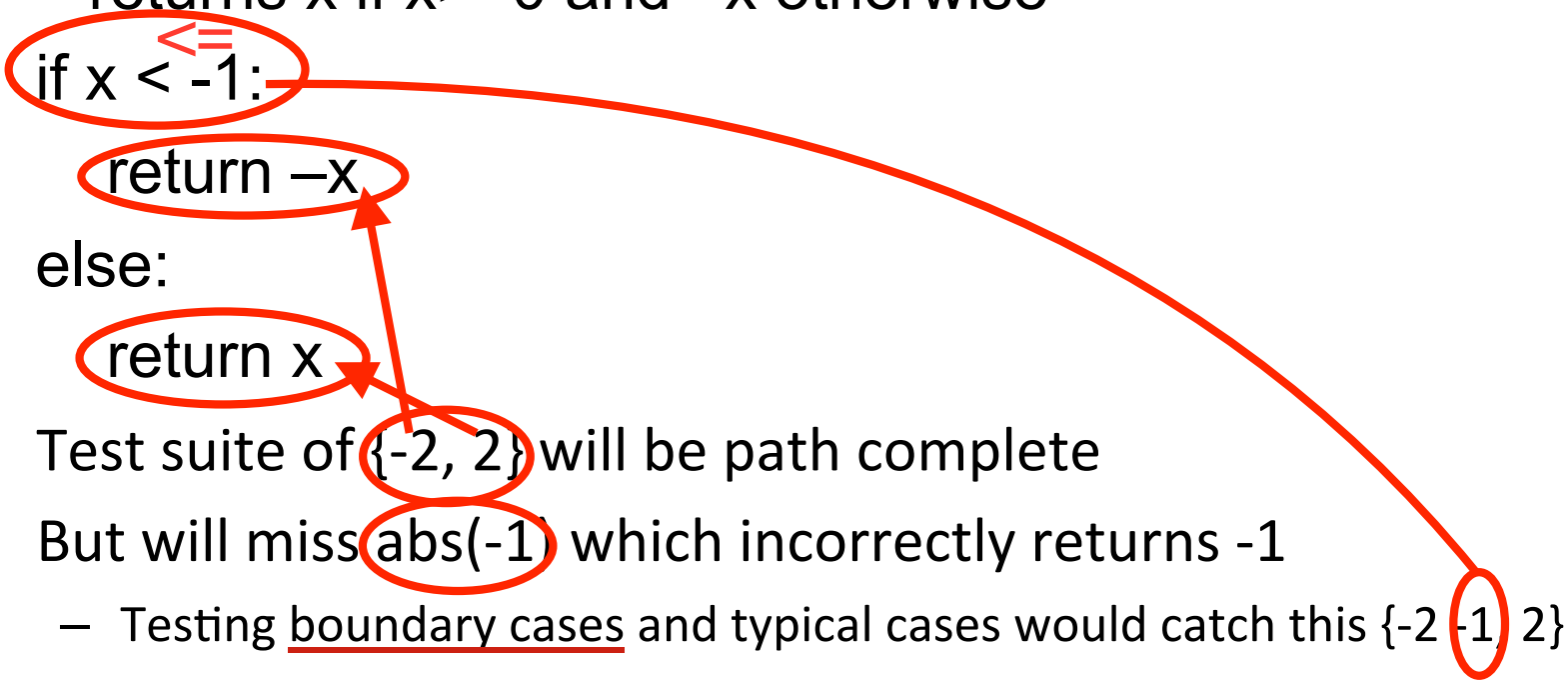
Glass-box Testing

- Use code directly to guide design of test cases
- Glass-box test suite is **path-complete** if every potential path through the code is tested at least once
  - Not always possible if loop can be exercised arbitrary times, or recursion can be arbitrarily deep
- Even path-complete suite can miss a bug, depending on choice of examples

Example

```
def abs(x):
    """Assumes x is an int
       returns x if x>=0 and –x otherwise"""
    if x < -1:
        return –x
    else:
        return x
```



- Test suite of {-2, 2} will be path complete
- But will miss abs(-1) which incorrectly returns -1
  - Testing boundary cases and typical cases would catch this {-2 -1 2}

Rules of thumb for glass-box testing

- Exercise both branches of all if statements
- Ensure each except clause is executed return
- For each for loop, have tests where:
  - Loop is not entered
  - Body of loop executed exactly once
  - Body of loop executed more than once
- For each while loop,
  - Same cases as for loops
  - Cases that catch all ways to exit loop
- For recursive functions, test with no recursive calls, one recursive call, and more than one recursive call