CECC	
<u>CFGS</u>	
<structure> -&gt; <defs></defs></structure>	
<defs> -&gt; <class_def> <defs></defs></class_def></defs>	
<defs> -&gt; <interface_def> <defs></defs></interface_def></defs>	
<defs> -&gt; <abs_class_def> <defs></defs></abs_class_def></defs>	
<defs> -&gt; <enum_def> <defs></defs></enum_def></defs>	
<defs> -&gt; E</defs>	
<class_def> -&gt; <nam> class ID <inh> { <cb> }</cb></inh></nam></class_def>	
<interface_def> -&gt; interface ID <int_inh>{ <int_b> }</int_b></int_inh></interface_def>	
<pre><abs_class_def> -&gt; abstract class ID <inh> { <abs_b> }</abs_b></inh></abs_class_def></pre>	
<pre><enum_def> -&gt; enum ID { <enum_body> }</enum_body></enum_def></pre>	
<nam> -&gt; final</nam>	
<nam> -&gt; E</nam>	
<inh> -&gt; extends ID <imp></imp></inh>	
<inh> -&gt; E</inh>	
<inh> -&gt; <imp></imp></inh>	
<cb> -&gt; <func_def> <cb></cb></func_def></cb>	
<cb> -&gt; <construct_def> <cb></cb></construct_def></cb>	
<cb> -&gt; <decl> <cb></cb></decl></cb>	
<cb> -&gt; <assign_st> <cb></cb></assign_st></cb>	
<cb> -&gt; E</cb>	
<int_inh> -&gt; extends ID <id dash=""></id></int_inh>	
<int_b> -&gt; <int_var> <int_b></int_b></int_var></int_b>	
<int_b> -&gt; <abs_func_def> <int_b></int_b></abs_func_def></int_b>	
<abs_b> -&gt; <func_def> <abs_b></abs_b></func_def></abs_b>	
<abs_b> -&gt; <decl> <abs_b></abs_b></decl></abs_b>	
<abs_b> -&gt; <abs_func_def> <abs_b></abs_b></abs_func_def></abs_b>	
<abs_b> -&gt; <assign_def> <abs_b></abs_b></assign_def></abs_b>	
<abs_b> -&gt; <construct_def> <abs_b></abs_b></construct_def></abs_b>	
<abs_b> -&gt; E</abs_b>	
<pre><enum_body> -&gt; stringConst <m stringconst=""></m></enum_body></pre>	
<enum_body> -&gt; E</enum_body>	
<imp> -&gt; implements ID <id dash=""></id></imp>	
<imp> -&gt; E</imp>	
<func_def> -&gt; <fnam> <rad> def ID (<param/>)</rad></fnam></func_def>	
{ <mst> }</mst>	
<construct_def> -&gt; def ID (<param/>) { <mst> }</mst></construct_def>	
<decl> -&gt; DT <ras> <ar> &lt; int1&gt; <li>st1&gt;</li></ar></ras></decl>	
<assign_st> -&gt; ID <las></las></assign_st>	
<b> -&gt; (<args>) <fna></fna></args></b>	
<b> -&gt; [<exp>] <d2da> <ref> <eexp></eexp></ref></d2da></exp></b>	
<b>-&gt; . ID <b></b></b>	
<b> -&gt; = <exp>;</exp></b>	

