first(SST4) = { id, func }
follow(SST4) = { \n }
selection set = same as first set

first(SST5) = { func }
follow(SST5) = { \n }
selection set = same as first set

first(oe) = { int_const, char_const, str_const, float_const, (, !, inc_dec, id }
follow(oe) = { \n, :,], ;, }, ., [, inc_dec, assign_op, (}

selection set = same as first set.

first(list) = { , , ε } (→ same as first set)
follow(list) = Follow(list) ∪ follow(dec)
= { : }
selection set = first(list) − ε ∪ follow(list)
= { ε } − ε ∪ { : }
= { } ∪ { : }
= { : }

first(dec) = { dt, id }
follow(dec) = { : }
selection set = same as first set

Selection set of CFG:

first (SST) = { while, do, if, for, return, break, continue, inc_dec, id, dt, void }
follow (SST) = { \n }

selection set = same as first set.

first (SST1) = { ., [, id, func }
follow (SST1) = { \n }
selection set = same as first set

first (SST2) = { id, [, func }
follow (SST2) = { \n }
selection set = same as first set

first (SST') = {], , }
first (SST') = { \n }
selection set = same as first set.

first (SST3) = { = }
follow (SST3) = { \n }
selection set = same as first set

first (SST3') = { new, (, !, inc_dec, id, float_const, int_const, char_const, str_const }
follow (SST3') = { \n }

selection set = same as first set

same as first set

first (V_1) → { , ε }

follow (V_1) → { \n }

selection set = first (V_1) − ε ∪ follow (V_1)

= { ε } − ε ∪ { \n }

= { \n } .

first (r) → { id }

follow (r) → { \n , inc_dec , assign_op }

selection set = same as first set

<PL> → ?

first (init) = { = , ε } (same as first set)

follow (init) = first (list) = { , , ε }
= { , , ε } − { ε } ∪ follow (dec)
= { , , : } { , , \n }

selection set = first (init) − {ε} ∪ follow (init)
= { = , ε } − {ε} ∪ { , , : }
= { = } ∪ { , , : }
= { = , , , : }

first (Assign_st) = { id }
follow { Assign_st } = { \n }
selection set = same as first set

first (w) = { . , [, inc-dec, assign-op , (}

follow (w) = { \n }
selection set = same as first set

first (w1) = {] , , }
follow (w1) = { . , inc-dec , assign-op , \n }
selection set = same as first set

Production rule	First set	Follow set	Selection set.
<x> → id <x1>	{id}	{\n, inc_dec, assign_dec}	same as first set
<x1> → . id <x2> |	{.}	{\n}	same as first set
ε |	{ε}		first(x1) - ε ∪ follow(x1) = {\n}
<x3>	{[}		same as first set
<x2> → <x4> |	{(}	{\n}	same as first set
<x4> |	{.}		same as first set
ε	{ε}		first(x2) - ε ∪ follow(x2) = {\n}
<x3> → [<oe>	{[}	{\n}	same as first set
<w1> <x4>			
<x4> → (<PL>)	{(}	{\n}	same as first set
<x4>			
<x4'> → <x3> |	{[}	{\n}	same as first set
<x>	{id}		same as first set
<break_st> → break |	{break}	{\n}	same as first set
ε	{ε}		first(break_st) - ε ∪ follow(break_st) = {\n}
<continue_st> → continue |	{continue}	{\n}	same as first set
ε	{ε}		first(continue_st) - ε ∪ follow(continue_st) = {\n}
<return_st> → return	{return}	{int_cnst, float_cnst, str_cnst, char_cnst, (, !, inc_dec, id}	same as first set

<struct_def> -> <ac> struct id <n> {<n> <sbody>}	{public, private, struct}	{\n}	{public, private, struct}
<sbody> -> <ac> <static> <sbody> | <n> <sbody> | <sbody'> -> void func id(<P>)<n> {<n> <MST>} | dt <sb> | id <sb1>	{public, private, static, void, dt, id}	{}	same as first set
	{void}	{\n}	same as first set
	{dt}		same as first set
	{id}		same as first set
<sb> -> <type'> func id (<P>) <n> {<n> <MST>} | <arr> | id <init> <list>	{[}	{\n}	same as first set
<sb1> -> <arr> | (<P>) {<n> <cons_body>} | <type'> func id (<P>) <n> {<n> <MST>}	{[}	{\n}	same as first set
	{id}		same as first set
	{(}		same as first set
<ac> -> public | private | ε	{public}	{struct, static, void, dt, id}	same as first set
	{private}		same as first set
	{ε}		first(ac) - ε U follow
			{struct, static, void, dt, id}

<n'> → <oe>|ε	{int_cnst, char_cnst, str_cnst, float_cnst (, !, inc_dec, id}	{\n}	same as first
<n> → \n<n'>	{\n}	{\n}	same as first
<n'> → <n>|ε	{\n} {ε}	{\n}	same as first first<n'> - ε ∪ fol = {\n}
<MST> → ε | <st><n><MST>	{ε} {while, do, if, for, return, break, continue, inc_dec}	{}	first(MST) - ε ∪ follow same as first set
<if_else> → if <oe> : <n><body> <op_else>	{if}	{\n}	same as first
<while_st> → while <oe> : <n> <body>	{while}	{\n}	same as first set
<body> → ε | <st> <n> | {<n><MST>} <n>	{ε, {while, do, if, for, return, break, continue, inc_dec, {}	{\n, else}	first(body) ε ∪ follow(n) {\n} same as first set

<op_else> -> else <n> <body> | ε	{else}	{\n}	same as first set
	{ε}		first(op_else) - ε U follow(op_else) = {\n}
<do_while> -> do <n> {<n> <MST> } <n> while <oe> : <n>	{do}	{\n}	same as first set
<for_st> -> for e1 : e2 : e3 <n> <body>	{for}	{\n}	same as first set
<e1> -> <dec> | <Assign_st> | ε	{dt, id} {id} {ε}	{:}	same as first set same as first set first(e1) - ε U follow(e1) = {:}
<e2> -> <oe> | ε	{inc_const, char_const, str_const, float_const, (, !, inc_dec, id} {ε}	{:}	same as first set {ε}
<e3> id <e3'> | inc_dec <n> | ε	{id} {inc_dec} {ε}	{\n}	same as first set same as first set first(e3) - ε U follow(e3) = {\n}
<e3'> -> <assign_op> | inc_dec	{=} {inc_dec}	{\n}	same as first set same as first set
<assign_st> -> <n> <assign_op> <oe>	{id}	{assign_op}	same as first set
<arr_def> -> id <arr> | dt <arr>	{id} {dt}	{\n}	same as first set same as first set

<cons-body> → <Assign_st> <n> <cons_body> | ε	{id}	{\n, }}	same as first set
	{ε}		first(cons_body) - ε ∪ follow (cons-body) = {\n, }}
<S> → <defs> class id <inh> {<n> <c-body> public static void Main() {<n> <MST> } <n> <c_body> } <n> <defs>	{Acss_modifier, static, class, public, private, struct, ab_sealed}	{\n}	same as first set