**CFGs**

<S> <import\_st> void ID { } : <body> <defs>

<import\_st> €

<import\_st> import ID from ID

<import\_st> from ID import ID

<defs> <func\_def> <defs> | <class\_def> <defs> | <interface\_def> <defs> | <enums\_def> <defs> |  €

“ class or interface me header ki left factoring hogi “

<func\_def> done

<class\_def> done

<interface\_def> done

<enums\_def> done

< body > → ( <MST> <return\_st> )

<MST> 🡪 <SST> <MST> | €

<SST> 🡪 <dec\_var> | <dec\_arr> done

<SST> 🡪 <dec\_obj> | <dec\_enum>

<SST> 🡪 <dec\_dict>

<SST> 🡪 <ifelse\_st>

<SST> 🡪 <while\_st>

<SST> 🡪 flowcontrol

<SST> 🡪 <trycatch\_st>

<print\_st> handled in fncall

<input\_st> handled in fncall

<SST> 🡪 <assgn\_var> | <fn\_call> | <instof\_st> | <assgn\_arr> | <assgn\_obj> | <assgn\_dict> | <assgn\_enum>

<SST> 🡪 <throw\_st>

<dec\_var> done

<dec\_obj> done

<dec\_arr> done

<dec\_dict> done

<dec\_enum> done

<ifelse\_st> if { <OE> } : <body> <else\_if> <else>

<else\_if> Elseif { <OE> } : <body> <else\_if> | €

<else> else <body> | €

<while\_st> while { <OE> }: <body>

<instof\_st> <TS> ID <option> instanceof ID

<return\_st> → € | return <OE>

<fn-call> → <TS> ID <option> { <args\_list> }

<TS> TS . |  €

<option> → ∈ | . ID <option> | [ <OE> ] <option> | ( <args\_list> ) <option2>

<option2> → . ID <option>

<args\_list> <OE> <list\_args> | ∈

<list\_args> , <OE> <list\_args> | ∈

<trycatch\_st> try : <body> catch {ID ID} : <body> finally : <body>

<list>-->,<OE><List> | ∈

Syntax:

Assgn ke saare assignemt ki file me

<throw\_st> throw new ID { <args\_list> }

<args\_list> <OE> <list\_args> | ∈

<list\_args> , <OE> <list\_args> | ∈