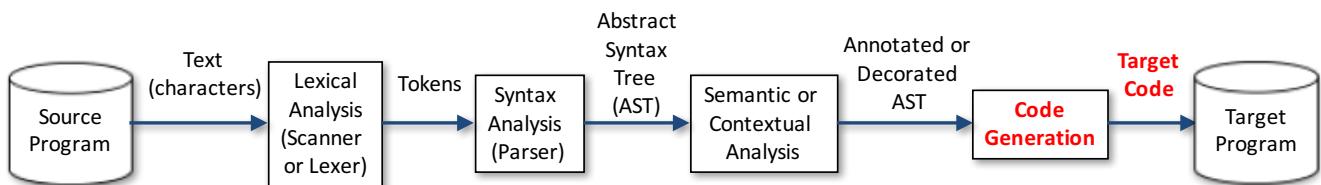


## Laboratory 12 – Code Generation II

The objective of this lab is to finish our C-- compiler, generating code for the remaining AST nodes:

- While and if statements (remember that, as in the C language, the else body is optional).
- The [ ] indexing operation for arrays (multi-dimensional arrays must be supported).
- Function invocation.
- The return statement.
- Definition of global and local variables.
- Use of local variables and parameters as expressions.



### Description

Recall you have to implement three code functions: execute for program, definitions and statements; value for expressions; and address for l-value expressions. The associated visitors are `ExecuteCGVisitor`, `ValueCGVisitor` and `AddressCGVisitor`.

Two example input (C--) files and their corresponding output (MAPL) programs are provided. Analyze them to figure out the code generation templates you must define. Although the output code generated by your compiler could be different from the one provided, the target MAPL code generated by your compiler must have the same semantics as the input program.

### Go ahead

- Define the remaining code templates for the three code functions.
- Implement the code templates in the three code generation visitors.
- Make sure the generated programs are correct by running them with the MAPL VM and checking its runtime semantics.
- Make sure you have implemented all the C-- language correctly by compiling and running different example programs.