Are these 3 boolean expressions equivalent? 1. `!(x == 0 || x >= 1)` , 2. `!(x == 0) && !(x >= 1)` , 3. `(x != 0) && (x < 1)`

Save & Run | 7/14/2024, 4:14:27 PM - 3 of 3 | Download | Show CodeLens | Reformat | Pair?

```
1 //Code by Akshat Garg
2 public class EquivalentExpressions
3 {
4     public static void main(String[] args)
5     {
6         int x = -1;
7         System.out.println(!(x == 0 || x >= 1));
8         x = 0;
9         System.out.println(!(x == 0 || x >= 1));
10        x = 1;
11        System.out.println(!(x == 0 || x >= 1));
12
13        // add print statements for expressions in #2 and #3
14        x = -1;
15        System.out.println(!(x == 0) && !(x >= 1));
16        x = 0;
17        System.out.println(!(x == 0) && !(x >= 1));
18        x = 1;
19        System.out.println(!(x == 0) && !(x >= 1));
20
21        // to see if they are equivalent when x = -1, 0, and 1.
22        x = -1;
23        System.out.println((x != 0) && (x < 1));
24        x = 0;
25        System.out.println((x != 0) && (x < 1));
26        x = 1;
27        System.out.println((x != 0) && (x < 1));
28
29
30    }
31 }
32
33
```

```
true
false
false
true
false
false
true
false
false
```

Result	Expected	Actual	Notes
Pass	true	true	Checking that code contains (x != 0) && (x < 1)
Pass	true	true	Checking that code contains !(x == 0) && !(x >= 1)
Pass	true	true	Checking that code has been changed

You got 3 out of 3 correct. 100.00%

Activity: 3.6.4.1 ActiveCode (challenge3-6-booleanExpr)

3.7.4. Programming Challenge : Tracing Code

What will the following code print out? Trace through the code by drawing diagrams of what is going on in memory like the figures above, and then show the values of s1, s2, s3, s4 and the output after each line of code. Remember that you can use trace tables to track the values of variables as they change throughout a program. To trace through code, write down a variable in each column in a table and keep track of its value throughout the program as you go through it line by line.

```
String s1 = null;
String s2 = new String("hi");
String s3 = new String("hi");
String s4 = new String("bye");
if (s1 == null)
{
    s1 = s2;
}
if (s1 == s2)
{
    System.out.println("s1 and s2 refer to the same object");
}
if (s2 == s3)
{
    System.out.println("s2 and s3 refer to the same object");
}
if (s3 == s4)
{
    System.out.println("s3 and s4 refer to the same object");
}
if (s1.equals(s2) && s2.equals(s3))
{
    System.out.println("s1, s2, s3 are equal");
}
```

3-7-10: Write your tracing table here that keeps track of s1, s2, s3, s4 and the output.

```
s1: null    "hi"
s2: "hi"    "hi"
s3: "hi"    "hi"
s4: "bye" "bye"
        true    true    false   false    true && true

will print "s1 and s2 . . . " and "s1, s2, s3 . . . "  only
```

Save

Instructor's Feedback

Your answer has not been saved yet!

Activity: 3.7.4.1 shortanswer (challenge3-7-tracingStrings)

Good practice in tracing the origins of the objects and referencing different aliases.