CS1020E: DATA STRUCTURES AND ALGORITHMS I

Tutorial 5 – Linked List

(Week 7, starting 26 September 2016)

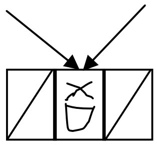
1. Linked List Operations

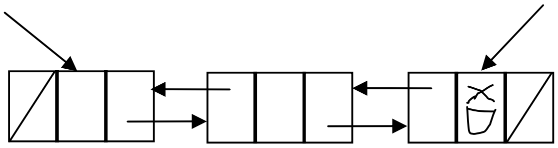
Examine the file `T5_mystery.cpp`, which contains a simplified implementation of a doubly linked list with a sentinel (dummy) node at the back. We are trying to simulate the STL `list<T>` data type.

- What is the **purpose of the class** `GuessWhatIsThis`?
- What does **each operation** in `mysteryA..L()` do?
- What is each method's **STL equivalent**, and how is that used?

This question is designed for you to **trace through** each linked list operation, based on the **implementation of each operation**. Only look at the behavior of each operation in main() to check your answers!

Remember, since tutorial 1, we have been asking you to draw diagrams of what happens in memory to trace through what each operation does. For reference-based data structures, it is often important to be able to visualize what each statement does to your data in memory. Use the diagrams below to help you.





2. More STL List<T> Operations - Online Discussion

How is each of the following list method used, and what does each do to the nodes in a linked list?

```
void merge(list<T>& other);
void resize(int newSize); // C++ 11 overloaded version
void reverse();
void splice(...); // The overload with the shortest signature
```

You can try writing code to implement those functions if you like, and explain them on the Facebook discussion group **after** the tutorials on Friday...!

- Now halfway through! ☺ -

Think of various cases
Design full algo before coding