Three dimensions for comparing specs:

deterministic: a single possible output? or a set of legal outputs (***underdetermined***)?
declarative: characterize what the output should be? or explicitly say how to compute the output (***operational***)?
strong: does the spec have a small set of legal implementations, or a large set?

A specification S2 is stronger than or equal to a specification S1 if:
○ S2's precondition is weaker than or equal to S1's, and
○ S2's postcondition is stronger than or equal to S1's for the states that satisfy S1's precondition and it's safe to replace S1 with S2 in your program

Diagraming Specification:
when S2 is stronger than S1:

- Consider the following specification

```
static int findCanBeMissing(int[] a, int val)
  requires: nothing
  effects:  returns index i such that a[i] == val,  or  -1 if no such i
```

and compare with:

```
static int findOneOrMore,FirstIndex(int[] a, int val)
  requires: val occurs at least once in a
  effects:  returns lowest index i such that a[i] == val
```

- which is true about their ***precondition*** and ***postcondition*** ?

- ○ `find`$^{\texttt{CanBeMissing}}$ has a weaker precondition and a stronger postcondition compared to `find`$^{\texttt{OneOrMore,FirstIndex}}$
- ○ `find`$^{\texttt{CanBeMissing}}$ has a stronger precondition and a weaker postcondition compared to `find`$^{\texttt{OneOrMore,FirstIndex}}$
- ◉ `find`$^{\texttt{CanBeMissing}}$ has a weaker precondition and a weaker postcondition compared to `find`$^{\texttt{OneOrMore,FirstIndex}}$

In-Class Quiz 2

- Which of the following is a sign of a good specification?

  - [x] the specification is declarative
  - [ ] the specification is operational
  - [ ] the specification is as strong as possible
  - [ ] the specification is as weak as possible
  - [ ] the implementation is allowed to ignore invalid arguments in the specification
  - [x] the implementation is allowed to use different algorithms depending on the arguments
  - [ ] the specification is making use of the reader's knowledge of the implementation