**Qs 5. Explain Semantic Analysis in mini–C Compiler?**

Three sorts of static tests are handled by the semantic analyzer.

● Variables must be stated before they may be used. The check declaration function is used to check if

the identifier supplied as an argument is available in the symbol table. If it isn't, a helpful error

message is displayed. When an identifier is met in a statement that isn't a declarative statement, the

check declaration function is invoked.

● It is not possible to redeclare variables. Variables cannot be redeclared even throughout loops since

our compiler expects a single scope. The add method has been modified to verify if the symbol is

already existing in the symbol table before inserting it. Error message is printed in case of

redeclaration.



● Variables in an arithmetic expression are type checked. Nothing is done if the types match. When a

variable has to be changed to a different type, a type conversion node is added to the syntax tree. To

keep track of the type of complex expressions that aren't in the symbol table, the type field is added

to the struct representing value and expression tokens. The annotated syntax tree is the result of this

phase

