//implementation file for the hash table itself, aka address book

#include "Address Book.h"

Client\_Address\_Book::Client\_Address\_Book(){

//default constructor will read data from input file "client\_address\_data.txt".

}

Client\_Address\_Book::Client\_Address\_Book(const Client\_Address\_Book &){

//Copy Constructor

}

Client\_Address\_Book::~Client\_Address\_Book(){

//Destructor

}

void Client\_Address\_Book::Insert(const string & s){

//Insert adds a new Client's information to the hash table

}

void Client\_Address\_Book::Remove(const string & s){

//Remove deletes a client from the hash table if it is there; otherwise a message should be printed stating so.

}

void Client\_Address\_Book::Update(const string & s){

//update record; see example below

}

void Client\_Address\_Book::Print\_BST(const string & s){

//Print a BST (cell in hash table) inorder to the screen

}

void Client\_Address\_Book::Print\_Hash\_Table(){

"Inside Client\_Address\_Book Print\_Hash\_Table\n";

//function will print hash table to the screen

}

void Client\_Address\_Book::Print\_Hash\_Table\_to\_File(const string & filename){

//function will print hash table to output file

}

bool \* Client\_Address\_Book::Search(const string & s){

"Inside Client\_Address\_Book Search\n"; return 0;

//return true if client found; otherwise false

}

unsigned int Client\_Address\_Book::Hash\_Function(const string & s){

//return the index of the BST in the hash table

}

//Hint:Remember that the insert, remove and search function for Clients\_Address\_Book will use Client\_Info\_BST’s insert, remove and search respectively.