// Dalton Wright
// CSCI 301 01
// Project #7
// UML Charts

Class FrontList1
-listPtr: LinkedList<ItemType>*
+FrontList1(); +FrontList1(const FrontList1<ItemType>& aList) +virtual ~FrontList1(); +insert(const ItemType& newEntry): bool +remove(): bool +isEmpty(): bool +const getLength (): int +clear(): void +peek(): ItemType +getEntry(const int& position) throw(PrecondViolatedExcep): ItemType

Class FrontList2
+FrontList2(); +FrontList2(const FrontList2<ItemType>& aList) +virtual ~FrontList2(); +insert(const ItemType& newEntry): bool +insert(int newPostion, const ItemType& newEntry) throw(PrecondViolatedExcep): bool +remove(): bool +remove(int position): bool +peek(): ItemType +setEntry(int& position, const ItemType& newEntry) throw(PrecondViolatedExcep): void

Class FrontList3
+FrontList3(); +FrontList3(const FrontList3<ItemType>& aList) +virtual ~FrontList3(); +insert(const ItemType& newEntry): bool +remove(): bool +isEmpty(): bool +const getLength (): int +clear(): void +peek(): ItemType +getEntry(const int& position) throw(PrecondViolatedExcep): ItemType

Class LinkedSortedList
-headPtr: Node<ItemType>*
-itemCount: int
-getNodeBefore(const ItemType& anEntry) const: Node<ItemType>*
-getNodeAt(int position) const: Node<ItemType>*
-copyChain(const Node<ItemType>* origChainPtr): Node<ItemType>*
+LinkedSortedList();
+LinkedSortedList(const LinkedSortedList<ItemType>& aList);
+virtual ~LinkedSortedList();
+insertSorted(const ItemType& newEntry): void
+removeSorted(const ItemType& anEntry): bool
+getPosition(const ItemType& newEntry) const: int
+isEmpty() const: bool
+getLength() const: int
+remove(int position): bool
+clear(): void
+getEntry(innt position): ItemType

Class Node
-nodeItem: ItemType
-previous: Node<ItemType>*
-next: Node<ItemType>*
+Node();
+Node(const ItemType& anItem);
+Node(const ItemType& anItem, Node<ItemType>* nextNodePtr);
+Node(const ItemType& anItem, Node<ItemType>* nextNodePtr, Node<ItemType>* prevNodePtr);
+setItem(const ItemType& anItem): void
+setNext(Node<ItemType>* nextNodePtr): void
+setPrev(Node<ItemType>* prevNodePtr): void
+getItem() const: ItemType
+getNext() const: Node<ItemType>*
+getPrev() const: Node<ItemType>*

Class PrecondViolatedExcep : public logic_error
+PrecondViolatedExcep(const string& message = “ “);