Object-oriented programming

Lecture #12: Container Classes

Lists

- A list consists of a set of sequentially organized elements
- A specific type of graph, where each node except the first has a single preceding node, and each node except the last has a single following node
- Contains 0-n nodes
- Implementation: array and linked lists

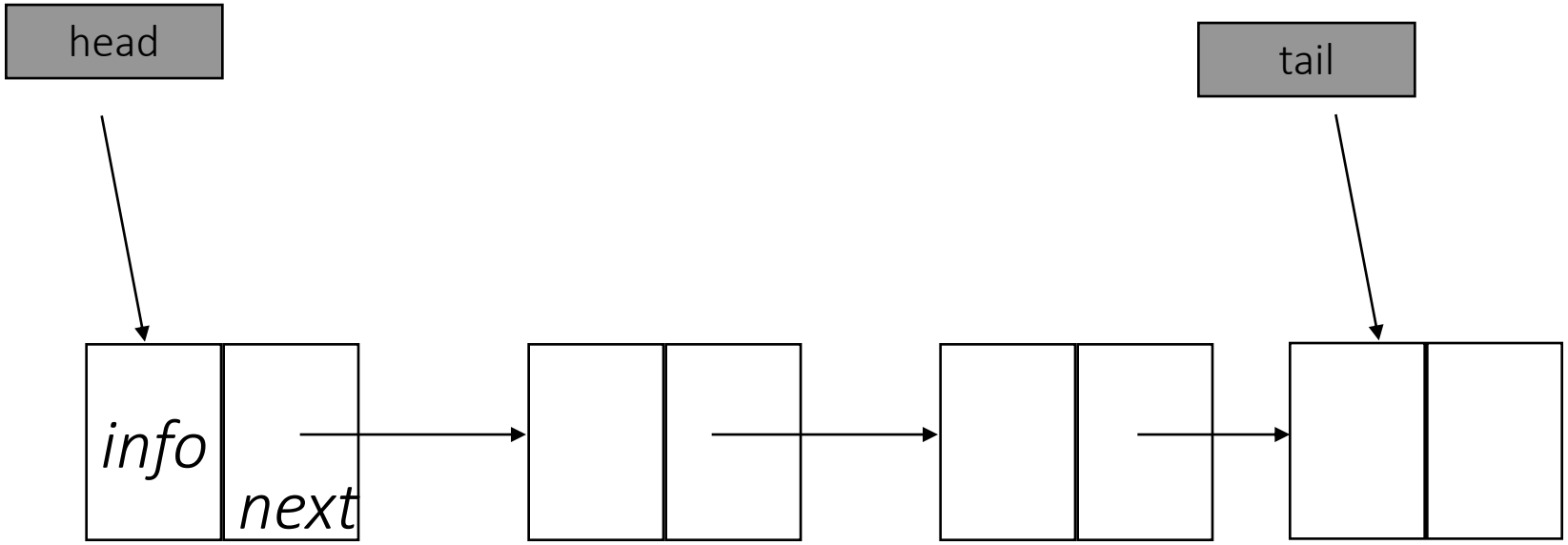
Linked Lists

- Each element in a list is called a **node**, and a connection between any two nodes is called a **link**
- Implementation: dynamic memory allocation (saves memory)
- Limitation: nodes are accessed sequentially

