CSE143 Cheat Sheet

`Math` Methods (3.2) *mathematical operations*

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

Stacks and Queues (14.2) *(LIFO and FIFO structures)*

Queues should be constructed using the Queue<E> interface and the LinkedList<E> implementation (you may not pass any arguments to the constructor). For example, to construct a queue of String values, you would say:

```
Queue<String> q = new LinkedList<>();
```

Stacks should be constructed using the Stack<E> class (there is no interface):

```
Stack<Integer> s = new Stack<>();
```

For Stack<E>, you are limited to the following operations (no iterator or foreach loop):

`push(`**value**`)`	pushes the given value onto the top of the stack
`pop()`	removes and returns the top of the stack
`isEmpty()`	returns `true` if this stack is empty
`size()`	returns the number of elements in the stack

For Queue<E>, you are limite d to the following operations (no iterator or foreach loop):

`add(`**value**`)`	adds the given value at the end of the queue
`remove()`	removes and returns the front of the queue
`isEmpty()`	returns `true` if this queue is empty
`size()`	returns the number of elements in the queue

`String` Methods (3.3) *(An object for storing a sequence of characters)*

`length()`	returns the number of characters in the string
`charAt(`**index**`)`	returns the character at a specific index
`equals(`**other**`)`	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(`**other**`)`	returns true if this string starts with the given text
`substring(`**start, stop**`)`	returns a new string composed of characters from start index (inclusive) to stop index (exclusive)
`substring(`**start**`)`	returns a new string composed of characters from start index (inclusive) to the end of the string