My Version of a Doubly-Linked List

This is my very own rendition of a doubly-linked list data structure which store nodes that are connected to each other via pointers along with a testing suite to showcase its functions. I used C++ to write this program featuring functions that not only build new nodes but adds those nodes to the linked list. Also, the node and linked list classes are created using generic templates which allows them to store various types of data for the convenience of the user.

|  |  |  |  |
| --- | --- | --- | --- |
| Class/Function Name | Parameters | Return Type | Explanation |
| **Node( )** | None | None | This creates a new Node constructor that contains no data |
| **Node(a,b,c)** | Generic\_Type | None | This creates a new Node that contains data specified by what is entered in as an argument |

Classes

**Node**

**DL\_List**

|  |  |  |  |
| --- | --- | --- | --- |
| Class/Function Name | Parameters | Return Type | Explanation |
| **DL\_List( )** | None | None | This creates a new DL\_List constructor that contains no nodes |
| **DL\_List(a,b,c)** | Generic\_Type | None | This creates a new linked list object, DL\_List, that contains nodes starting with a top node with null head and tail pointers. |
| **Insert(a)** | Generic\_Type | Void | Creates a new node with a pointer & adds it to the top of the existing linked list. Uses data passed to it as an argument as the node’s data. |
| **\*Find(a)** | Generic\_Type | Pointer | Iterates through the current linked list nodes for data that matches data passed to it as argument. Returns pointer of matching node |
| **Delete(a)** | Generic\_Type Node pointer | Void | Iterates through current linked list for node that matches node pointer passed to it as argument. Deletes the node that matches its argument |
| **DisplayDLL( )** | None | Void | Iterates though current linked list and prints data from every node in list. Usually done through console. |

**Main/Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Main( )** | None | Void | Main function that begins execution of the entire program. It also starts the menu loop. |
| **Menu** | N/A | N/A | Uses a do While loop to offer the user options for executing the testing suite. When user chooses an option certain functions are automatically ran to test for a full range of conditions including creating empty constructors, creating lists containing the correct nodes and the performance of all functions. |