Table

Programmer Manual
Table

1. Problem Description

The Table class consists of a mapping function (for details see the Mappings programmer manual) which maps key value pairs to an appropriate index in the Table and stores them in an array. These pairs can then be looked up later for fast access. This table with the mapping function is used to run a lock simulation program (for details see the overall program's programmer manual).

2. Class Table

Private data members:

int tableSize	the amount of data the table can hold
Pair* the_table	the pairs inserted into the Table

Private member functions:

int (*Mapping)	mapping function to assign locations to a pair

Public member functions:

Table	constructor for a table object
Table	copy constructor for a table object
print	prints the data in the table
insert	inserts a pair into the table
remove	removes a pair from the table
lookUp	tries to find a given pair in the table
operator=	overloaded equality operator
empty	determines if a table has no data in it
full	determines if a table can hold no more data
size	returns the amount of data in the table
isIn	determines whether a given pair is in the table
loadTable	reads data from a file into the table
lock	runs a lock simulation using tables to look up states in a finite state machine

3. High Level Program Solution

Table constructor
sets the mapping function to the function defined in Mapping.h
sets the table size
creates an array of pairs
initializes all of the pairs in the array to empty

Table copy constructor
sets the argument table to the current table

empty
looks through each index location in the table
if one of the pairs is not empty, the table is not empty
otherwise it is empty

operator=
sets the argument table's mapping function and table size to the current table
creates an array of pairs the same size as the current table
copies all of the pairs in the current table to the argument table

full
looks through each index location in the table
if one of the pairs is empty, the table is not full
otherwise it is full

size
sets the size to 0
looks through each index location in the table
if that location is not empty, increment the size
return the final size of the table

isIn
set the index value to the location of the passed in key using the mapping function
if that key is in the table, the key has been found
otherwise return -1

insert
look up the given pair and if that location is unavailable, return false
if the table is full, return false
set the index to the given pair using the mapping function
if that index location is available, put that pair into that location and return true
otherwise a pair already exists in that location and return false

remove
if the table is empty, return false
find the key to remove
set the index location of that pair to empty and return true
otherwise the pair was not found and return false

print
loops through each index location in the table
print out the data in each pair

loadTable
open up a file to get the data from
put each piece of data into the correct part of a pair
insert this pair into the table

lock

set the initial state to nke
prompt the user to input the password
make a pair out of the input character and the state given by the table
continue until 4 characters have been successfully entered
if the password was incorrect and the number of attempts is less than 3, prompt for the password again
if the password was incorrect and the number of attempts is 3, sound the alarm and exit
if the password was correct, unlock