>
<i>&lt;assign_opr&gt; →</i>	<i>+=</i>

<i>&lt;assign_opr&gt; →</i>	<i>-=</i>
<i>&lt;assign_opr&gt; →</i>	<i>*=</i>
<i>First&lt;assign_opr&gt;=</i>	<i>{ =, +=, -=, *=, /=, %= }</i>
<i>follow&lt;assign_opr&gt; =</i>	<i>{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, !, [ }</i>
<i>Selection=</i>	<i>{=, +=, -=, *=, /=, %= }</i>

*Increment Decrement Operator:*

<i>&lt;inc_dec_st&gt; →</i>	<i>&lt;inc_dec&gt; ID &lt;X&gt;:</i>
<i>&lt;inc_dec_st&gt; →</i>	<i>ID &lt;X&gt; &lt;inc_dec&gt; :</i>
<i>First&lt;inc_dec_st&gt; =</i>	<i>{++,--,ID}</i>
<i>Follow&lt;inc_dec_st&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>Selection =</i>	<i>{++,--,ID}</i>

<i>&lt;inc_dec &gt; →</i>	<i>++</i>
<i>&lt;inc_dec &gt; →</i>	<i>--</i>
<i>First&lt;inc_dec&gt; =</i>	<i>{++,--}</i>
<i>follow&lt;inc_dec&gt; =</i>	<i>{: , ID , ] }</i>
<i>selection =</i>	<i>{++,--}</i>

Constant:

<i>&lt;const&gt; →</i>	<i>num</i>
<i>&lt;const&gt; →</i>	<i>deci</i>
<i>&lt;const&gt; →</i>	<i>alpha</i>
<i>&lt;const&gt; →</i>	<i>str</i>
<i>&lt;const&gt; →</i>	<i>Boolean</i>
<i>&lt;const&gt; →</i>	<i>Null</i>
<i>First&lt;const&gt; =</i>	<i>{num.deci,alpha,str,null,boolean}</i>
<i>follow&lt;const&gt;=</i>	<i>{ MDM , PM, ROP , AND , OR, ], : , , ,( , ) }</i>
<i>selection =</i>	<i>{num.deci,alpha,str,null,boolean}</i>

For Loop:

<i>&lt;for_st&gt; →</i>	<i>for [&lt;c1&gt;&lt;c2&gt; :&lt; c3&gt;] &lt;body&gt;</i>
<i>First&lt;for_st&gt; =</i>	<i>{for}</i>
<i>follow&lt;for_st&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{for}</i>

$\langle c1 \rangle \rightarrow$	$\langle dec \rangle$
$\langle c1 \rangle \rightarrow$	$\langle assign\_st \rangle$
$\langle c1 \rangle \rightarrow$	:
$First\langle c1 \rangle =$	$\{DT, ID, :\}$
$follow\langle c1 \rangle =$	$\{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, !, [ : \}$
$selection =$	$\{DT, ID, :\}$

$\langle c2 \rangle \rightarrow$	$\langle OE \rangle$
$\langle c2 \rangle \rightarrow$	$\epsilon$
$First\langle c2 \rangle =$	$\{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, !, [ , \epsilon \}$
$follow\langle c2 \rangle =$	$\{ : \}$
$selection =$	$\{ this. , ID , num, deci , alpha , str, Boolean , null, ++ --, !, [ , : \}$

$\langle c3 \rangle \rightarrow$	$ID \langle X \rangle \langle c4 \rangle$
$\langle c3 \rangle \rightarrow$	$\langle inc\_dec \rangle ID \langle X \rangle$
$\langle c3 \rangle \rightarrow$	$E$
$First\langle c3 \rangle =$	$\{ID, ++, --, \epsilon\}$
$follow\langle c3 \rangle =$	$\{ \}$
$selection =$	$\{ID, ++, --, \}$

<i>&lt;c4&gt; →</i>	<i>&lt;assign_opr&gt; &lt;OE&gt;</i>
<i>&lt;c4&gt; →</i>	<i>&lt;inc_dec&gt;</i>
<i>First&lt;c4&gt; =</i>	<i>{=, +=, -=, *=, /=, %=, ++, --}</i>
<i>follow&lt;c4&gt;=</i>	<i>{ } }</i>
<i>selection =</i>	<i>{=, +=, -=, *=, /=, %=, ++, --}</i>

