Activity 5: Boolean Logic

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

Content Learning Objectives

After completing this activity, students should be able to:

- Recognize the value of developing process skills.
- Evaluate boolean expressions with relational operators (<, >, <=, >=, ==, !=).
- Explain the difference between assignment (=) and equality (==) operators.
- Evaluate boolean expressions that involve comparisons with &&, `||`, and !.

Process Skill Goals

During the activity, students should make progress toward:

- Evaluating complex logic expressions based on operator precedence. (Critical Thinking)

Facilitation Notes

Model 1 is ultimately about process skills and should help with student buy-in. If you are using the Role Cards, have students look at the definitions on the reverse side. Each activity targets specific "process skill goals" from these categories.

Model 2 mentions DrJava, but it can be replaced with another IDE or Java REPL. Give students about three minutes to fill in the table without using a computer. Then show them the actual results interactively (or in an example program) on the projector.

When reporting out, ask students to explain what *expressions* are and how they differ from *statements*. Reinforce what it means to *evaluate* an expression (i.e., compute a single value) versus *execute* a statement (i.e., run an entire line of code).

During Model 3, explain that the variables *p* and *q* are often used to represent logic values in discrete math. Make sure students understand that ! is a *unary* operator, and that `&&` and `||` are *binary* operators.

Model 1 What Employers Want

"What do employers look for when they are seeking new college graduates to take on jobs? According to NACE's *Job Outlook 2016* survey, they are looking for leaders who can work as part of a team." http://www.naceweb.org/s11182015/employers-look-for-in-new-hires.aspx

Attributes employers seek on a candidate's resume

	Attribute	**% of respondents**
1.	Leadership	80.1%
2.	Ability to work in a team	78.9%
3.	Communication skills (written)	70.2%
4.	Problem-solving skills	70.2%
5.	Communication skills (verbal)	68.9%
6.	Strong work ethic	68.9%
7.	Initiative	65.8%
8.	Analytical/quantitative skills	62.7%
9.	Flexibility/adaptability	60.9%
10.	Technical skills	59.6%

Questions (10 min) Start time: ________

1. What is the relationship between the top two attributes employers seek?

Leadership implies working with other people, most likely as part of a team. Working in teams is essential to developing leadership skills.

2. How is communication (written and verbal) related to problem-solving?

Solving problems in teams involves talking to other people and trying different approaches. Writing solutions down is necessary to solidify the details and share them with others.

3. As a team, come up with a short description/example of each attribute.

1. leading a group of people
2. geting along well with others
3. writing and reading effectively
4. finding solutions creatively
5. speaking and listening effectively
6. self-motivated to work hard
7. acting or taking charge early
8. analyzing data and reasoning
9. being able to handle change
10. computer/technology literacy

4. Which of these skills do you expect to develop in this course? Why?

Ideally, all of them. Students will definitely learn technical computer programming skills. But working in teams provides the opportunity to develop many other employable skills.

Model 2 Relational Operators

Some IDEs (including DrJava) have an "Interactions" feature that allows you to enter a single line of code and see what it does. If you type an *expression* like 1 + 2 and leave off the semicolon, it will display the resulting *value* (i.e., 3). In the table below, predict what values will be displayed and identify the relational operator. 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`	none	>
`System.out.println(isLarger)`	false	none
`three == four`	false	==
`three < four`	true	<
`three <= four`	true	<=
`three = four`	4	none
`three == four`	true	==

Questions (15 min)

Start time: ________

5. List the four unique *boolean expressions* used in Model 2.

`three > four` `three == four` `three < four` `three <= four`

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

a) What three actions are performed in this single line of code? It declares the variable `isLarger`, compares the values of `three` and `four`, and assigns the result to `isLarger`.

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

```
boolean isLarger;
isLarger = three > four;
```

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

`5 != 5` is false (because they *are* equal)

8. Explain why the same boolean expression `three == four` resulted with two different values in the table.

The line `three = four` assigned the value of four to three, making the two variables equal. They started out not being equal, but they ended up with the same value.

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

The = operator assigns a value to a variable, and the == operator compares two values.

10. 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.

< is less than

<= is less than or equal to

> is greater than

>= is greater than or equal to

== is equal to

`!=` is not equal to

Model 3 Conditional Operators

Boolean expressions use conditional operators to implement basic logic. If all three operators appear in the same expression, Java will evaluate `!` first, then `&&`, and finally `||`. If there are multiples of the same operator, they are evaluated from left to right. Relational operators are evaluated before `&&` and `||`, so there is generally no need for parentheses.

Operator	Meaning
`!`	not
`&&`	and
`||`	or

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 (20 min) Start time: ________

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

12. Give different examples of boolean expressions that:

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

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

   c) uses any variables, but evaluates to false weird

   d) uses any variables, but evaluates to true funny

13. Using your answers from the previous question, write the boolean expression `p && q` where `p` is your answer to step a) and `q` is your answer to step b).

   a) Your expression: `!(a < b) && !(b > c)`

   b) Result of `p && q`: false (no matter what)

14. Complete the following table:

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

15. Using the values in Model 3, 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	`!`	`!`false	true
3rd	`>`	`b > c`	false
4th	`&&`	true `&&` false	false

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

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

b) New result: true

17. Review the table from #14 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?

You only need to look at `q` when `p` is true. If `p` is false, you know the expression will be false.

18. Review the table from #14 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?

You only need to look at `q` when `p` is false. If `p` is true, you know the expression will be true.

19. 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: `q` — if it's false, then `p` won't be evaluated

b) left of the `||` expression: `p` — if it's true, then `q` won't be evaluated

20. What is the result of the following expressions?

a) `1 + 0 > 0 && 1 / 0 > 0` java.lang.ArithmeticException: / by zero

b) `1 + 0 > 0 || 1 / 0 > 0` true