**Definitions CFGs**

Class Definiton CFG

**Note:-**

**Return statement SST me nhi rkhana or return st optional h**

<class\_def> → <class\_header> class ID <extend\_st> <implement\_st> : ( <class\_body> )

**Agar AM kch bhi nhi h to bydefault public hoga**

<class\_header> → AM final | AM | final |  €

<extend\_st> → extends ID |  €

<implement\_st> → implements ID <interface\_rec> |  €

<interface\_rec> → , ID <interface\_rec> |  €

<class\_body> → <attrb> <class\_body> | <constructor> <class\_body> | <method> <class\_body> |  €

<attrb> → <attrb&method\_header> <dec>

<dec> 🡪 <dec\_var> | <dec\_arr> | <dec\_obj> | <dec\_enum> | <dec\_dict>

<attrb&method\_header> → € | AM | static | final | AM static | AM final | static final | AM static final

<method> → <attrb&method\_header> <func\_def>

<constr> → AM ID { <params\_list> } : ( <constr\_body> )

<constr\_body> →  <pointer\_st> <MST>

<pointer\_st> → TS { <args\_list> } <pointer\_st> |  €

<args\_list> 🡪 <OE> <list\_args> | ∈

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

**Interface CFG**

**Note:-**

**Interface Defintion CFG**

<interface\_def> → AM interface ID <extend\_st> : ( <interface\_body> )

<interface\_body> → <attrb> <interface\_body> | <method> <interface\_body>

**Func def**

Syntax

Returntype ID { int a,char b, bool c }: (

SST

…

)

<func\_def> 🡪 DT ID { <params\_list> } : <body>

<func\_def> 🡪 String ID { <params\_list> } : <body>

<func\_def> 🡪 void ID { <params\_list> } : <body>

<params\_list> 🡪 <params> | €

<params> → DT ID <list>

<list> → , <params> | €

<body> → ( <MST> <return\_st> )

<SST> → <while\_st> | <dec\_st> | <if\_st> | …

<MST> 🡪 <SST>  <MST> | €

<return\_st> → € | return <OE>

**Enum Def**

Syntax

enum Direction : (

North,

South,

East,

West

)

CFG

<enum\_def> 🡪 Enum ID : ( ID <list> )

<list> 🡪 , ID <list> | €