

# CSE143 Cheat Sheet

## Linked Lists (16.2)

Below is a partial implementation of ListNode

```
public class ListNode {
    public int data;
    public ListNode next;

    public ListNode() {
    }

    public ListNode(int data) {
        this.data = data;
    }
}
```

## Math Methods (3.2)

*mathematical operations*

Math.abs( <i>value</i> )	absolute value
Math.min( <i>v1</i> , <i>v2</i> )	smaller of two values
Math.max( <i>v1</i> , <i>v2</i> )	larger of two values
Math.round( <i>value</i> )	nearest whole number
Math.pow( <i>b</i> , <i>e</i> )	b to the e power

## Iterator<E> Methods (11.1)

*(An object that lets you examine the contents of any collection)*

hasNext()	returns true if there are more elements to be read from collection
next()	reads and returns the next element from the collection
remove()	removes the last element returned by next from the collection

## List<E> Methods (10.1)

*(An ordered sequence of values)*

add( <b>value</b> )	appends value at end of list
add( <b>index</b> , <b>value</b> )	inserts given value at given index, shifting subsequent values right
clear()	removes all elements of the list
indexOf( <b>value</b> )	returns first index where given value is found in list (-1 if not found)
get( <b>index</b> )	returns the value at given index
remove( <b>index</b> )	removes/returns value at given index, shifting subsequent values left
set( <b>index</b> , <b>value</b> )	replaces value at given index with given value
size()	returns the number of elements in list
isEmpty()	returns true if the list's size is 0
addAll( <b>collection</b> )	adds all elements from the given collection to the end of the list
contains( <b>value</b> )	returns true if the given value is found somewhere in this list
remove( <b>value</b> )	finds and removes the given value from this list
removeAll( <b>list</b> )	removes any elements found in the given collection from this list
iterator()	returns an object used to examine the contents of the list

## Set<E> Methods (11.2)

*(A fast-searchable set of unique values)*

add( <b>value</b> )	adds the given value to the set
contains( <b>value</b> )	returns true if the given value is found in the set
remove( <b>value</b> )	removes the given value from the set
clear()	removes all elements of the set
size()	returns the number of elements in the set
isEmpty()	returns true if the set's size is 0
addAll( <b>collection</b> )	adds all elements from the given collection to the set
containsAll( <b>collection</b> )	returns true if set contains every element from given collection
removeAll( <b>collection</b> )	removes any elements found in the given collection from this set
retainAll( <b>collection</b> )	removes any elements <i>not</i> found in the given collection from this set
iterator()	returns an object used to examine the contents of the set

**Map<K, V> Methods (11.3)***(A fast mapping between a set of keys and a set of values)*

put ( <b>key</b> , <b>value</b> )	adds a mapping from the given key to the given value
get ( <b>key</b> )	returns the value mapped to the given key (null if none)
containsKey ( <b>key</b> )	returns true if the map contains a mapping for the given key
remove ( <b>key</b> )	removes any existing mapping for the given key
clear ()	removes all key/value pairs from the map
size ()	returns the number of key/value pairs in the map
isEmpty ()	returns true if the map's size is 0
keySet ()	returns a Set of all keys in the map
values ()	returns a Collection of all values in the map
putAll ( <b>map</b> )	adds all key/value pairs from the given map to this map

**Stack<E> Methods (14.1)***(A first in, last out structure)*

push ( <b>value</b> )	pushes a given value to the top of the stack
pop ()	returns the value at the top of the stack, removing it
size ()	returns the number values in the stack
isEmpty ()	returns true if the stack's size is 0

**Queue<E> Methods (14.1)***(A first in, first out structure)*

add ( <b>value</b> )	adds a given value to the back of the queue
remove ()	returns the value at the front of the queue, removing it
size ()	returns the number of values in the queue
isEmpty ()	returns true if the queue's size is 0

**NOTE** that you **MAY NOT** use a **foreach** loop or **peek()** with **Stacks** and **Queues**.**String Methods (3.3)***(An object for storing a sequence of characters)*

length ()	returns the number of characters in the string
charAt ( <b>index</b> )	returns the character at a specific index
compareTo ( <b>other</b> )	returns how this string compares to the other
equals ( <b>other</b> )	returns true if this string equals the other
toUpperCase ()	returns a new string with all uppercase letters
toLowerCase ()	returns a new string with all lowercase letters
startsWith ( <b>other</b> )	returns true if this string starts with the given text
substring ( <b>start</b> , <b>stop</b> )	returns a new string composed of character from start index (inclusive) to stop index (exclusive)

**Collections Implementations**

List<E>	ArrayList<E> and LinkedList<E>
Set<E>	HashSet<E> and TreeSet<E> (values ordered)
Map<K, V>	HashMap<K, V> and TreeMap<K, V> (keys ordered)
Stack<E>	Stack<E>
Queue<E>	LinkedList<E>