Simple_stat

-data_obj: LList*
-data_mean: double = 0.0
-data_mean: double = 0.0
-data_sqsum: double = 0.0
-data_var: double = 0.0
-data_sd: double = 0.0
-data_min: N = 0
-data_max: N = 0
-data_sum: double = 0.0

-calc_stats(new_element:N, addval:int): void
+Simple_stat()
+Simple_stat(data_feed:L&)
+~Simple_stat()
+operator[](indexnum:int): auto
+append(num:N): void
+removem(m:N, reps:int): void
+empty(): void
+search(num:N): std::pair<int,int>
+length_unique(): int
+length_total(): int
+unique_set(): std::set<N>
+feed(data_feed: const L&): void
+get_mean(): double
+get_SD(): double
+get_min(): auto
+get_max(): auto

LList

-head: E*
-tail: E*
-curr: E*
-cnt: int

-init(): void
-removeall(): void
+LList(size: int) = 0
+~LList()
+print(): void
+clear(): void
+insert(it: const E&): void
+append(it: const E&): void
+remove(): E
+moveToStart(): void
+moveToEnd(): void
+prev(): void
+next(): void
+length(): int
+currPos(): int
+moveToPos(pos: int): void
+getValue(): const E&
+reverse(): void
+isAtEnd(): bool

List

-operator=(const List&): void
-List(const List&)
+List()
+~List()
+clear(): void
+insert(const E&): void
+append(const E&): void
+remove(): E
+moveToStart(): void
+moveToEnd(): void
+prev(): void
+next(): void
+length(): int
+currPos(): int
+moveToPos(pos: int): void
+getValue(): const E&
+reverse(): void
+isAtEnd(): bool

Link

-freelistSL: Link<E>*
+element: E
+next: Link*

+Link(elemval: const E&, nextval: Link* = NULL)
+Link(nextval: Link* = NULL)
+operator new(size_t): void*
+operator delete(ptr: void*): void