**CFGs**

<S> -> <import\_st> void ID { } : <body1> <defs>

<import\_st> -> €

<import\_st> -> import ID from ID ln

<import\_st> -> from ID import ID ln

<defs> -> <func\_def> <defs>

<defs> -> class ID <extend\_st> <implement\_st> : ( <class\_body> ) <defs>

<defs> -> AM <defs2>

<defs> -> final class ID <extend\_st> <implement\_st> : ( <class\_body> ) <defs>

<defs> -> interface ID <extend\_st\_interface> : ( <interface\_body> ) <defs>

<defs> -> <enums\_def> <defs>

<defs> ->  €

<defs2> -> <ch2> class ID <extend\_st> <implement\_st> : ( <class\_body> ) <defs>

<defs2> -> interface ID <extend\_st\_interface> : ( <interface\_body> ) <defs>

<func\_def> done

<enums\_def> done

<body> -> ( <MST> <return\_st> )

<body> -> ln

<MST> -> <SST> <MST>

<MST> -> €

<SST> -> DT ID <dec1> ln

<SST> -> String ID <dec1> ln

<SST> -> dict ID <dec3> ln

<SST> -> <assgn\_var\_or\_arr> | <fn\_call> | <instof\_st> | <assgn\_obj\_or\_enum> | <assgn\_dict> | <dict\_access>

SST -> TS . ID <OPTION> <SST2>

<SST> -> ID <SST’>

<SST2> -> ln

<SST2> -> = <SST3>

<SST2> -> COMPASS <OE> ln

<SST2> -> instanceof ID ln

<SST3> -> <OE> ln

<SST3> -> new ID { <arguments>} ln

<SST’> -> <option> <SST2> | ID <dec2> ln

Cover <assgn\_var\_or\_arr> | <fn\_call> | <instof\_st> | <assgn\_obj\_or\_enum> | <assgn\_dict> | <dict\_access>

<SST> -> <ifelse\_st>

<SST> -> <while\_st>

<SST> -> flowcontrol ln

<SST> -> <trycatch\_st>

<SST> -> <throw\_st>ln

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

<else\_if> -> Elseif { <OE> } : <body> <else\_if>

<else\_if> -> €

<else> -> else <body>

<else> -> €

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

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

<return\_st> -> return <OE>

<return\_st> -> €

<TS> -> TS .

<TS> ->  €

<option> -> ∈

<option> -> . ID <option>

<option> -> [ <OE> ] <option>

<option> -> ( ID) <option>

<option> -> {<args\_list>} <option2>

<option2> -> . ID <option>

<option2> -> ln

<args\_list> -> <OE> <list\_args> | ∈

<list\_args> -> , <OE> <list\_args> | ∈

<trycatch\_st> -> try : <body1> catch {ID ID} : <body1> finally : <body1>

<body1> -> ( <MST> <return\_st> )

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> | ∈

Note:

Input, print fn\_call me handle horhe islye SST me nahi aenge