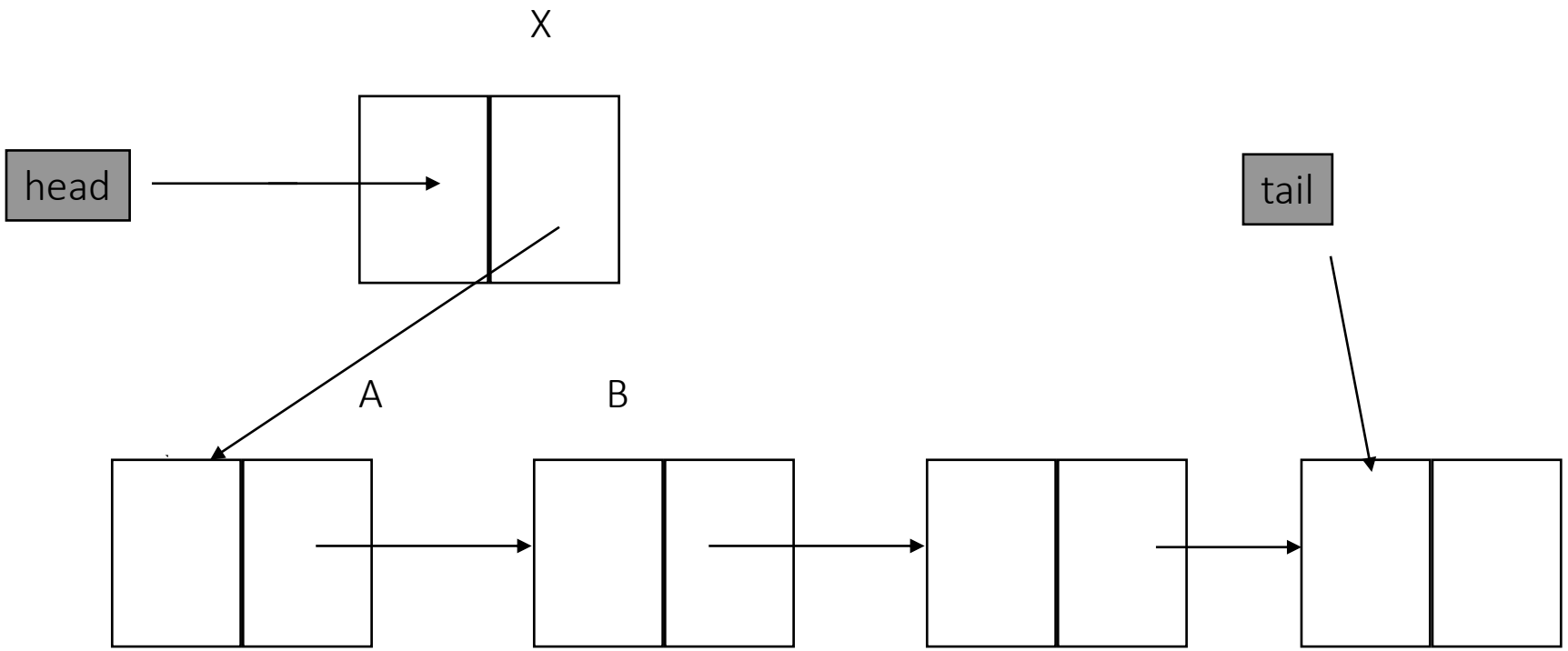
Operations on a List

- append: add a node at the end
- prepend: add a node at the beginning
- insert: insert a node in place
- find: find a specific node
- get: get a node at the current position
- replace: replace the content of a node
- isEmpty: find out if the list is empty
- remove: remove a node
- clear: remove all the nodes

