**Symbol Table:**

The compiler creates and maintains a data structure to store information about the occurrence of various entities such as variable and function names, objects and classes is known as Symbol table.

Items stored in symbol table:

Variable names and constants, Procedure and function names, Literal constants and strings, Compiler generated temporaries, Labels in source languages.

Symbol table can be implemented in one of the following ways:

Linear (sorted or unsorted) list

Binary Search Tree

Hash table

And other ways.

**Implementation of symbol table using hash table:**

In hashing two tables are maintained a hash table and a symbol table and are most used method to implement symbol tables.

A hash table is an array with an index range 0 to table size -1. These entries are pointers pointing to names/labels of the symbol table.

Insertion and deleting are very fast in hash table. The time complexity is O(1).

Insert() operation is used to store values in hash table. When a new value is stored in hash table, it is assigned an index number. The index is calculated using hash table.

Search() operation helps us to find particular stored data from the hash table. To search for a name/label we use a hash function that will result in an integer between 0 to table size-1.

Display() operation is used display the stored data of hash table using the index values assigned to the stored data.

Exit() operation will terminate the whole process.

The advantage is quick to search is possible and the disadvantage is that hashing is complicated to implement.