v) As the number of symbols appearing in a program vary to a great extent, the symbol table should be able to grow as symbols are added to it.

vi) It must support duplicate entries. In most of programming languages, a variable at an inner nesting level can have the same name as a variable at an outer level. Inspite of having the same name as a variable at an outer nesting level, a distinct symbol table entry is required for each of these variables as these are different variables. The scoping rules of the language determines which of these variables are active at a given time.

vii) The symbol table is the most important and most complex data structure in the compiler. Therefore, the underlying data structure and its support functions must be organized in such a fashion that someone other than the compiler writer can maintain them if required.

## Symbol Table Organization (or Symbol-Table entries) :

Symbol table contains an entry for each name in the source program. We can implement each entry as a record consisting of a sequence of consecutive words of memory. The associated information saved about a name depends on the usage of that name so the format of these entries cannot be unique. To make it uniform, we can move some of the information about a name outside the table entry and store a pointer to this information in the record.

[P.T.O.]