𝑃𝑟𝑜𝑔𝑟𝑎𝑚→𝑑𝑡 𝑖𝑑(𝑃𝑎𝑟𝑎𝑚𝐿𝑖𝑠𝑡){𝑆𝑡𝑚𝑡𝑠}   
𝑃𝑎𝑟𝑎𝑚𝐿𝑖𝑠𝑡→𝑑𝑡 𝑖𝑑 𝑃𝐿𝑖𝑠𝑡   
𝑃𝐿𝑖𝑠𝑡→ ,𝑑𝑡 𝑖𝑑 𝑃𝐿𝑖𝑠𝑡 | ∈   
𝐷𝑒𝑐𝑆𝑡𝑚𝑡𝑠→𝑑𝑡 𝑖𝑑 𝑂𝑝𝑡𝑖𝑜𝑛𝑎𝑙𝐴𝑠𝑠𝑖𝑔𝑛 𝐿𝑖𝑠𝑡   
𝐿𝑖𝑠𝑡→ ,𝑑𝑡 𝑂𝑝𝑡𝑖𝑜𝑛𝑎𝑙𝐴𝑠𝑠𝑖𝑔𝑛 𝐿𝑖𝑠𝑡 | ∈   
𝑂𝑝𝑡𝑖𝑜𝑛𝑎𝑙𝐴𝑠𝑠𝑖𝑔𝑛→ =𝐸𝑥𝑝𝑟 ;| ∈   
𝐸𝑥𝑝𝑟→*TE’*  
*E′ → +TE′| ε   
T → FT′   
T’ →\* FT′ | ε   
F → (Expr)| id*𝐹𝑜𝑟𝑆𝑡𝑚𝑡→𝑓𝑜𝑟 ( 𝑇𝑦𝑝𝑒 𝑖𝑑 𝐸𝑥𝑝𝑟 ; 𝐸𝑥𝑝𝑟 𝑟𝑒𝑙𝑜𝑝 𝐸𝑥𝑝𝑟;𝑖𝑑++) { 𝑆𝑡𝑚𝑡𝑠}   
𝑇𝑦𝑝𝑒→𝑑𝑡 | ∈   
𝐼𝑓𝑆𝑡𝑚𝑡→𝑖𝑓 (𝐸𝑥𝑝𝑟 𝑟𝑒𝑙𝑜𝑝 𝐸𝑥𝑝𝑟) { 𝑆𝑡𝑚𝑡𝑠 } 𝑂𝑝𝑡𝑖𝑜𝑛𝑎𝑙𝐸𝑙𝑠𝑒  
𝑂𝑝𝑡𝑖𝑜𝑛𝑎𝑙𝐸𝑙𝑠𝑒→𝑒𝑙𝑠𝑒 { 𝑆𝑡𝑚𝑡𝑠 } | ∈   
𝑅𝑒𝑡𝑢𝑟𝑛𝑆𝑡𝑚𝑡→𝑟𝑒𝑡𝑢𝑟𝑛 𝐸𝑥𝑝𝑟;

As it can be seen there is no need for left recursion elimination or left factoring required in the grammar. Now we need to confirm whether this grammar is LL(1) or not.

Computing FIRST sets for each non-terminal

1. FIRST(Program) = {dt}
2. FIRST(ParamList) = {dt}
3. FIRST(PList) ={**,** ∈}
4. FIRST(Stmts) =FIRST(Stmts’)= FIRST( U FIRST( U FIRST( U FIRST( U FIRST(Return U ∈

FIRST(Stmts) = {dt} U {if} U {for} U {id} U {return} U ∈ = {dt, if, for, return, id, ∈}

1. FIRST(Stmts’) = {dt, if, for, return, id, ∈}
2. FIRST(DecStmts) = {dt}
3. FIRST(List) = FIRST(,𝑑𝑡 𝑂𝑝𝑡𝑖𝑜𝑛𝑎𝑙𝐴𝑠𝑠𝑖𝑔𝑛 𝐿𝑖𝑠𝑡) U ∈ = {, ∈}
4. FIRST(OptionalAssign) = FIRST(=𝐸𝑥𝑝𝑟; ) U ∈ = {=, ∈}
5. FIRST(Expr) =FIRST(TE’) = FIRST(FT’) =FIRST(F) = {(, id}
6. FIRST(E’)= FIRST(*+TE′) U ε = {+, ε}*
7. FIRST(T) = FIRST(FT’) = FIRST(F) = {(, id}
8. FIRST(T’) = FIRST(*\* FT′) U ε = {\*, ε}*
9. FIRST(F) = {(, id}
10. *FIRST(ForStmt) = {for}*
11. *FIRST(Type) = {dt,* ∈}
12. *FIRST(IfStmt)={if}*
13. *FIRST(AssignStmt) = {id}*
14. *FIRST(OptionalElse) = {else,* ∈}
15. *FIRST(ReturnStmt) = {return}*

*Computing follow sets for every production*

1. *FOLLOW(Program) = {$}*
2. *FOLLOW(ParamList) = { ) }*
3. *FOLLOW(PList) = FOLLOW(ParamList) = { ) }*
4. *FOLLOW(Stmts) = {* ***}*** *}*
5. *FOLLOW(Stmts’} = FOLLOW(Stmts) = {* ***}*** *}*
6. *FOLLOW(DecStmts) = (FIRST(Stmts’) -* ∈) U FOLLOW(Stmts’) U FOLLOW(Stmts) *=* {dt, if, for, id, return, **}** }
7. FOLLOW(List) = *FOLLOW(DecStmts) =*{dt, if, for, return, id,**}** }
8. FOLLOW(OptionalAssign) = (FIRST(List)- ∈) U FOLLOW(List) U FOLLOW(DecStmts) = {**,** , dt, if, for, return, id, **}**}
9. FOLLOW(Expr) = {**;** relop, **)**}
10. FOLLOW(E’)= FOLLOW(Expr) = {**;** relop, **)**}
11. FOLLOW(T) = FIRST(E’)- *ε* U FOLLOW(Expr) = {**+**, **;**, relop, **)**}
12. FOLLOW(T’) = FOLLOW(*T) =* {**+**, **;**, relop, **)**}
13. FOLLOW(F) = (FIRST(T’) – *ε) U*  FOLLOW(T) U FOLLOW(T’)= {\*, **+**, **;**, relop, **)**}
14. *FOLLOW(ForStmt) = FIRST(Stmts’) -* ∈ U FOLLOW(Stmts’)*=* {dt, if, for, return, id, **}** }
15. *FOLLOW(Type) = {id*}
16. *FOLLOW(IfStmt)= FIRST(Stmts’) -* ∈ U FOLLOW(Stmts’)*=* {dt, if, for, return, id, **}**}
17. *FOLLOW(AssignStmt) = FIRST(Stmts’) -* ∈ U FOLLOW(Stmts’)*=* {dt, if, for, return, id, **}**}
18. *FOLLOW(OptionalElse) = FOLLOW(IfStmt) =* {dt, if, for, return, id, **}**}
19. *FOLLOW(ReturnStmt) = FIRST(Stmts’)-* ∈ U FOLLOW(Stmts’) *=* )*=* {dt, if, for, id, return, **}** }