Testing

James Wilcox and Kevin Zatloukal

February 2025

Our first rules supersede all others:

- **Exhaustive Testing**: for functions with 10 or fewer *allowed* inputs, test every input.
- **Disallowed inputs**, either by the types or the specification, do not require testing.
- Test each function **individually**. Assume the other functions it calls work properly.

If a function has more than 10 allowed inputs, then the tests must meet the following requirements:

- There are at least two tests.
- **Statement Coverage**: every statement reachable by some allowed input is reached by some test.
- **Branch Coverage**: every conditional, both of whose branches are reachable by some allowed input, has both branches reached by some tests.
- **Loop Coverage**: every loop has tests that cause the loop body to execute 0, 1, and many times. The same rules apply to **recursion**: the tests must cause 0, 1, and many recursive calls.

While *not required* in class, the following heuristics are also important in practice:

- Make sure that every argument is tested with at least two different values.
- If the code contains a conditional of the form `a <= b` (or `<`, `>`, `>`), test the two closest numbers on each side of the dividing line ("boundary") that separates inputs going to the two branches.[1]
- If the code contains a conditional of the form `P || Q` (or `P && Q`), then test a case where `P` is true and `Q` is false, and a case where `P` is false and `Q` is true. (This is in addition to the branch coverage requirement that you also test a case where the entire conditional is true/false.)
- Test special allowed values that commonly lead to bugs: null, undefined, 0, NaN, [], etc.

[1] Always look at the **specification**, not the code, to decide the correct output for each test.