

CSE310

Compiler Sessional Lab

Assignment 01: Construction of a Symbol-Table

The target of this sessional is to implement a primitive compiler. As the first step towards the development, we will build a symbol-table in this assignment. At this initial stage of the task, we will omit many details regarding an actual symbol-table and we will simply adhere to the basic concept. A symbol-table is an efficient data-dictionary for the symbols used in a program. Thus, our focus in this assignment is to construct a simple hash-based data-dictionary based on chaining.

Input:

The input to your program will be a sequence of two-tuples, where each element in each tuple is a string. An example of input sequence is given below.

- int, KEYWORD
- myFunction, FUNCTION
- x, IDENTIFIER
- 111192019, NUMBER
- (, PARENTHESIS

The first element of each tuple will be the key of the record to be stored in the symbol-table. Hence, you have to apply the hash function on the first element of each tuple.

Implementation:

Implement the following two classes:

- class SymbolInfo: The definition of this class will grow gradually throughout the development of this project. For this assignment, we simply need two members, one for storing the symbol (e.g. “x”) and another for storing the type of the symbol (e.g. “IDENTIFIER”).
- class SymbolTable: Since our symbol-table will be a hash-table based on chaining, we will have to start with an array of pointers where each pointer points to a list of nodes of type class SymbolInfo. class SymbolTable will have such an array of pointers. For this assignment, the choice of the size of this array, as well as of the hash function is left upto you(20 can be preferable). In addition to this array of pointers, class SymbolTable will have three methods for the following purposes:
 - insert(): to insert a new symbol along with its type into the symbol table.

- lookup(): to lookup whether a given symbol already exists in the symbol table or not
- dump(): to dump the contents of the symbol table to the console.

Important Notes:

You have to obey the following guidelines for all the assignments in this course,

- Use dynamic memory allocation instead of hard-coding technique
- Install a Linux environment and develop the whole project in it
- Write explanatory comments and design extensive test cases
- Maintain individual authorship, **plagiarism detection software will be used** to screen assignments; any kind of cheating will possess **severe consequences** including but may not be limited to **F grade**.

Submission:

You have to submit in next sessional class **only**. Submissions after the deadline will not be accepted.

Evaluation:

- Offline: 40% marks are allocated for offline assignment
- Online: 60% marks are allocated for **online** assignment, a modification of the assignment will be directed in the class and a student should be able to modify during the class