

# ^\_^ CSE 143 MIDTERM EXAM CHEAT SHEET ^\_^

## Constructing Various Collections

```

List<Integer> list = new ArrayList<Integer>();
Queue<Double> queue = new LinkedList<Double>();
Stack<String> stack = new Stack<String>();
Set<String> words = new HashSet<String>();
Map<String, Integer> counts = new TreeMap<String, Integer>();
    
```

### Methods Found in ALL collections (Lists, Stacks, Queues, Sets, Maps)

|                             |   |
|-----------------------------|---|
| clear()                     | removes all elements of the collection                                |
| equals( <b>collection</b> ) | returns true if the given other collection contains the same elements |
| isEmpty()                   | returns true if the collection has no elements                        |
| size()                      | returns the number of elements in the collection                      |
| toArray()                   | returns an array of the elements in this collection                   |
| toString()                  | returns a string representation such as "[10, -2, 43]"                |

### Methods Found in both Lists and Sets (ArrayList, LinkedList, HashSet, TreeSet)

|                                |   |
|--------------------------------|---|
| add( <b>value</b> )            | adds value to collection (appends at end of list)                           |
| contains( <b>value</b> )       | returns true if the given value is found somewhere in this collection       |
| iterator()                     | returns an Iterator object to traverse the collection's elements            |
| remove( <b>value</b> )         | finds and removes the given value from this collection                      |
| removeAll( <b>collection</b> ) | removes any elements found in the given collection from this one            |
| retainAll( <b>collection</b> ) | removes any elements <i>not</i> found in the given collection from this one |

### List<E> Methods (10.1)

|                             |  |
|-----------------------------|--|
| add( <b>index, value</b> )  | inserts given value at given index, shifting subsequent values right             |
| 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   |
| lastIndexOf( <b>value</b> ) | returns last index where given value is found in list (-1 if not found)          |
| remove( <b>index</b> )      | removes/returns value at given index, shifting subsequent values left            |
| set( <b>index, value</b> )  | replaces value at given index with given value                                   |
| subList( <b>from, to</b> )  | returns sub-portion at indexes <b>from</b> (inclusive) and <b>to</b> (exclusive) |

### Stack<E> Methods

|                      |  |
|----------------------|--|
| peek()               | returns the top value from the stack without removing it   |
| pop()                | removes the top value from the stack and returns it;<br>peek/pop throw an <code>EmptyStackException</code> if the stack is empty |
| push( <b>value</b> ) | places the given value on top of the stack   |

### Queue<E> Methods

|                     |   |
|---------------------|---|
| add( <b>value</b> ) | places the given value at the back of the queue   |
| peek()              | returns the front value from the queue without removing it;<br>returns null if the queue is empty                                   |
| remove()            | removes the value from the front of the queue and returns it;<br>throws a <code>NoSuchElementException</code> if the queue is empty |

# ^\_^ CSE 143 MIDTERM EXAM CHEAT SHEET ^\_^

## Map<K, V> Methods (11.3)

|                           |  |
|---------------------------|--|
| containsKey( <b>key</b> ) | true if the map contains a mapping for the given key |
| get( <b>key</b> )         | the value mapped to the given key (null if none)     |
| keySet()                  | returns a Set of all keys in the map                 |
| put( <b>key, value</b> )  | adds a mapping from the given key to the given value |
| remove( <b>key</b> )      | removes any existing mapping for the given key       |
| toString()                | returns a string such as "{a=90, d=60, c=70}"        |
| values()                  | returns a Collection of all values in the map        |

## String Methods (3.3, 4.4)

|                                |   |
|--------------------------------|---|
| charAt( <b>i</b> )             | the character in this String at a given index                                     |
| contains( <b>str</b> )         | true if this String contains the other's characters inside it                     |
| endsWith( <b>str</b> )         | true if this String ends with the other's characters                              |
| equals( <b>str</b> )           | true if this String is the same as <i>str</i>                                     |
| equalsIgnoreCase( <b>str</b> ) | true if this String is the same as <i>str</i> , ignoring capitalization           |
| indexOf( <b>str</b> )          | first index in this String where given String begins (-1 if not found)            |
| lastIndexOf( <b>str</b> )      | last index in this String where given String begins (-1 if not found)             |
| length()                       | number of characters in this String   |
| startsWith( <b>str</b> )       | true if this String begins with the other's characters                            |
| substring( <b>i, j</b> )       | characters in this String from index <i>i</i> (inclusive) to <i>j</i> (exclusive) |
| toLowerCase(), toUpperCase()   | a new String with all lowercase or uppercase letters                              |

## Random Methods (5.1)

|                       |   |
|-----------------------|---|
| nextBoolean()         | random true/false result                |
| nextDouble()          | random real number between 0.0 and 1.0  |
| nextInt()             | random integer                          |
| nextInt( <b>max</b> ) | random integer between 0 and <i>max</i> |