Abbreviations used :

DS -> Data Structure

Project #2: Generic CPP Data structure library (a mimic of many of the great methods in .NET)

**A:** Types of Data Structures:  
 1) Linked List  
 2) Stack  
 3) Trees  
 4) Graphs

1. Linked List:-

\*constructed using Template feature of C++

Member function : insert(), insertAt(), insertRange(), insertRangeAt(), delete(), deleteAt(), deleteRange(), deleteRangeAt()

The insert() and insertRange() methods will insert the passed data at either beginning or at the end of the target DS depending upon the optional index passed (0 for beginning and 1 for end)  **#DONE IMPLEMENTING**

The insertAt() and the insertRangeAt() methods will insert the passed data element /array at the specified index only. **#DONE IMPLEMENTING**

The remove() method deletes the 1st item if no index parameter passed and the last item if END macro is passed.

\*an overloaded definition should be such that the supplied argument itself gets deleted from the list

The removeRange() method deletes the specified length of items as the argument to it from the beginning if no 2nd argument is passed and deletes the same length from end if the END macro is passed as the 2nd argument

\*there can be another overloaded definition for removeRangeAt()

The removeAt() method simply deletes the element at the index specified as the argument.

The removeRangeAt() method deletes the length of items from the given position as in the argument list. Has two arguments – length of items , start index

*\*implement a special method getridOf(); which will take an argument whose every instance form the referenced list must be removed.* ***DONE IMPLEMENTING***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.no.** | **Method Name** | **Method Role** | **Argument List** | **Return Type** | **STATUS** |
| 1 | insert() | Takes in an item as an argument and adds it to the beginning of the data structures (linked list) if a second argument as END is passed then adds the item to the end of the list. | (item,END)  OR  (item,BEG) | void | DONE IMPLEMENTING |
| 2 | insertAt() | Takes in an item and a mandatory index value to insert that item into the Data structure | (item,index) | void | DONE |
| 3 | insertRange() | Takes in a Link List or another data structure as an argument and adds it at the beginning if no second argument is specified or else adds it to the end of our List if END macro is specified. | (list\_name,END)  OR  (list\_name,BEG) | void | DONE *\*needs some correction* |
| 4 | insertRangeAt() | Takes in a list as an argument to insert into the referenced List at the specified index | (list\_name,index) | void | DONE |
| 5 | remove() | Takes in an item as an argument, finds it in the list and removes it. | (item) | void | DONE |
| 6 | removeAt() | Takes in an index as argument and removes the item there | (index) | void | DONE |
| 7 | removeRange() | -- | -- | -- | -- |
| 8 | removeRangeAt() | -- | -- | -- | -- |
| 9 | getridOf() | This is a special method which removes all instances of the passed item form the reference list. | (item) | void | DONE |
| 10 | ToUpperCase() | Will convert the string into upper case of language | - | void |  |