

## Requirement Violation

### Description

The Solidity `require()` construct is meant to validate external inputs of a function. In most cases, such external inputs are provided by callers, but they may also be returned by callees. In the former case, we refer to them as precondition violations. Violations of a requirement can indicate one of two possible issues:

1. A bug exists in the contract that provided the external input.
2. The condition used to express the requirement is too strong.

### Remediation

If the required logical condition is too strong, it should be weakened to allow all valid external inputs.

Otherwise, the bug must be in the contract that provided the external input, and one should consider fixing its code by making sure no invalid inputs are provided.

### Example:

*Code:*

```
1 pragma solidity ^0.4.25;
2
3 contract Bar {
4     Foo private f = new Foo();
5     function doubleBaz() public view returns (int256) {
6         return 2 * f.baz(0);
7     }
8 }
9
10 contract Foo {
11     function baz(int256 x) public pure returns (int256) {
12         require(0 < x);
13         return 42;
14     }
15 }
```

*Explanation:*

Instead of the require function statement, the uint data type will be used to resolve the condition on Line 12 and Line 6 will no longer be used to verify invalid input. It will stop wasting the transaction's gas value.