Specification

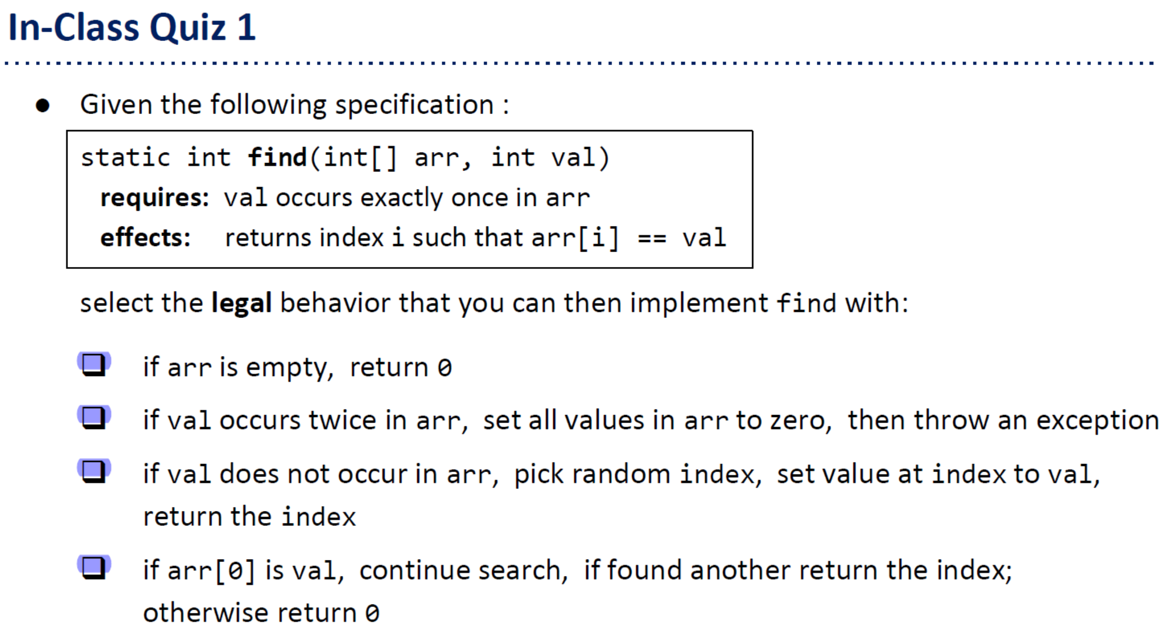
A ***specification*** is like a contract for part of your program
○ saying what it can count on from the rest of the program,
○ and what it's expected to do in return

A specification of a method consists of two clauses:
○ a ***precondition***, indicated by the keyword requires
○ a ***postcondition***, indicated by the keyword effects

Invariants

An ***invariant*** is a condition that is guaranteed to be ***true*** during code execution.

For example, SLList with Sentinel Node has the following invariants:
○ sentinel instance variable always points to a sentinel node
○ the first node, if it exists, is always at sentinel.next
○ size instance variable is always the total number of items added



In-Class Quiz 1

- Given the following specification :

```
static int find(int[] arr, int val)
  requires: val occurs exactly once in arr
  effects:  returns index i such that arr[i] == val
```

select the **legal** behavior that you can then implement `find` with:

☐ if `arr` is empty, return `0`

☐ if `val` occurs twice in `arr`, set all values in `arr` to zero, then throw an exception

☐ if `val` does not occur in `arr`, pick random `index`, set value at `index` to `val`, return the `index`

☐ if `arr[0]` is `val`, continue search, if found another return the index; otherwise return `0`

In-Class Quiz 2

- Given the following specification :

```
static boolean isPalindrome(String word)
  requires:  word contains only alphanumeric characters
  effects:   returns true if and only if word is a palindrome
```

which line of the Javadoc comment is ***problematic***:

```
/**
 * Check if a word is a palindrome.                                        (1)
 * A palindrome is a sequence of characters                                (2)
 * that reads the same forwards and backwards.                             (3)
 * @param String word to check, must contain only alphanumeric characters  (4)
 * @return true if and only if word is a palindrome                        (5)
 */
```