

CSC331: Data Structures - BMCC Spring 2022

Professor Byron

Program #2 specifications: Linked list

Due: 1159pm Mon 3-21-2022

No credit received for late submission

Your task for this assignment is to implement a linked list data structure in C++.

1. Implement a transaction-based linked list data structure using a C++ object. The program will be interactive. A transaction will be entered at the command line after a short prompt and output will be displayed on the console display. A batch of input transactions in a file can be processed using redirection.
2. A linked list of names and birth years will be created, updated and managed using Add, Delete and List transactions. Any non-blank name is valid. The birth year must be in the range from 1900 to 2022. Transaction types will be processed as follows:
 - Add – To add a name and birth year to the list, enter a transaction in the form of “A”, space, name, space, birth year. For example: “A Paco 1988”. If the new name is not on the list and the birth year is valid, the name and birth year will be added. When a new name is added, the message “<name> added” will be displayed (showing the name added). If the name is already on the list or if the birth year is invalid, the name will not be added, and the message “<name> not added” will be displayed.
 - Delete – To delete a name and birth year from the list, enter a transaction in the form of “D”, space, name. For example: “D Ling”. If the name is not on the list, the message “<name> not found” will be displayed. If the name is on the list, the name and birth year will be removed and the message “<name> deleted” will be displayed.
 - List – To display the names and birth years in a numbered list, enter a transaction in the form of “L”. Each name in the list will be displayed with a sequential number and the respective birth year on a line by itself.
 - Quit – To terminate the program, enter a transaction in the form of “Q”.

In this assignment, use the partially completed class called personList.h. You are to complete the methods add, exists and remove.

3. Here is sample dialogue of the running program:

```
$ prog2
enter transaction: A Joan 2004
Joan added
```

```
enter transaction: A Manuel 2004
Manuel added
enter transaction: A Abdul 3034
Abdul not added
enter transaction: A Joan 2000
Joan not added
enter transaction: L
1. Joan 2004
2. Manuel 2004
Enter transaction: D Cho
Cho not deleted
Enter transaction: Q
$
```

A sample file of transactions, such as prog2.dat, may contain the following:

```
A Joan 2004
A Manuel 2004
A Abdul 2024
A Joan 2000
L
D Cho
Q
```

Using redirection, the program will be run as follows:

```
$ prog2 < prog2.dat
```

The output produced will be similar to the output shown above with redirection.

4. Your C++ program file should be named csc331_prog2_lastname.cpp. Your program should contain comments starting on line 1 of the program containing the following information:
 - a. course ID and section
 - b. your full name
 - c. the program file name
 - d. the program assignment number and due date
 - e. the program purpose
 - f. special instructions to create an executable program, if needed

You are encouraged to add additional comments throughout the program that you feel might be helpful to the reader of your source code.

5. Submit your C++ program source file (i.e., cpp file) and modified personList header file as attachments to an email message to kbyron@bmcc.cuny.edu using a subject in this form: "csc331_prog2_lastname". Do your own work. Students submitting copies of the same program and/or header file will receive grades of zero for the assignment.

6. Grading rubric

Partial or extra credit will be awarded as follows:

50%: significant logic with compiling errors
60%: significant logic with no compiling errors
90%: list and quit commands working
100%: add, list and quit commands working
110%: add, list, quit and delete commands working