Comment: The following code snippets are all the same, but the names of variables are different.









……

Comment: there are anomaly injection fragments with issues that prevent the injected anomalies from being activated…..

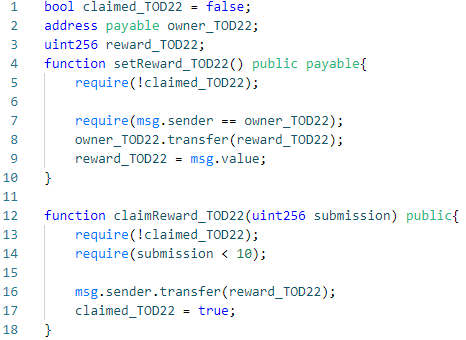


Fig.2 a: An example of incapability of SolidiFI to accurately inject and precisely label anomalies. The original purpose of the code snippet is to inject a transaction order dependence anomaly into the contract by switching the execution order between two functions *setReward\_TOD22* and *claimReward\_TOD22*. If *claimReward\_TOD22* is executed before *setReward\_TOD22,* the statement in line 5 will throw an exception. However, since SolidiFI does not assign a suitable value to the variable *owner\_TOD22*, it causes that *owner\_TOD22* keeps the default initial value (0x0) and the *require-statement* in line 7 throws an exception. This eventually results in the statements in lines 8 and 9 turning into dead code and the executing function *setReward\_TOD22* becoming useless.