

## *Aim: what is the contract between hashCode() and equals()*

Parent class of all Java Objects: OBJECT Class.

Methods :

getClass() , hashCode() , wait() , toString() , clone() , equals() , finalize() , notify() , notifyAll() .

## hashCode() and equals() methods

hashCode() and equals() methods have been defined in Object class which is parent class for java objects. For this reason, all java objects inherit a default implementation of these methods.

Java hashCode()

Object class defined hashCode() method like this:

```
public int hashCode() {  
    // TODO return the hashCode ;  
}
```

Java equals()

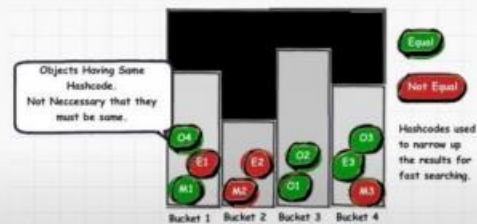
Object class defined equals() method like this:

```
public boolean equals(Object obj) {  
    return (this == obj);  
}
```

## The Contract

The contract between equals() and hashCode() is:

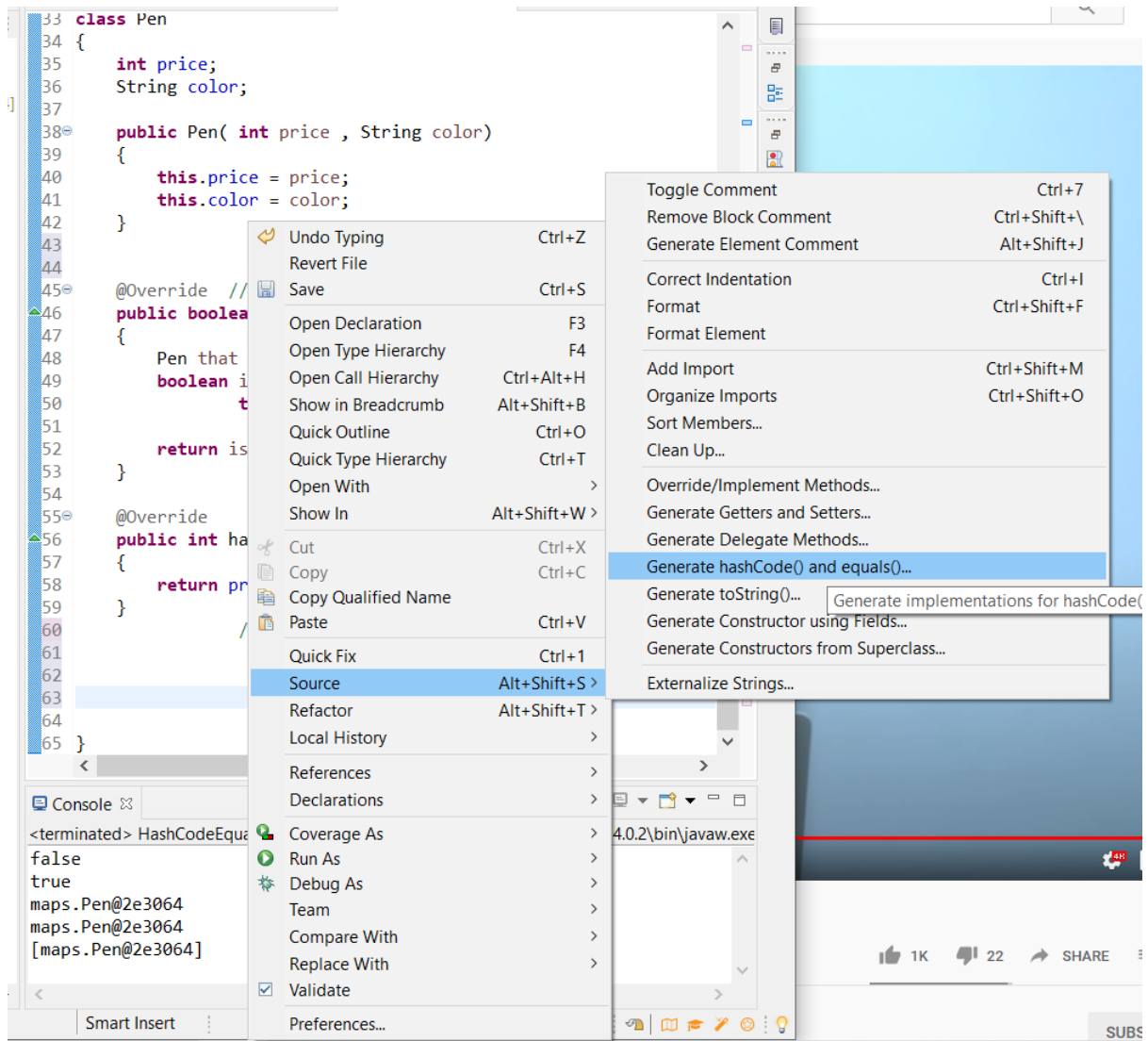
- 1) If two objects are equal, then they must have the same hash code.
- 2) If two objects have the same hash code, they may or may not be equal.



