Activity 5: Boolean Expressions

The primitive data type `boolean` has two values: `true` and `false`. Boolean expressions are built using *relational operators* and *conditional operators*.

Model 1 Relational Operators

When you leave out the semicolon in the Interactions pane, it will display the *value* of the expression you have entered. Fill in the below table (the first four rows are completed for you).

Interactions	Value displayed	Relational operator
`int three = 3`	none	none
`int four = 4`	none	none
`System.out.println(four)`	4	none
`three > four`	false	>
`boolean isLarger = three > four`		
`System.out.println(isLarger)`		
`three == four`		
`three < four`		
`three <= four`		
`three = four`		
`three == four`		

Questions (10 min) Start time: _____

1. Examine the fifth line of Java code in the above model.

   a) What three actions are performed in this single line of code?

   b) Write two lines of code, ending with semicolons, that would perform these same actions (but in two lines instead of a single line).

2. List the four unique boolean expressions used in the model.

3. The `!=` operator means "not equals". Give an example of a boolean expression that uses `!=` and evaluates to false.

4. Explain why the same boolean expression `three == four` resulted with two different boolean values in this Model.

5. What is the difference between = and == in Java?

6. List the six relational operators that can be used in a boolean expression. (Five have been used so far, but you should be able to guess the sixth.) Explain briefly what each one means.

Model 2 Conditional Operators

Boolean expressions may also use conditional operators to implement basic logic. Relational operators are always executed first, so there is generally no need for parentheses.

Operator	Meaning
!	Not
&&	And
||	Or

If all three operators appear in the same expression, Java will evaluate the `!` first, then `&&`, and finally `||`. If there are multiples of the same operator, they are evaluated from left to right.

Example Variables:

```
int a = 3;
int b = 4;
int c = 5;
boolean funny = true;
boolean weird = false;
```

Example Expressions:

```
a < b && funny
a < b && b < c
c < a || b < a
funny && a < c
!funny || weird
```

Questions (15 min) Start time: _____

7. What are the values (true or false) of the example expressions?

8. Give different examples of boolean expressions that:

   a) uses a, b, and !, and evaluates to false

   b) uses b, c, and !, and evaluates to true

   c) uses any variables, but evaluates to false

   d) uses any variables, but evaluates to true

9. Using your answers from the previous question, write the boolean expression `p && q` where `p` is your first answer and `q` is your second answer.

   a) Your expression:

   b) Result of `p && q`:

10. Complete the following table:

`p`	`q`	`p && q`	`p || q`	`!p`
false	false			
false	true			
true	false			
true	true			

11. Using the values in the model, give the result of each operator in the following expression. In other words, show your work as you evaluate the code in the same order that Java would.

```
!(a > c) && b > c
```

	Operator	**Expression**	**Result**
1st	>	`a > c`	false
2nd			
3rd			
4th			

12. Add parentheses to the boolean expression from the previous question so that the `&&` is evaluated before the `!`. Then remove any unnecessary parentheses.

a) Expression:

b) New result:

13. Review the table from #10 for evaluating `&&` and `||`. Looking only at the `p` and `&&` columns, when is it necessary to examine `q` to determine how `p && q` should be evaluated?

14. Review the table from #10 for evaluating `&&` and `||`. Looking only at the `p` and `||` columns, when is it necessary to examine `q` to determine how `p || q` should be evaluated?

15. In Java, `&&` and `||` are *short circuit* operators, meaning they evaluate only what is necessary. If the expression `p` is more likely to be true than the expression `q`, which one should you place on the left of each operator to avoid doing extra work?

a) left of the `&&` expression:

b) left of the `||` expression:

Model 3 Case Study: Panic Attack

Frank was behind in his programming assignment. He approached Martin to see if he could get some help. But he was so far behind and so confused that Martin just gave him his code with the intent that he would "just look at it to get some ideas."

In the paraphrased words of Frank: "I started the assignment three days after you put it up. But then other assignments came in and I started on them too. I felt like I was chasing rabbits and began to panic. It was already past the due date and I got really scared. That's when I went to Martin to see if he could help." Frank copied much of the code and turned it in as his own.

Questions (8 min) **Start time:** _____

16. Which, if any, of the students were at fault? Why?

17. Which specific Honor Code violations occurred?

18. What should Martin have done in this situation?

19. What options did Frank have besides cheating?

Model 4 Case Study: Oops!

Emily was working in the lab on her programming assignment. She finished the program, submitted it, and went on to do some other work. Shortly thereafter, she left the lab.

Another student, Kyle, was working nearby. He knew that she had successfully submitted the assignment, and he had not been able to get his to work properly. When Emily left, he noticed that she had not logged out of her computer. He moved to her workstation, found the work under her Documents directory, and copied it onto his memory stick. He then logged out, logged in as himself, and copied the code onto his Desktop where he modified the program a bit, then successfully submitted it.

Questions (8 min) Start time: _____

20. Which, if any, of the students were at fault? Why?

21. Which specific Honor Code violations occurred?

22. What should Emily have done in this situation?

23. What options did Kyle have besides cheating?