Code Handout to accompany MT 2 exam CSCI 455 [S21 Bono]

`Set<ElmtType>` Interface (selected methods)

The classes that implement this interface are: **`TreeSet`** and **`HashSet`.**

`boolean contains(ElmtType elmt)`	Returns true iff `elmt` is in the set
`int size()`	Returns number of elements in the set
`boolean add(ElmtType elmt)`	Ensures that `elmt` is in the set. Returns true iff the set changed as a result of this call
`boolean remove(ElmtType elmt)`	Removes `elmt` from the set. Returns true iff the set changed as a result of this call
`boolean isEmpty()`	Returns true iff the set contains no elements.
`Iterator<ElmtType> iterator()`	Returns an iterator over the elements in the set.

`Map<KeyType, ValueType>` Interface (selected methods)

The classes that implement this interface are: **`TreeMap`** and **`HashMap`.**

`ValueType put(key, value)`
Associates the specified value with the specified key in this map. If the map previously contained a mapping for this key, the old value is replaced by the specified value. Returns the previous value associated with specified key, or `null` if there was no mapping for key.

`ValueType get(key)`
Returns the value to which this map maps the specified key or `null` if the map contains no mapping for this key.

`boolean containsKey(key)`
Returns `true` iff the map contains a mapping for the specified key.

`ValueType remove(key)`
Removes the mapping for this key from this map if it is present, otherwise returns `null`.

`int size()` Number of key-value mappings in this map.

`boolean isEmpty()` Returns true if this map contains no key-value mappings.

`Set<Map.Entry<KeyType,ValueType>> entrySet()`
Returns a set view of the entries contained in this map.

`Set<Map.Entry<KeyType,ValueType>> keySet()`
Returns a set view of the keys contained in this map.

`Map.Entry<KeyType, ValueType>` Interface

`KeyType getKey()`	Return the key of the entry
`ValueType getValue()`	Return the value of the entry
`void setValue(newVal)`	Replace the current value with `newVal`

`Iterator<ElmtType>` Interface (selected methods)

Some classes that implement this interface are: **`Scanner`, `ListIterator`**:

`boolean hasNext()`
Returns `true` iff the iteration has more elements.

`ElmtType next()`
Returns the next element in the iteration. Each successive call returns a different element in the underlying collection. For `Scanner` the `ElmtType` is always `String`.

`Collections` class (selected methods)

The `Collections` class contains static methods that operate on collections. Note: `ArrayList` and `LinkedList` both implement the `List` interface used below.

`static void sort(List<ElmtType> list)`

Sorts the list into ascending order according to the natural ordering of its elements (i.e., using `compareTo`).

`static void sort(List<ElmtType> list, Comparator<ElementType> c)`

Sorts the list according to the order specified by the comparator.

`Comparator<Type>` interface

An object that can compare two objects of type `Type`. Has one method defined by the interface:

`int compare(Type object1, Type object2)`

Must return a negative number if `object1` should come before `object2`, `0` if `object1` and `object2` are equal, or a positive number if `object1` should come after `object2`.

`Comparable<Type>` interface

An object that can be compared to another object of the same type. Has one method defined by the interface:

`int compareTo(Type other)`

`a.compareTo(b)` must return a negative number if `a` should come before `b`,
`0` if `a` and `b` are equal, and a positive number otherwise

Selected methods of Java `Rectangle` class:

`new Rectangle(x, y, width, height)`

Constructs rectangle object whose top-left corner is at (`x`, `y`) and with given `width` and `height`.

`r.translate(deltaX, deltaY)`

Translates `Rectangle r` the indicated distance, to the right along the x coordinate axis by `deltaX`, and downward along the y coordinate axis by `deltaY`. This is a mutator.

`r.getX()`   `r.getY()`

Gets the x or y coordinate, respectively of `Rectangle r`

`r.getWidth()`   `r.getHeight()`

Gets the with or height, respectively of `Rectangle r`