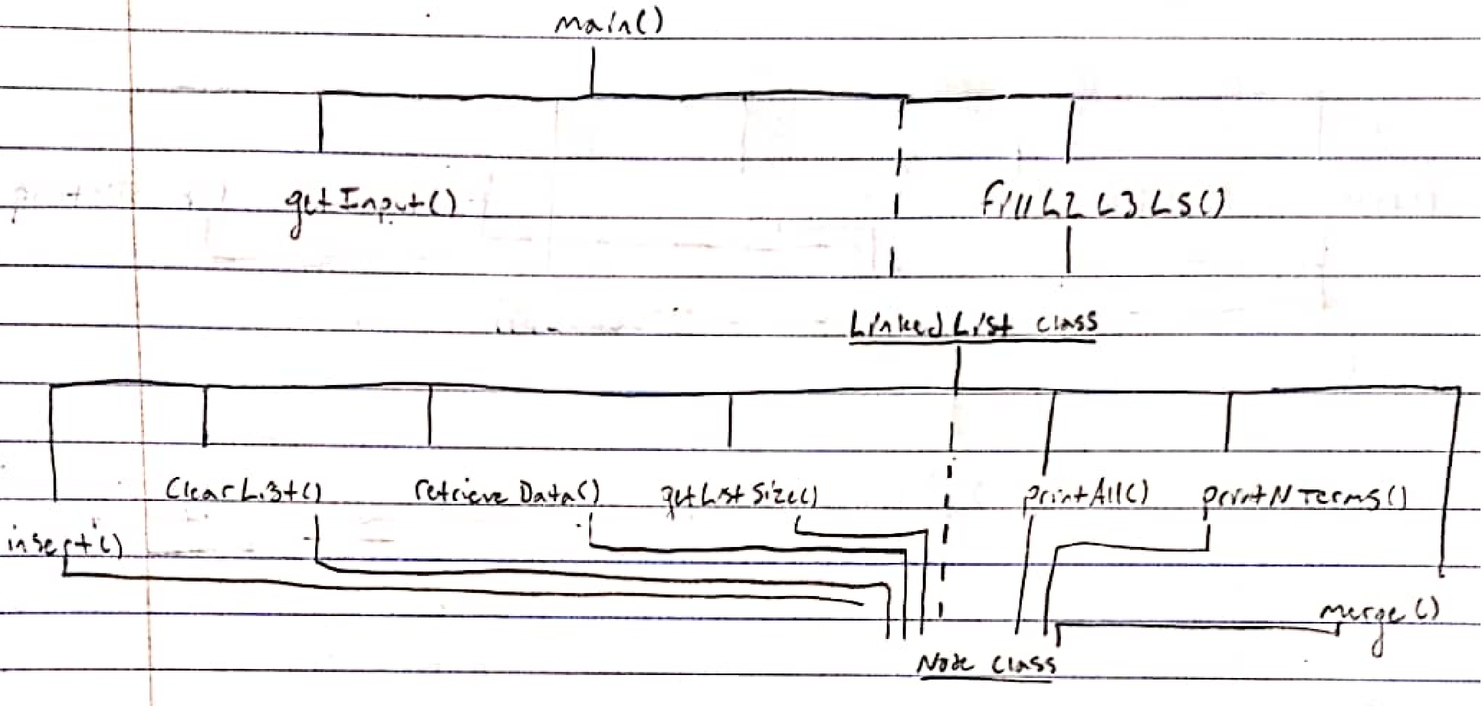
Dependencies



Testing begins by instantiating an object of type Linked List with an object of type Node:

- All methods in Linked List depend on insert(), so it should be tested first.
- retrieveData(), getListSize(), printAll() and PrintNTerms can be tested in any order after insert()
- merge() can be tested after printAll() is confirmed accurate
- Fill L2 L3 L5() requires insert() and merge() to function, requiring Linked List and Node to function.
- getInput() is standalone and may be tested at any time

Testing procedures

Linked List class:

Node class:

- Instantiate a node object with data in data
- Verify data in node is correct and nextPtr is null.

ClearList method:

- Verify the head Node object is set to null after execution.
- Attempt to reference nextPtr of the head object; Should result in error if ClearList() is successful.

retrieveData method:

- Call insert() method with 3 different points of data.
- Call retrieveData 3 times, with 0, 1, 2 for listIndex args each call. If successful, method should return values in the same order they were input.

getListSize method:

- Call insert n times; getListSize() should return n, where n is the number of elements in the Linked List.

PrintAll method:

- Not used in final program! For debugging!
- Call insert() with any number of data points; printAll() should output each data point in the order they were input, thus it is printing from beginning → end of the linked list.

printNTerms method:

- Call insert() with n > 1 data points.
- Call printNTerms() with n-1 as an arg.
- printNTerms() should print the first n-1 terms of the linked list in the same order they were input.
- Call printNTerms() with n as an arg.
- PrintNTerms() should print all terms in the list in the same order they were input.

merge method:

- Create two linked lists with n ≥ 2 distinct elements each, sorted in ascending order.
- Call merge() with the list containing the larger first integer in the list for arg1, other list for arg2.
- The method should return the head of the smaller linked list, with each successive node containing data greater than the preceding element. printAll() can be used to verify accuracy.
- Repeat the above with linked list containing smaller head as arg1 and verify the same result is achieved.

* Create two linked lists of length n both containing the same value integers.

- On calling merge(), the method should return list1's head and no elements in list1 should be changed

insert method:

- Create a linked list and call insert with n number of distinct integers.
- Call printAll() with the linked list as an arg; it should print each element in the order inserted.

Sequence Generation Class

fill L2 L3 L5 method:

- Create a linked list with -3, -2 ... 2, 3 as elements.
- Create three additional linked lists to store output
- Call fill method with the populated list as arg1, other lists arg2, arg3, arg4
- Call printAll() for each list after fill method
- Output should read: -3, -2 ... 2, 3 for arg1 list
  -6, -4, ... 4, 6 for arg2 list
  -9, -6 ... 9, 6 for arg3 list
  -15, -10, .. 10, 15 for arg4 list

getInput method:

- Call method and test non-integer inputs; should output that the input is invalid
- Call method and input an integer; print the value returned by the method, should be the same integer as input

main():

- Input an integer n when prompted; output should contain n integers ascending with no prime factors other than 2, 3, or 5.