		1
<fna> -&gt; ;</fna>		
<fna> -&gt; . ID <b></b></fna>		
<fna> -&gt; . IU <b></b></fna>		
.0204 [ ]		
<d2da> -&gt; [<exp>]</exp></d2da>		
<d2da> -&gt; E</d2da>		
102010 7 2		
<ref> -&gt; . ID <b></b></ref>		
<ref> -&gt; E</ref>		
<eexp> -&gt; = <exp> ;</exp></eexp>		
, ,		
<las> -&gt; = <exp> ;</exp></las>		
<las> -&gt; . ID <b></b></las>		
<las> -&gt; ID <obj r=""></obj></las>		
,		
<pre><obj r=""> -&gt; = new ID (<args>);</args></obj></pre>		
<obj r=""> -&gt; ;</obj>		
110 L		
<id dash=""> -&gt; , ID <id dash=""></id></id>		
<id dash=""> -&gt; E</id>		
THE BUSINESS E		
<int_var> -&gt; DT ID = <exp>;</exp></int_var>		
THE VALLE OF THE COMPT ,		
<abs_func_def> -&gt; abstract <rtd> def ID (<param/>);</rtd></abs_func_def>		
dbs_rane_acts s abstract (trus act is (training),		
<m stringconst=""> -&gt; , StringConst <m stringconst=""></m></m>		
<m stringconst=""> -&gt; E</m>		
<fnam> -&gt; final</fnam>		
<fnam> -&gt; static</fnam>		
<fnam> -&gt; basic</fnam>		
<rtd> -&gt; void</rtd>		
<rtd> -&gt; DT <an></an></rtd>		
<rtd> -&gt; ID</rtd>		
<an> -&gt; []</an>		
<an> -&gt; E</an>		
\AIV> E		
«Daram» » «das» «Daramii»		
<param/> -> <dec> <param1></param1></dec>		
<param/> -> <obj> <param1></param1></obj>		
<param/> -> E		
<param1> -&gt; , <dec_obj> <param1></param1></dec_obj></param1>		
<param1> -&gt; E</param1>		
N GIGINAZ / L		
<dec> -&gt; DT ID <an></an></dec>		
NUCCE / DITO NAME	<u> </u>	1

sahis s ID ID	
<obj> -&gt; ID ID</obj>	
<dec_obj> -&gt; <dec></dec></dec_obj>	
<dec_obj> -&gt; <obj></obj></dec_obj>	
<mst> -&gt; <sst> <mst></mst></sst></mst>	
<mst> -&gt; E</mst>	
(MOTA ) E	
DAG LID AMG	
<ras> -&gt; ID <mas></mas></ras>	
<mas> -&gt; (<arg>) <ref1></ref1></arg></mas>	
<mas> -&gt; [<exp>] <d2da> <ref1></ref1></d2da></exp></mas>	
<mas> -&gt; . ID <mas></mas></mas>	
<mas> -&gt; E</mas>	
<ref1> -&gt; . ID <mas></mas></ref1>	
<ref1> -&gt; E</ref1>	
<k617 -=""> E</k617>	
(1	
<ar> -&gt; [] <d2db></d2db></ar>	
<ar> -&gt; E</ar>	
<init1> -&gt; = <init1></init1></init1>	
<init1> -&gt; E</init1>	
<li><li><!--ist1--> -&gt; ;</li></li>	
<pre><li><li><li><li></li></li></li></li></pre> <pre></pre>	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
data December 1944	
<init d=""> -&gt; <ras> <init1></init1></ras></init>	
<init d=""> -&gt; <const></const></init>	
<args> -&gt; <exp> <argsd></argsd></exp></args>	
<args> -&gt; E</args>	
<argsd> -&gt; , <exp> <argsd></argsd></exp></argsd>	
<argsd> -&gt; E</argsd>	
(dig3D) / L	
COVER > CAND OD COVER	
<exp> -&gt; <and op=""> <expd></expd></and></exp>	
<expd> -&gt; II <and op=""> <expd></expd></and></expd>	
-> E	
<and op=""> -&gt; <ropop> <andopd></andopd></ropop></and>	 
<andopd> -&gt; &amp;&amp; <ropop> <andopd></andopd></ropop></andopd>	
-> E	
-/ L	

