If-Else and Boolean Expressions

Decide who will be what role for today's activity. List the names of your team members next to their role.

Manager:
Spokesperson:
Quality Control/Recorder:
Process Analyst:

If you have only three people, one may serve as both manager and spokesperson. If you have five team members, then assign two people to process analyst.

Exercise 1: Clarity With If-Else (10 min)

Imagine we are working on a game application that requires us to determine when the player has won. Players win when their score *exceeds* 100 points. Here are five possible implementations (assume that `win` is declared as a boolean variable.)

```
// A
if (points > 100) {
   win = true;
} else if (points < 100){
   win = false;
}
```

```
// B
if (points > 100) {
   win = true;
} else if (points <= 100){
   win = false;
}
```

```
// C
if (points > 100) {
   win = true;
} else {
   win = false;
}
```

```
// D
if (points > 100) {
   win = true;
}
if (points < 100){
   win = false;
}
```

```
// E
win = points > 100;
```

1. Complete the table below with the value of win that will result from each of the implementations above. Every entry should be `true` or `false`.

	points == 99	**points == 100**	**points == 101**
A			
B			
C			
D			
E			

2. Which of these five implementations are correct?

3. Of the correct implementations, which is easiest to understand? Why?

Exercise 2: Logical Operators (15 min)

Boolean expressions may include the logical operators !, &&, and || to express logic that is either `true` or `false`. If all three operators appear in the same expression, Java will evaluate the ! first, then &&, and finally ||. If there are multiples of the same operators (e.g., two && operators), they are evaluated from left to right.

```
int a;
int b;
int c;

a = 3;
b = 4;
c = 5;
```

Questions

4. Give an example of an expression using logical operators that:

a) uses a and b, and evaluates to true

b) uses b and c, and evaluates to true

c) uses any variables, but evaluates to false

5. Using your answer to #4a and #4b, write the boolean expression p && q, where p is your answer to #4a and q is your answer to #4b.

a) Give the expression here:

b) Give the result of p && q (either `true` or `false`).

6. Complete the following table for the && (AND) and || (OR) operators.

p	q	p && q	p || q
false	false		
false	true		
true	false		
true	true		

7. Using the variables above, give the result of each operation in the following expression. In other words, show your work as you evaluate the following expression in the same way that Java would.

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

	Operator (! > &&)	**Expression**	**Result (true or false)**
1st			
2nd			
3rd			
4th			

8. Suppose you are unsure about the order of operations for ! and &&. Add parenthesis to the boolean expression from the previous question in different ways, so that:

a) the ! is evaluated before the &&

b) What does your expression in (a) evaluate to?

c) the && is evaluated before the !

d) What does your expression in (c) evaluate to?

9. Review the table from #6 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?

10. The Java && and || are *short circuit operators*, meaning they evaluate only what is necessary. If one expression is more likely to be true than the other expression, which one should you place on the left of each operator? (Assume here that the goal is to write code that executes as quickly as possible.)

a) left of the && expression

b) left of the || expression

Exercise 3: Applying Logical Operators (10 min)

The following two methods use different criteria for determining a bonus:

```
public static int bonusA(int performance,
                         int seniority) {
   int bonus;
   bonus = 0;

   if (performance > 30) {
      if (seniority > 10) {
         bonus = 100;
      }
   }
   return bonus;
}
```

```
public static int bonusB(int performance,
                         int seniority) {
   int bonus;
   bonus = 0;

   if (seniority != 10) {
      if (performance > 30) {
         bonus = 100;
      }
      if (seniority > 10) {
         bonus = 100;
      }
   }

   return bonus;
}
```

11. Rewrite both methods to use logical operators instead of nested if-statements. Each method should have only one if-statement.

Team Reflection

Process Analyst: Provide one piece of positive feedback to each member of the team.

Quality Control: Summarize for your team the main concepts you learned in this week's activity. Make sure everyone's worksheets are correct and complete.

Everyone: Review the job description on your role card. Evaluate privately on how well you performed in your role. Write down what you need to improve:

Acknowledgements

This activity is based on materials developed by Helen Hu, modified by Chris Mayfield and Nathan Sprague, and licensed under CC BY-NC 4.0 International.