

CSE 331: More Testing

```
/**
 * @requires 0 <= lo <= hi <= a.length
 * @returns *some* index i with lo <= i < hi and a[i] == x,
 *         or -1 if no such index exists
 */
public static int indexOf(int[] a, int lo, int hi, int x) {
    if (!(0 <= lo && lo <= hi && hi <= a.length)) {
        throw new IllegalArgumentException("Violated precondition");
    }
    for (int i = lo; i < hi; i++) {
        if (a[i] == x) {
            return i;
        }
    }
    return -1;
}
```

A *code coverage* heuristic measures how much of the code is exercised by the tests.

- Statement coverage: percentage of statements executed at least once
- Branch coverage: percentage of branches executed at least once
- Loop coverage: branch + each loop should be executed 0/1/many times
- Path coverage: percentage of paths through the program executed at least once

100% coverage is not always achievable. For example: _____

Examples showing difference between statement/branch/path coverage:

```
static int min(int a, int b) {
    int r = a;
    if (a <= b) {
        r = a; // !
    }
    return r;
}

static int quadrant(int x, int y) {
    int q = 1;
    if (x < 0) {
        q = 2;
    }
    if (y < 0) {
        q = q + 1; // !
    }
    return q;
}
```

Test suite: _____

Test suite: _____

Methods are hard to unit test when:

- _____
- _____
- _____

Defensive programming: checking properties you “know” to be true at run time.

Fail fast (early) and fail friendly (with a good message)

How to signal a problem to your client:

- Special value: null, -1, 0, NaN, etc. Pros/cons: _____
- Throw an unchecked exception. Pros/cons: _____
- Throw a checked exception. Pros/cons: _____