# CMPT 280

## Topic 1: Linear Data Structures

Mark G. Eramian

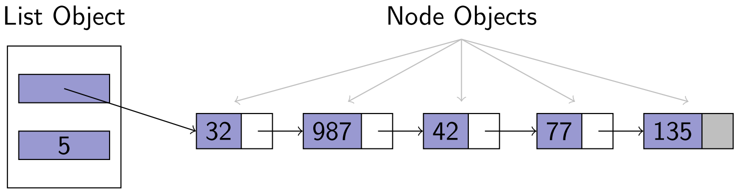
University of Saskatchewan

## References

- Textbook, Chapter 1

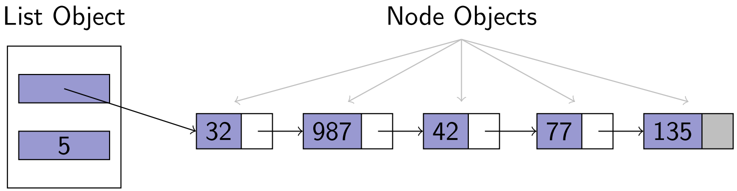
## Linked Lists

Recall the structure of a Java implementation singly-linked list:



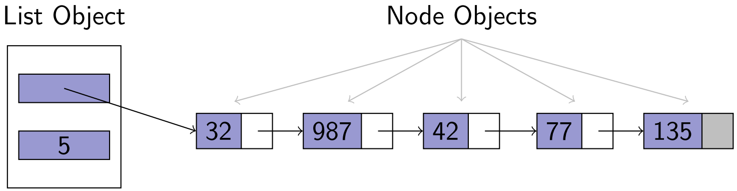
## Exercise 1

- Write the class definitions (instance variables and constructors only) for the Node and List objects. We would like to be able to store elements of any type.



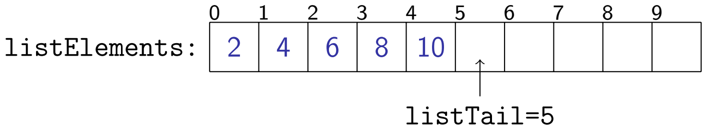
## Exercise 2

- Write the following methods for the list:
  - `isEmpty`
  - `isFull`
  - `insertFirst`
  - `deleteFirst`
  - `firstItem`



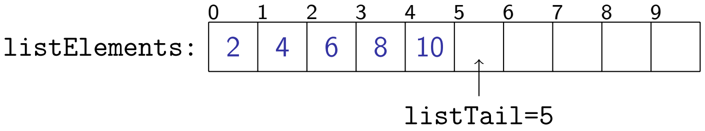
## Exercise 3

- Write a java class definition for array-based list described in the readings. Define the instance variables and a constructor, but for now, don't worry about defining the methods. Can you write it so we can store any type of element we want in the list?



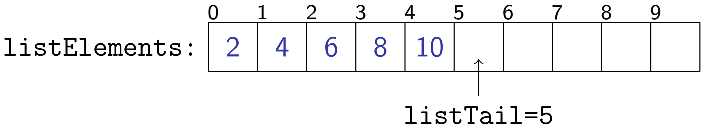
## Exercise 4

- Write two methods that test whether the list is full and empty, respectively.



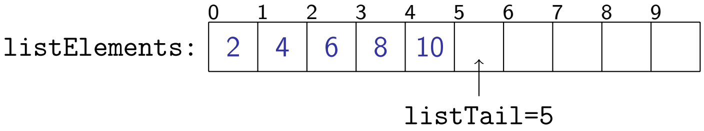
## Exercise 5

- Write the `insertFirst` method for our array-based list class which inserts a new element at the beginning of the list.



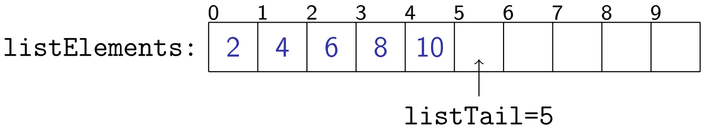
## Exercise 6

- Write the `deleteFirst` method for our array-based list class that removes the element at the beginning of the list.



## Exercise 7

- Write the `firstItem` method for our array-based list class that returns the data element at the beginning of the list, but does not modify the list.



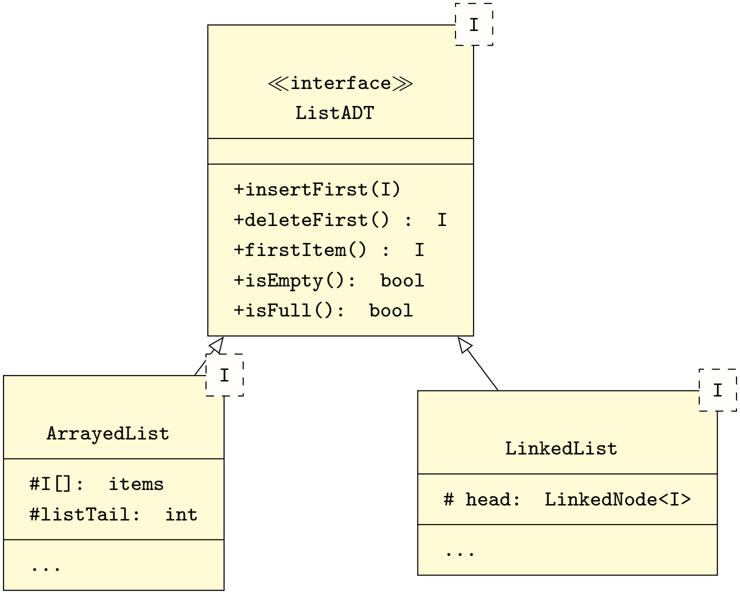
## List Observations

- Both versions of our list (linked and arrayed) look the same to the user – same interface, different internals!

## Exercise 8

- Write a java interface for the `listADT` interface.
- Update the class definitions of `LinkedList` and `ArrayedList` to declare that they implement the `ListADT` interface.

## Common List Interface



## What Next?

- How do we know whether what we just wrote works?
- Next class reading: Chapter 2 – Regression Testing.