

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)



Model 1 What Employers Want

The following data is from the *Job Outlook 2019* survey by the National Association of Colleges and Employers (NACE). A total of 172 organizations responded to the survey.

Attributes Employers Seek on a Candidate's Resume

Attribute	% of respondents
Ability to work in a team	78.7%
Analytical/quantitative skills	71.9%
Communication skills (verbal)	67.4%
Communication skills (written)	82.0%
Detail-oriented	59.6%
Initiative	74.2%
Leadership	67.4%
Problem-solving skills	80.9%
Strong work ethic	70.8%
Technical skills	59.6%

Source: <https://www.naceweb.org/talent-acquisition/candidate-selection/>

Questions (10 min)

Start time: _____

1. What are the top three attributes that employers look for on a resume?
 - #1:
 - #2:
 - #3:
2. Describe the process your team used to answer to the previous question.
3. How is communication (written and verbal) related to problem solving and teamwork?
4. How does the team-based learning approach in this class help you develop these skills?

Model 2 Relational Operators

In Model 1, you determined the top three attributes by comparing percentages. We can declare variables to represent these percentages in Java:

```
double written = 82.0; // Communication skills (written)
double problem = 80.9; // Problem-solving skills
double teamwork = 78.7; // Ability to work in a team
```

In the table below, predict the result of each expression and identify the operator (if any). The first three rows are completed for you.

Expression	Result	Operator
written	82.0	none
written > problem	true	>
problem < teamwork	false	<
82.0 < written		
82.0 > written		
82.0 == written		
problem == written		
teamwork == problem		
teamwork = problem		
teamwork == problem		
problem		
teamwork		

Questions (15 min)

Start time: _____

5. A *relational operator* compares two values; the result is either `true` or `false`. Identify the three relational operators used in the table above.

6. Explain why the same expression `teamwork == problem` resulted with two different values in the table.

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

8. The != relational operator means “not equals”. Give an example of a boolean expression that uses != and evaluates to false.

9. The >= relational operator means “greater than or equal to”. Give an example of a boolean expression that uses >= and evaluates to true.

10. Java has six relational operators. Only five have been mentioned, but you should be able to guess the sixth. List all six below, and explain briefly what each one means.

Model 3 Conditional Operators

Boolean expressions, like `written > problem` and `teamwork < 75.0`, can be combined using the *conditional operators*:

Operator	Meaning
!	not
&&	and
	or

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.

Example Variables:

```
double initiative = 74.2;
double analytical = 71.9;
double workEthic = 70.8;
boolean hired = true;
boolean fired = false;
```

Example Expressions:

```
analytical < initiative && fired
workEthic < 71.0 && 71.0 < initiative
initiative < 70.0 || workEthic > 70.0
fired || workEthic < 50.0
hired && !fired
```

Questions (20 min)

Start time: _____

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

12. Give different examples of boolean expressions that:

a) uses `initiative`, `analytical`, and `!`, and evaluates to false

b) uses `analytical`, `workEthic`, and `!`, and evaluates to true

c) uses any variables, and evaluates to false

d) uses any variables, and evaluates to true

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

a) Your expression:

b) Result of `p && q`:

14. Complete the following table, which explores all possible values for `p` and `q`:

<code>p</code>	<code>q</code>	<code>p && q</code>	<code>p q</code>	<code>!p</code>
false	false			
false	true			
true	false			
true	true			

15. Using the values in Model 2, give the result of each operator below. In other words, show your work as you evaluate the code in the same order that Java would.

```
!(written < teamwork) && problem > teamwork
```

	Operator	Expression	Result
1st	<code><</code>	<code>written < teamwork</code>	false
2nd			
3rd			
4th			

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:

b) New result:

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?

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?

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:

b) left of the `||` expression:

20. What is the result of the following expressions?

a) `1 + 0 > 0 && 1 / 0 > 0`

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