2100.021 - Program 2 - Linked List

My Solutions

Write a C++ program that implements a **singly linked list** list data structure for C++ strings. Your list must have the following functionality: insert, delete, clear, search, print and getsize.

Once the program is started, it will print out the promt "list**>** " (> is followed by a whitespace):

```
g++ *.cpp
./a.out

list>
```

You will implement the following commands:

insert

Insert takes a single argument, which is a string. Once the list node information has been entered, the prompt is repeated. Make sure to always insert at the end of the list.

```
list> insert someword
list> insert something
list> insert however
list>
```

delete

Delete takes a single argument, which is a string. Once the list node information has been entered, the prompt is repeated. If the word that is to be deleted is not in the list, you must print out an error message starting with "Error!".

```
list> delete someword
list> delete hello
Error! Element does not exist.
list>
```

clear

Clear takes no arguments. It removes all the elements from the list and repeats the prompt.

```
list> clear
list>
```

search

Search takes a single argument, which is a string. Print out "true" or "false" depending on the outcome of the search and repeat the prompt.

```
list> search something
true
list> search hi
false
list>
```

print

Print takes no arguments. The elements of the list are printed separated by hyphens. Once the list has been printed, repeat the prompt. If the list is empty, you must print out an error message starting with "Error!".

```
list> print
something-however
list> clear
list> print
Error! The list is empty.
list>
```

getsize

Getsize takes no arguments. Print out the number of elements in the list and repeat the prompt.

```
list> getsize
0
list>
```

quit

Exit the program.

```
list> quit
```

Your program must be "fool-proof", i.e. you can not rely on the user typing valid commands or arguments. For any "user-mistake", output an error message starting with "Error!".

Submit at least the following files:

- Your main class file (e.g Main.cpp) controling the flow of the program
- The prototype for your node class
- The implementation of your node class
- The prototype for your list class
- The implementation of your list class

Example of program execution:

```
g++ *.cpp

./a.out

list> insert someword
list> insert something
list> insert however
list> print
someword-something-however
list> getsize
3
```

```
list> delete someword
list> delete hello
Error! Element does not exist.
list>  getsize
2
list> print
something-however
list> search something
true
list> search hi
false
list> clear
list> print
Error! The list is empty.
list> getsize
0
list> quit
```

Grading

Your program will be judged on the following:

- 45% - Passes I/O requirements
- 40% - Code satisfies requirements of assignment
- 15% - Professional coding style
  - 5% Adequate comments
  - 5% Modularity (small main function, separate functions, etc)
  - 5% Readability (line length, indentation, variable names)

As with all other programming assignments in this class, your program will receive a 0 if it does not compile. If it does not compile on Praktomat it MUST compile on cse01.cse.unt.edu.