<ropop> -&gt; <e> <ropopd></ropopd></e></ropop>			
<ropop> -&gt; E</ropop>			
<e> -&gt; <t> <ed></ed></t></e>			
<e>-&gt; &lt;1&gt;<eu></eu></e>			
<ed> -&gt; PM <t> <ed></ed></t></ed>			
<ed> -&gt; E</ed>			
<t> -&gt; <f> <td></td><td></td></f></t>			
<td>-&gt; MDM <f> <td></td><td></td></f></td>	-> MDM <f> <td></td><td></td></f>		
<td> -&gt; E</td> <td></td>	-> E		
(ID>-> E			
<f> -&gt; (<exp>)</exp></f>			
<f> -&gt; <const></const></f>			
<f> -&gt; ! <f></f></f>			
<f> -&gt; <ras></ras></f>			
<sst> -&gt; <white_st></white_st></sst>			
<sst> -&gt;   <pre> </pre> <pre> <pre> <pre> </pre> <pre> </pre> <pre> <pre> <pre> </pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> <pre> </pre> <pre> &lt;</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></sst>			
<u> </u>			
<sst> -&gt; <for_st></for_st></sst>			
<sst> -&gt; <if_elif_else></if_elif_else></sst>			
<sst> -&gt; <switch_st></switch_st></sst>			
<sst> -&gt; <continue_st></continue_st></sst>			
<sst> -&gt; <return_st></return_st></sst>			
<sst> -&gt; <assign_st></assign_st></sst>			
<sst> -&gt; <decl></decl></sst>			
<sst> -&gt; <try_catch_st></try_catch_st></sst>			
1.11 / 2.40= 2			
<while_st> -&gt; while (<exp>) { <mst> }</mst></exp></while_st>			
<for_st> -&gt; for (<p1> <p2> ; <p3>) { <mst> }</mst></p3></p2></p1></for_st>			
<p1> -&gt; decl</p1>			
<p1> -&gt; assign_st</p1>			
<p1> -&gt; ;</p1>			
(D2) > (OVD)			
<p2> -&gt; <exp></exp></p2>			
<p2> -&gt; E</p2>			
<p3> -&gt; <for_assign></for_assign></p3>			
<p3> -&gt; <inc_dec></inc_dec></p3>			
<p3> -&gt; E</p3>			
<for_assign> -&gt; ID <vas></vas></for_assign>			
Ini_assignz -> ID \ANS			
MAG			
<vas> -&gt; = <exp></exp></vas>			

NACO > ID ofor D>	
<vas> -&gt; . ID <for_b></for_b></vas>	
<for_b> -&gt; (<args>) <fnb></fnb></args></for_b>	
<for_b> -&gt; [<exp>] <d2da> <ref2> <etexp></etexp></ref2></d2da></exp></for_b>	
<for_b> -&gt; . ID <for_b></for_b></for_b>	
<for_b> -&gt; = <exp></exp></for_b>	
<fnb> -&gt; . ID <for_b></for_b></fnb>	
_	
<ref2> -&gt; . ID <for_b></for_b></ref2>	
<ref2> -&gt; E</ref2>	
NCIZZ Z E	
<etexp> -&gt; = <exp></exp></etexp>	
SELEXITY SCAP	
<continue-st> -&gt; continue;</continue-st>	
Sometime services and the services of the serv	
<pre><break_st> -&gt; break;</break_st></pre>	
2.53.,	
<return_st> -&gt; return <expn> ;</expn></return_st>	
,	
<expn> -&gt; <exp></exp></expn>	
<expn> -&gt; E</expn>	
CAPINA A L	
<if_else_elif> -&gt; if (<exp>) {<mst>} <elif_st></elif_st></mst></exp></if_else_elif>	
<else_st></else_st>	
<pre><elif_st> -&gt; elif (<exp>) { <mst> } <elif_st></elif_st></mst></exp></elif_st></pre>	
<elif_st> -&gt; E</elif_st>	
(also at a place (all ACTs.)	
<else_st> -&gt; else : {<mst>}</mst></else_st>	
<else_st> -&gt; E</else_st>	
<pre><switch_st> -&gt; switch (<switch_in>) { <switch_b> }</switch_b></switch_in></switch_st></pre>	
<switch_in> -&gt; ID</switch_in>	
<switch_in> -&gt; <const></const></switch_in>	
<pre><switch_b> -&gt; <condition> { <mst> } <switch_b></switch_b></mst></condition></switch_b></pre>	
<pre><switch b=""> -&gt; E</switch></pre>	
SWIGHT DY YE	
<condition> -&gt; case <const> :</const></condition>	
<pre><condition> -&gt; default :</condition></pre>	
Condition -> deladit.	
<try_catch_st> -&gt; try { <mst> } <otc></otc></mst></try_catch_st>	
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
COTC > catch (exception ID) ( MST ) COTCD	
<otc> -&gt; catch (exception ID) {<mst>} <otfd></otfd></mst></otc>	
<0TC>-> <0TF>	

<otf> -&gt; finally { <mst> }</mst></otf>	
<otfd> -&gt; finally { <mst> }</mst></otfd>	
<otfd> -&gt; E</otfd>	