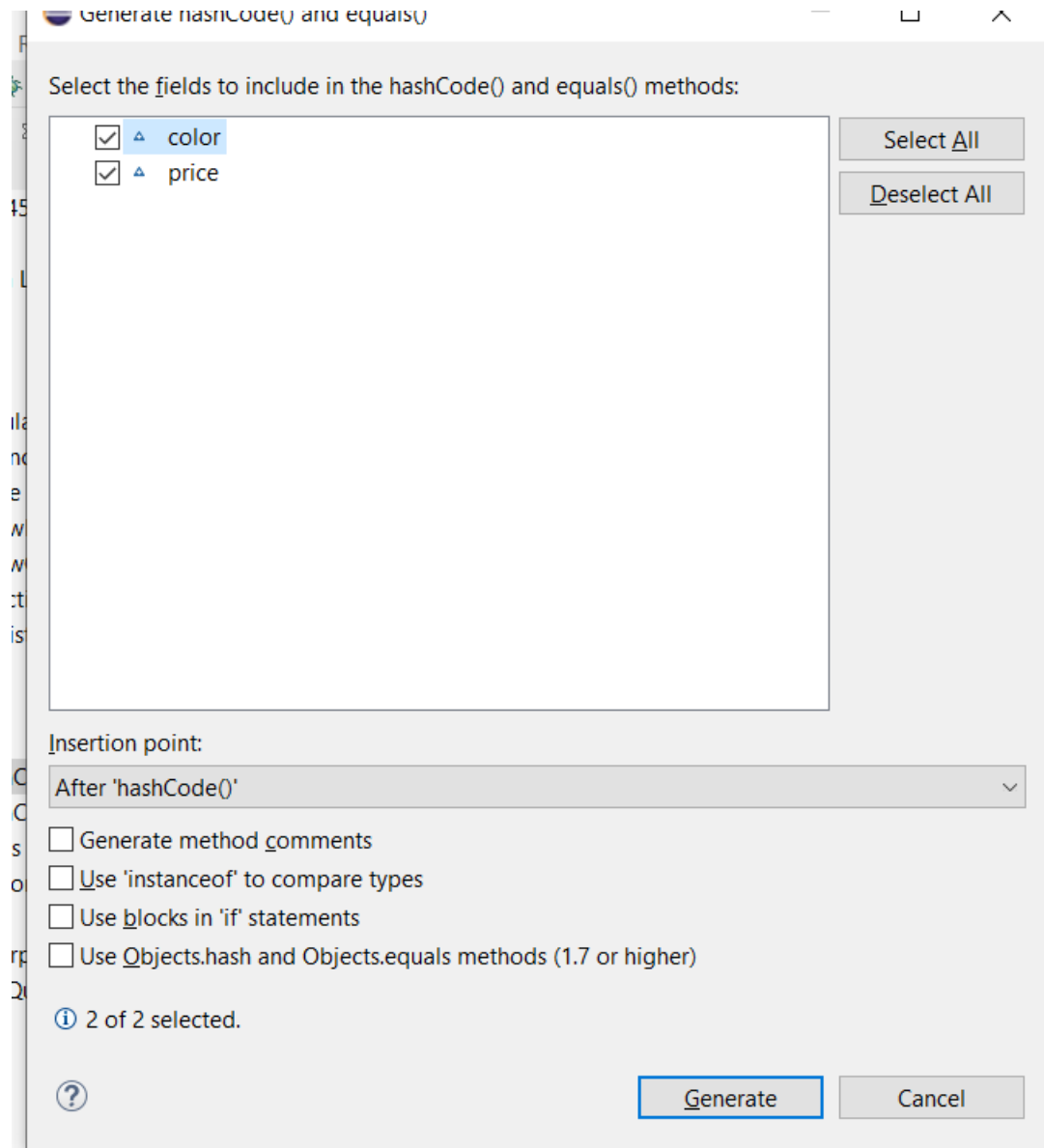
## Best Practices

1. Always use same attributes of an object to generate hashCode() and equals() both.
2. equals() must be *consistent* (if the objects are not modified, then it must keep returning the same value).
3. Whenever a.equals(b), then a.hashCode() must be same as b.hashCode().
4. If you override one, then you should override the other.

***Aim: what is the contract between hashCode() and equals()***



***Aim: what is the contract between hashCode() and equals()***



***Aim: what is the contract between hashCode() and equals()***

```
        @Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Pen other = (Pen) obj;
    if (color == null) {
        if (other.color != null)
            return false;
    } else if (!color.equals(other.color))
        return false;
    if (price != other.price)
        return false;
    return true;
}

@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((color == null) ? 0 : color.hashCode());
    result = prime * result + price;
    return result;
}
```