Object Model Violation: Just one of the equals() and hashCpde() Defined Mitigation SOP

# Development Mitigation SOP

Java objects are expected to obey a number of invariants related to equality. One of these invariants is that equal objects must have equal hashcodes. In other words, if a.equals(b) = = true then a.hashCode() = = b.hashCode(). Failure to uphold this invariant is likely to cause trouble if objects of this class are stored in a collection. If the objects of the class in question are used as a key in a Hashtable or if they are inserted into a Map or Set, it is critical that equal objects have equal hashcodes.

# Defense Against [DEFECT]

The FindBugs documentation recommends the following simple “starter” implementation of hashCode(). It is highly inefficient, but it will produce correct results. If you do not believe that hashCode() is important for your program, consider using this implementation.

Public class halfway() {

public boolean equals(Object obj){

…

}

}equals(Object obj){

…

}

}

## Explanation

The previous class overrides equals() but not hashCodes().

## Recommendation

The hashCode method was implemented.

public class halfway() {

public Boolean equals(Object obj) {

…

}

public int hashCode() {

assert false : “hahsCode not designed”;

return 42; //any arbitrary constant will do

}

}

**Resources**

1. [D. H. Hovermeyer, FindBugs User Manual](https://wiki.sei.cmu.edu/confluence/display/java/MET09-J.+Classes+that+define+an+equals%28%29+method+must+also+define+a+hashCode%28%29+method)
2. [MET09-J. Classes that define an equals() method must also define a hashCode() method, CERT](https://wiki.sei.cmu.edu/confluence/display/java/MET09-J.+Classes+that+define+an+equals%28%29+method+must+also+define+a+hashCode%28%29+method)