Integer Comparator

What's wrong with the following code:

```java
Comparator<Integer> naturalOrder = (i, j) -> (i < j) ? -1 : (i == j ? 0 : 1);
```

Can you write a buggy program that demonstrates the bug?

What would be the output of the following code snippet:

```java
import java.util.Comparator;
import java.util.TreeSet;

public class IntegerComparator {
    public static void main(String[] args) {
        final Comparator<Integer> naturalOrder = (i, j) -> (i < j) ? -1 : (i == j ? 0 : 1);

        TreeSet<Integer> treeSet = new TreeSet<>(naturalOrder);

        Integer num1 = Integer.valueOf(1234);
        Integer num2 = Integer.valueOf(1234);

        treeSet.add(num1);
        treeSet.add(num2);

        System.out.println("TreeSet size: " + treeSet.size());
    }
}
```

Would the output change if we set `num1` and `num2` to `Integer.valueOf(123)` ?

Click here to expand...

Bug: When comparing `Integer` objects, we should use the `equals()` method to compare their values correctly.

The `Integer.valueOf()` method caches `Integer` objects for small values in the range -128 to 127.