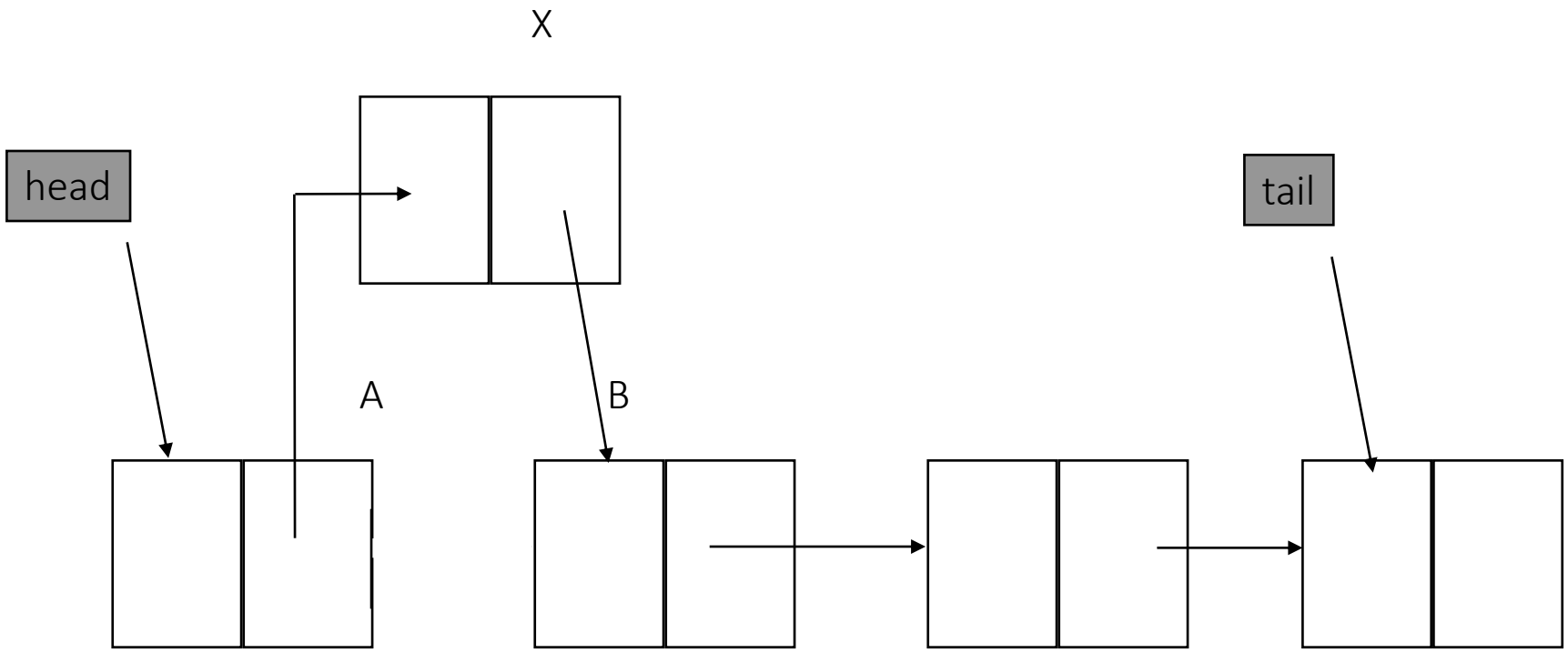
Generic Linked List



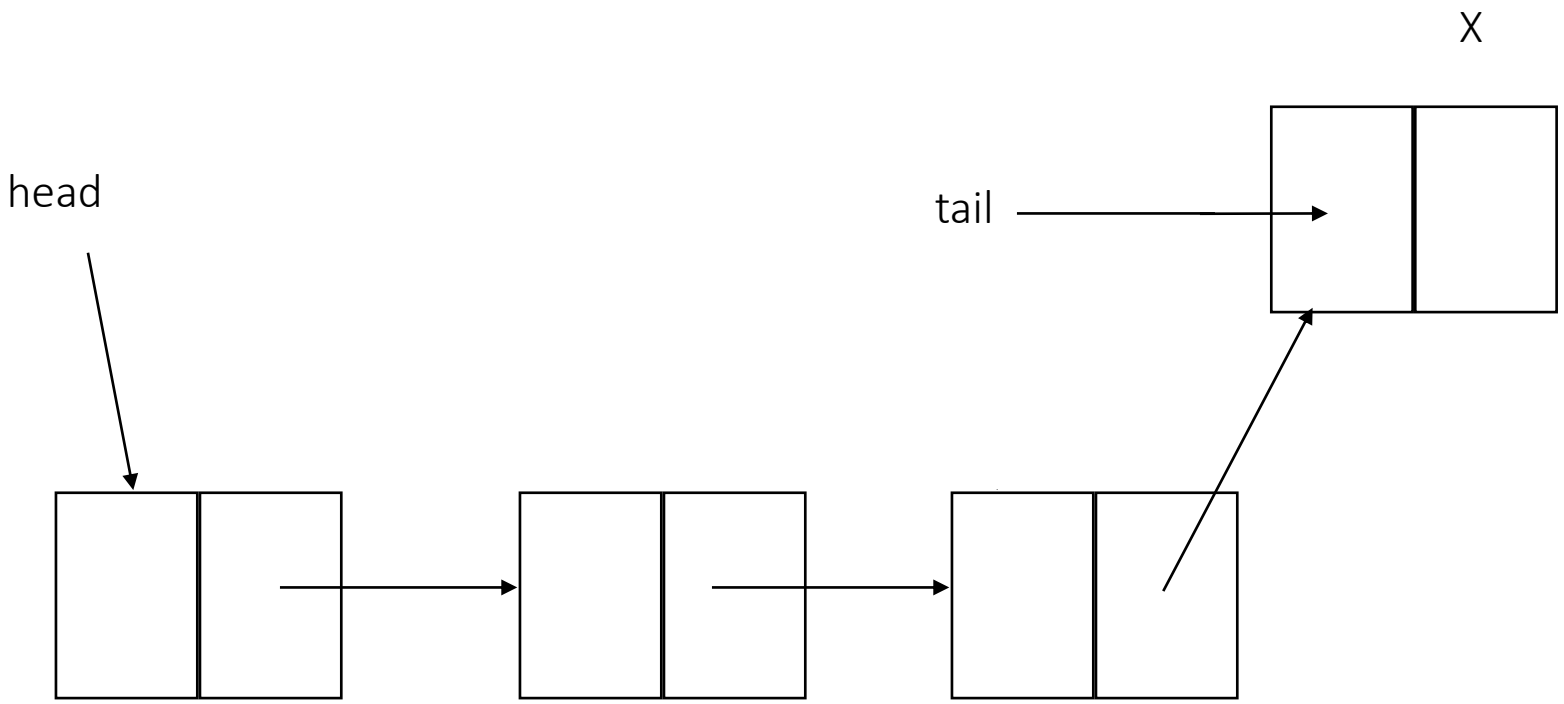
Prepend Operation



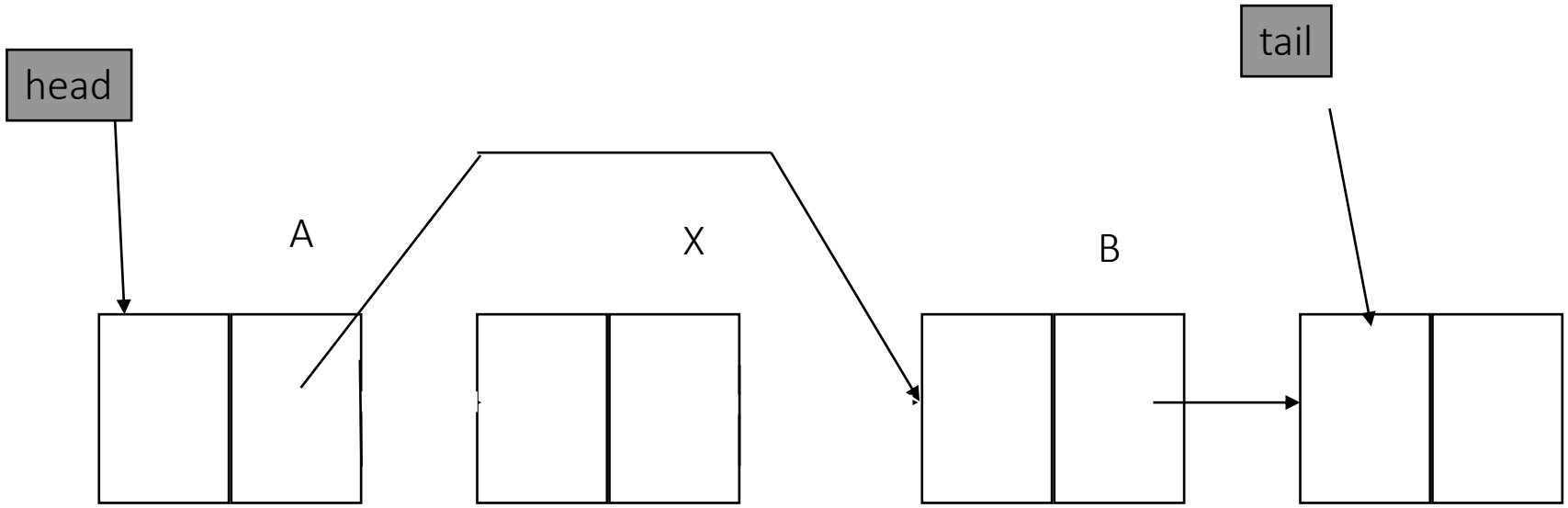
Insert Operation



