**CFGs**

<S> 🡪 <import\_st> <defs> DT 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> |  €

<func\_def> 🡪 done

<class\_def> 🡪 done

<interface\_def> 🡪 done

<enums\_def> 🡪 done

< body > → <SST> <return\_st> | ( <MST> <return\_st> )

<MST> 🡪 <SST> <MST> | €

<SST> 🡪 <dec\_var> | <dec\_obj> | <dec\_arr> | <dec\_dict> | dec\_enum | <ifelse\_st> | <while\_st> | <instof\_st> | <fn\_call> | <trycatch\_st> | flowcontrol | <print\_st> | <input\_st> | <assgn\_var> | <assgn\_arr> | <assgn\_obj> | <assgn\_dict> | <assgn\_enum> | <incdec\_st> | <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> 🡪 ID instanceof ID

<return\_st> → € | return <OE>

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

<TS> 🡪 TS . |  €

<option> → ∈ | . ID <option> | [ <exp> ] <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>

syntax of print:   
dikhao{a,b,55,7+8}

<print\_st> --> print{<OE><list>}

<list>-->,<OE><List> | ∈

<input\_st> --> input{str\_const}

Assgn ke saare assignemt ki file me

<incdec\_st> 🡪 inc-dec <TS> ID <option> | <TS> ID <option> inc-dec

<throw\_st> 🡪 throw new ID { <param-list> }