

Introduction

We will use our `Hashtable` class to create an “atlas” of the 50 states of the United States. For each state, we will store its two-letter abbreviation, name, capital city, population, number of representatives in the House of Representatives, the date when it was admitted to the union, and its area (in mi^2). See the file `States.txt`. The main program will display a menu with the following menu items:

- (L) List all states – Display a list of the names of the states, in alphabetical order.
- (F) Find a state – Given its two-letter abbreviation, find and display the state’s data.
- (I) Insert a state – Add a new state to the atlas by providing its abbreviation, name, capital city, population, number of representatives, date when admitted to the union, and area.
- (D) Delete a state – Given its abbreviation, delete that state from the atlas.
- (E) Edit a state’s data – Given a state’s abbreviation, locate that state’s record and replace the old values with the new ones.
- (A) Display all states and their data.
- (H) Dump the hash table – Display the raw contents of the hash table. Display them as one bucket per line. Each bucket should be displayed as a list of records. This might be handy for debugging.
- (Q) Quit – Exit the program.

The Menu Class

Your application program will be menu-driven. I will provide the `Menu` class. See the document `Menu Class.pdf` for details. It is easy to use. Once you create a menu, you may add any number of menu items to it by using the `addItem()` member function, with prototype

```
void addItem(string item, char shortcut);
```

The parameter `item` is the text to be displayed in the menu. The parameter `shortcut` is a single character which will serve as a keyboard shortcut to select the item.

The `Menu` function `getChoice()`, with prototype

```
char getChoice();
```

will read one character from the keyboard, verify that it is valid, and return that character.

The simplest way to handle menu selections is to use a `while` loop structured as follows.

```
cout << main_menu << endl;
char choice = main_menu.getChoice();
while (choice != 'Q')
{
    switch (choice)
    {
        case 'L': // List the states
        case 'F': // Find a state
            :
    }
}
```

The Application Program

The application program, named `Atlas.cpp`, will begin by reading the states' data from the file `States.txt` into the hash table. Then it will create the main menu, as described above. The user is presented with the main menu. When he makes a choice, the application will act on that choice. In the case of the Edit choice, the program will display a submenu of edit choices, as follows.

```
(B) Edit the abbreviation
(N) Edit the name
(C) Edit the capital
(P) Edit the population
(R) Edit the number of representatives
(D) Edit the date of admittance
(A) Edit the area
(Q) Quit
```

The user will be presented with this menu to make any number of changes, until he choose "Quit."

When you are finished, turn in the files `arraylist.h`, `linkedlistnode.h`, `linkedlist.h`, `hashtable.h`, `hashtableentry.h`, `menu.h`, `menu.cpp`, `state.h`, and `state.cpp`. You work will be due by 5:00pm on Friday, March 23.