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. Point out that "Technical skills" is ranked tenth, after all the others.

At the beginning of Model 2, you might instruct students to check their work using JShell or a similar tool. Give students about three minutes to fill in the table without using a computer. If computers are not available, show them the actual results interactively on the projector. Have teams discuss any of their predictions that were incorrect.

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. Refer back to Model 1 to reinforce the importance of students communicating with each other as a team.

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: Communication skills (written)
- #2: Problem-solving skills
- #3: Ability to work in a team

2. Describe the process your team used to answer to the previous question.

We searched the table for the highest three percentages and then made sure that we all had the same answers. (Note that searching/sorting involves comparison, which is the next model.)

3. How is communication (written and verbal) related to problem solving and teamwork?

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.

4. How does the team-based learning approach in this class help you develop these skills?

POGIL activities provide an opportunity to develop these skills during class. Ideally, students will learn these skills both in the classroom and in other activities like clubs and internships.

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`	false	<
`82.0 > written`	false	>
`82.0 == written`	true	==
`problem == written`	false	==
`teamwork == problem`	false	==
`teamwork = problem`	80.9	=
`teamwork == problem`	true	==
`problem`	80.9	none
`teamwork`	80.9	none

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.

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

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

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

8. The != relational 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)

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

`5 >= 5` is true (because they *are* equal)

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.

< is less than	> is greater than	== is equal to
<= is less than or equal to	>= is greater than or equal to	!= is not equal to

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?

false true true false true

12. Give different examples of boolean expressions that:

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

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

c) uses any variables, and evaluates to false fired

d) uses any variables, and evaluates to true hired

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: `!(initiative < analytical) && !(analytical > workEthic)`

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

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

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 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	<	`written < teamwork`	false
2nd	!	`!false`	true
3rd	>	`problem > teamwork`	true
4th	&&	`true && true`	true

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

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