CSE 310-July 2016

Assignment 1

Department of CSE, BUET August 28, 2016

1 Introduction

The purpose of this course is to construct a very simple compiler. In the first step to do so, we are going to implement a symbol-table. A *symbol-table* is a data structure maintained by compilers in order to store information about the occurrence of various entities such as identifiers, objects, function names etc. Information of different entities may include type, value, scope etc. At the starting phase of constructing a compiler, we will not go into many detail. We will simply construct a symbol-table based on hashing where collision is resolved by chaining. Fig 1 illustrates a sample symbol table.

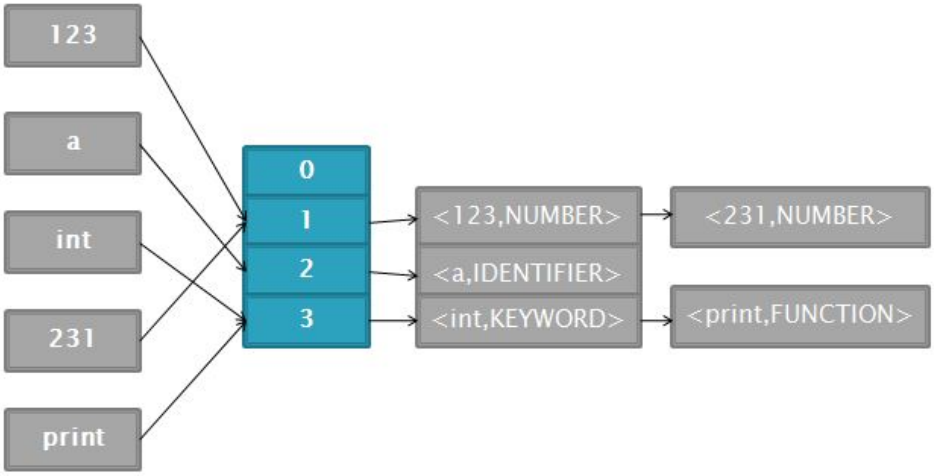

Figure 1: Symbol Table using hashing

2 Tasks

You have to implement following two classes.

- **SymbolInfo:** This class contains the information regarding a symbol faced in the source program. In the first step we will limit ourselves to only two members. One is for the **Name** of the symbol and other is the **Type** of the symbol. But remember, this class may extend as we progress to develop our compiler.
- **SymbolTable:** This class implemets the hash table. You may need an array (array of pointers) and a hash function (decent one). Then you need to implement following four functions.

Sample Input	Sample Output
I 123 NUMBER	Inserted at position 2,0
I 231 NUMBER	Inserted at position 2,1
I a IDENTIFIER	Inserted at porsition 1,0
P	0 –> 1 –> <a, IDENTIFIER> 2 –> <123,NUMBER> <231, NUMBER> 4 –>
L 123	Found at 2,0
D 123	Deleted from 2,0

Table 1: Sample Input Output

- **Insert:** Insert into symbol table if already not inserted.
- **Look up:** Search the hash table for particular symbol.
- **Delete:** Delete an entry from the symbol table.
- **Print:** Print the whole symbol table in the console.

3 Input

Each line of the input will start with a code letter indicating the operation you want to perform. The letters will be among 'I', 'L', 'D' and 'P'. 'I' stands for insert which is followed by two space separated strings where the first one is symbol name and the second one is symbol type. As you might already guessed symbol name will be the key of the record to be stored in the symbol-table. 'L' means lookup which is followed by a string containing the symbol to be looked up in the table. 'D' stands for delete which is also followed by a string to be deleted. At last 'P' stands for print the symbol table.

4 Output

For insert command you have to print the array index of the hash table as well as the index of the chain where the symbol is placed by your program. If your program fail to insert then it should print reason of failure. For lookup or delete you have to print the position of look upped or deleted element just like the insert command if you can carry out the command, otherwise show appropriate error message. At last for print command, print the whole symbol table (you may skip the empty indexes). See Table 1 for more clarification. Check the sample io given in moodle and try to follow it.

5 Important Notes

Please try to follow the instructions listed below while implementing your assignment:

- Implement using C++ programming language
- Avoid hard coding

- Use dynamic memory allocation
- Take input from file. You may output both in console and file.
- Try to get accustomed to a Linux platform

6 Rules

- You have to submit all your source code via moodle. All the file name will be in following format

  <your student id>_offline1_<your file name>

  For example, my submitted file would look like 1005004_offline1_symboltable.cpp
- Any type of plagiarism is strongly forbidden and may result in severe punishments.
- Prepare for an online evaluation.

7 Deadline

Deadline is set at 9:00 am, September 4, 2016 for all lab groups.