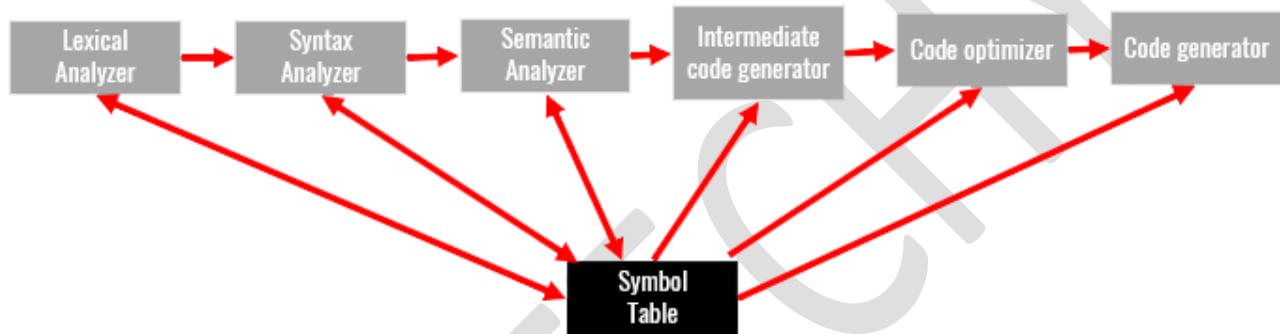


Symbol Table in Compiler Design

- Symbol tables are data structures that are used by compilers to hold information about source-program constructs. The information is collected incrementally by the analysis phases of a compiler and used by the synthesis phases to generate the target code.



- The symbol table, which stores information about the entire source program, is used by all phases of the compiler.
- An essential function of a compiler is to record the variable names used in the source program and collect information about various attributes of each name.
- These attributes may provide information about the storage allocated for a name, its type, its scope.
- A symbol table can be implemented in one of the following ways:
 - Linear (sorted or unsorted) list
 - Binary Search Tree
 - Hash table

- Among the above all, symbol tables are mostly implemented as hash tables, where the source code symbol itself is treated as a key for the hash function and the return value is the information about the symbol.
- A symbol table may serve the following purposes depending upon the language in hand:
 - To store the names of all entities in a structured form at one place.
 - To verify if a variable has been declared.
 - To implement type checking, by verifying assignments and expressions.
 - To determine the scope of a name (scope resolution).

For More Details Click Here:

<https://www.wikitechy.com/tutorials/compiler-design/symbol-table-in-compiler-design>