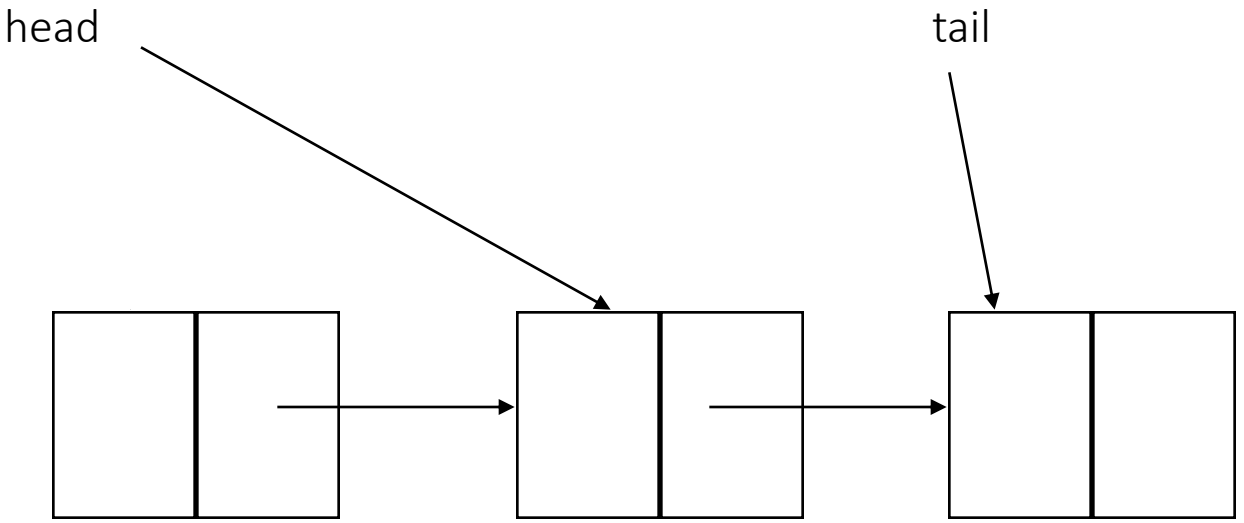
Append Operation



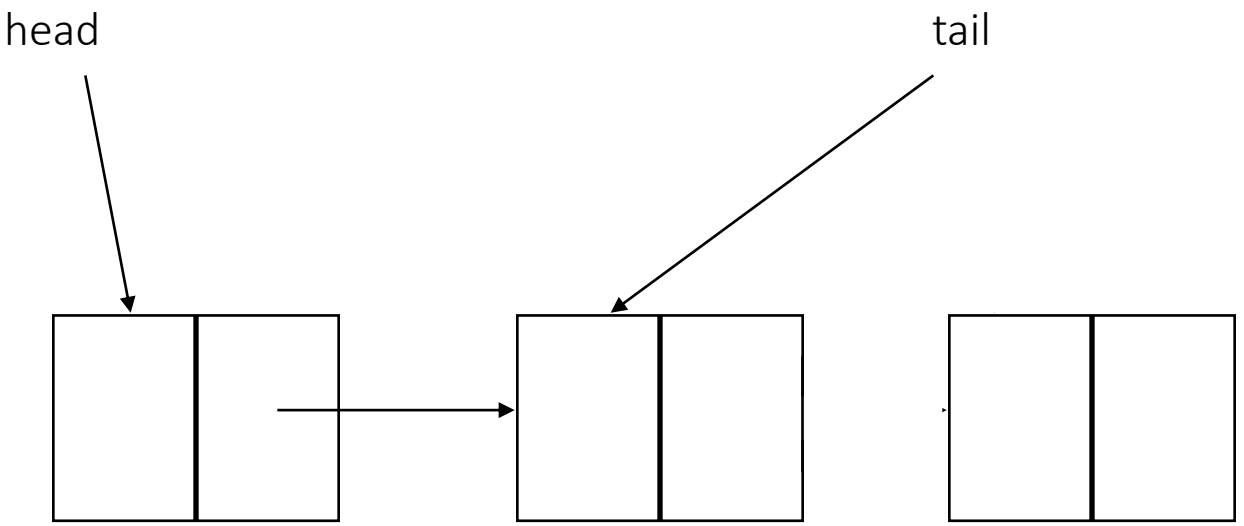
Remove a Node



Remove Head



Remove Tail



List with Different Types

- Using templates
- With the same structure, we can have integer, float, or string as a value of a node
- We can have a list with values in different types with the same base class

Exercises