While Loop:

<i>&lt;While_st&gt; →</i>	<i>while &lt;OE&gt; &lt;body&gt;</i>
<i>First&lt;while_st&gt; =</i>	<i>{while}</i>
<i>follow&lt;While_st&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{while}</i>

Array Declaration:

1D array:

<i>&lt;array&gt; →</i>	<i>DT ID [] &lt;arr_list&gt;</i>
<i>First&lt;array&gt; =</i>	<i>{DT}</i>
<i>follow&lt;array&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{ DT }</i>

<i>&lt;arr_list&gt; →</i>	<i>:</i>
<i>&lt;arr_list&gt; →</i>	<i>= (&lt;const&gt; &lt;arr2_list&gt;)</i>
<i>First&lt;arr_list&gt; =</i>	<i>{:,=}</i>
<i>follow&lt;arr_list&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{ :,= }</i>

<i>&lt;arr2_list&gt; →</i>	<i>, &lt;const&gt; &lt;arr2_list&gt;</i>
<i>&lt;arr2_list&gt; →</i>	<i>E</i>
<i>First&lt;arr2_list&gt; =</i>	<i>{ , , ε }</i>
<i>follow&lt;arr2_list&gt;=</i>	<i>{ ) }</i>
<i>selection =</i>	<i>{ , , ) }</i>



2D array:

<i>&lt;2d-array&gt; →</i>	<i>DT ID [] [] &lt;B1&gt;</i>
<i>First&lt;2d_array&gt;=</i>	<i>{DT}</i>
<i>follow&lt;2d-array&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{DT}</i>

<i>&lt;B1&gt; →</i>	<i>:</i>
<i>&lt;B1&gt; →</i>	<i>= ( (&lt;const&gt; &lt;B2&gt;</i>
<i>First&lt;B1&gt; =</i>	<i>{ : , =}</i>
<i>follow&lt;B1&gt; =</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{ : , =}</i>

<i>&lt;B2&gt; →</i>	<i>, &lt;const&gt; &lt;B2&gt;</i>
<i>&lt;B2&gt; →</i>	<i>) &lt;B3&gt;</i>
<i>&lt;B2&gt; →</i>	<i>) ):</i>
<i>First&lt;B2&gt; =</i>	<i>{ , , )}</i>

<i>follow</i> <B2>=	{ <i>DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , )</i> }
<i>selection</i> =	{ , , }

<B3> →	, (<const> <B2>
<i>First</i> <B3> =	{,}
<i>follow</i> <B3>=	{ <i>DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , )</i> }
<i>Selection</i> =	{,}

Array List:

<Array-L> →	<i>ArrayList (DT) ID = new ArrayList[]:</i>
<i>First</i> <Array-L> =	{ <i>ArrayList</i> }
<i>follow</i> <Array-L>=	{ <i>DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , )</i> }
<i>selection</i> =	{ <i>ArrayList</i> }

Switch Case:

<i>&lt;switch_case&gt; →</i>	<i>switch &lt;OE&gt; (&lt;case&gt;)</i>
<i>First&lt;Switch_case&gt; =</i>	<i>{switch}</i>
<i>follow&lt;switch_case&gt;=</i>	<i>{ DT, switch , public ,private, void, if, ArrayList, for, while, ID, ++, --, end:, jump:, this.ID: , ) }</i>
<i>selection =</i>	<i>{switch}</i>

<i>&lt;case&gt; →</i>	<i>opt &lt;const&gt; : &lt;body&gt; end: &lt;case&gt;   &lt;default_st&gt;</i>
<i>First&lt;case&gt; =</i>	<i>{opt, default:}</i>
<i>follow&lt;case&gt; =</i>	<i>{ ) }</i>
<i>selection =</i>	<i>{opt, default:}</i>

<i>&lt;default_st &gt; →</i>	<i>default: &lt;body&gt;</i>
<i>&lt;default_st &gt; →</i>	<i>ε</i>
<i>First&lt;default_st&gt; =</i>	<i>{ default: , ε }</i>
<i>follow&lt;default_st&gt;=</i>	<i>{ ) }</i>
<i>selection =</i>	<i>{ default: , ) }</i>