CS 240

Data Structures and Algorithms

Fall 2013

1 Lab 05

For today's lab assignment, you will be validating your Node class is of the correct form and then submitting. Afterwards you are to begin work on the pre-lab for next week.

2 Assignment

The Node class should have the following methods.

- Default Constructor – named `Node`, takes no parameters, makes no allocations
- Explicit Value Constructor – named `Node`, takes an Element by value, allocates space for it, stores that value in the allocated space
- Copy Constructor –named `Node`, takes an Element by reference, copies the node and all nodes connected to it
- Assignment Operator – named `operator=`, takes an Element by reference, returns an Element by reference
- Destructor – named ~`Node`, deletes the node and all nodes connected to it
- Next Evaluation – named `hasNext`, takes no parameters, returns a bool

- Next Setter – named `setNext`, takes the value of a Node's address, and performs an assignment to the data member that points to the next Node
- Next Getter – named `getNext`, takes no parameters, returns the address of the next Node (pointed to by this one)
- Element Getter – named `getData`, takes no parameters, returns the value of the Element stored in the Node
- Overloaded Output – named `operator<<`, prints the value of the data member for each Node

Further ensure that the type of data stored by your Node can be changed in one place within the Node class. (Try changing the type to some other type, for example, change form an int to a string and change your Driver code to reflect a string instead of an int. For submission you will need to change your Node class and Driver so that Node stores a float.)

Also ensure that your Node class has the following structure, (in particular, make sure that your data is the correct type – what does the arrow in Figure 1 indicate about the type of the data field?)

Finally, ensure that you provided a labeled picture of what is happening to your Node class for each of `The Big Three`. Make sure that on your drawings it is clear which functions are being called at which times. Take care that your illustrations are adequately labeled so that someone else can follow your work. If things happen over a series of steps, the steps should be ordered.

2.1 Submission

Modify your makefile so that one executable is produced, and that executable is named `node_b`. Make sure that the type of data corresponding to the Element name is `float`. (This may require updating your Driver code to reflect this change.) Additionally, make sure that your Driver is named `NodeDriver.cpp`. To prepare for submission, move all of your code (.cpp files and .h files) and your Makefile into a folder named `username_05_NodeB`, where the username is your Binghamton userid (e.g. jdoe6).

Your submitted code should be free of object files, executables that are produced by your Makefile and files that end in $\sim$. Use `ls -A` to see that there are no hidden files in your folder.

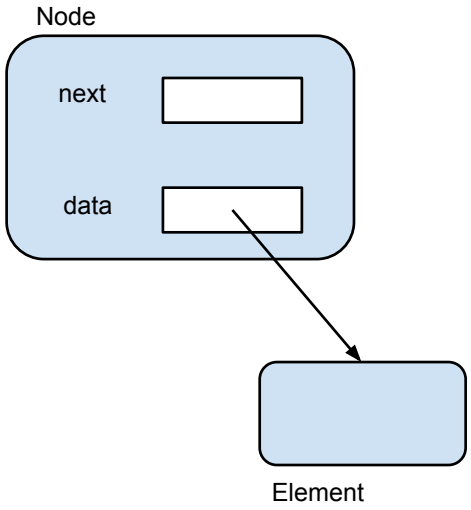

Figure 1: A pictorial representation of the Node class.

Compress the folder and its contents into a tar.gz file that shares the same name as the folder using the `tar` command.

Using our working example, the command to do this will be:

```
tar -czvf jdoe6_05_NodeB.tar.gz jdoe6_05_NodeB/
```

Verify that your submission is correct and upload the appropriate .tar.gz file to blackboard using the submission link in the Lab 05 folder.

The assignment should be uploaded prior to the end of your lab session today. Any file submitted after the end of your lab session will be considered late.

Before you leave, make sure that your name is on your pre-lab drawings and that you hand them in to the CA/TA.