|  |  |  |
| --- | --- | --- |
| **American University of Sharjah**  **School of Engineering**  Computer Engineering Department  P. O. Box 26666 Sharjah, UAE |  | **Instructor:** Dr. Ghassan Z. Qadah  **Lab Instructor:** Ms.Praveena Kolli  **Office**: EB2-126  **Phone**: 971-6-515-2352  **e-mail**: pkolli@aus.edu  **Semester**: Spring 2019 |

**CMP305L Data Structures and Algorithms**

**Lab #5 –Unsorted List**

**Objectives:**

* To understand and implement unsorted linear and linked list
* To understand processing of unsorted linear and linked list

**Exercise 1:**

Implement the given below UnsortedType class.

|  |
| --- |
| class UnsortedType  {  public :  // LINKED LIST IMPLEMENTATION  UnsortedType ( ) ;  void MakeEmpty ( ) ;  bool IsFull ( ) const ;  int Length ( ) const ;  void RetrieveItem ( ItemType& item, bool& found ) ;  void InsertItem ( ItemType item ) ;  void DeleteItem ( ItemType item ) ;  void ResetList ( );  void GetNextItem ( ItemType& item ) ;  ~UnsortedType ( ) ; //one of the three safety functions  private :  NodeType\* listData;  int length;  NodeType\* currentPos;  } ; |

**Exercise 2:**

Overload the following operators and test them.

* & for returning intersection of items UnsortedType *operator &(const UnsortedType &list1, const UnsortedType &list2)* as *friend* function.
* | for returning union of items *UnsortedType operator|(const UnsortedType &list1, const UnsortedType &list2)* as *friend* function*.*

**Note:** Must ensure that no duplicate items exist in the lists that are passed as arguments to the functions implementing intersection (&) and union (|).

**Note**: Write an appropriate main for each exercise to test the functionality.