Lab 06: Basic Control Flow
CSE-020-01, Fall 2019 **Points: 20**

Overview

This week we will use the Boolean expression we learned to create from the last lab in controlling the flow of the program. This allows us to produce more sophisticated programs that behave differently based on the inputs or some other conditions.

If Statement

The executable statement inside `if` is only executed when the boolean expression is true. For example, assume we have the following code:

```
if (a < b)
      System.out.println(a + "is less than " + b);
```

So, for the case where `a` is `4` and `b` is `10`, the output will say "`4 is less than 10`". However, for the case where `a` is `10` and `b` is `4`, `println` won't be executed so we will not see anything on the output.

If-else Statement

A more complex control flow statement is a branching decision where it takes either one path or another. An example of that type of decision is done with `if-else` statement. For example, assume we have the following code:

```
if (a < b)
      System.out.println(a + "is less than " + b);
else
      System.out.println(a + "is not less than " + b);
```

So, for the *true* case where `a` is `4` and `b` is `10`, the output will say "`4 is less than 10`". For the *false* case where `a` is `10` and `b` is `4`, the output will say "`10 is not less than 4`".

Block

We need something for when we need to execute more than one statement inside a conditional statement. There is a construct called Block that you have been using which allows us to put in any number of statements. The Block is denoted by a pair of curly brackets `{ }` . You may have noticed the same curly brackets after Class definition and `main` definition.

```
if (a < b) {
      System.out.println(a + "is less than " + b);
      z = b - a;
}
```

A block can be applied to the else clause as well, as the following examples shows:

```
if (a < b) {
      System. out.println(a + "is less than " + b);
      z = b - a;
      System.out.println("The difference is " + z);
} else {
      System.out.println(a + "is not less than " + b);
      z = a - b;
      System.out.println("The difference is " + z);
}
```

Nested If-else Statement

If we want to do multiple branch of decisions and only 2 is not enough, then we apply a sequence or nested `if-else` statements to model that decision process.

```
        if (a < b) {
              System.out.println(a + "is less than " + b);
              z = b - a;
              System.out.println("The difference is " + z);
        } else if (a == b) {
              System.out.println("There is no difference");
        } else {
              System.out.println(a + "is greater than " + b);
              z = a - b;
              System.out.println("The difference is " + z);
        }
```

This code -- a sequence of `if-else` statements -- checks for three cases where first is "`a is less than b`", second is "`a equal to b`" and the last is implicitly "`a is greater than b`". We don't need `a > b` check because two numbers can only have 3 relations, and if the first two are false, then the last must be true.

Getting Started

After starting Eclipse, create a new project called **Lab20_6**. Import `BooleanIf.java`, `BooleanIfElse.java`, `Division.java` and `Manipulatev2.java` files from the Lab 06 assignment page into the project and load them.

Part 1: Find and Replace

`BooleanIf.java` – Replace all the `true` in the `if` statements with the appropriate relational check between `a` and `b` as indicated by the `println` statement inside the `if` statement.

`BooleanIfElse.java` – Replace all the `true` with the appropriate relational check between `a` and `b` as indicated by the `println` statement inside the `else if` statement. Notice the warning by Eclipse about dead code due to the use of `true` in the `else if` statements. It should go away after you replace the `true` with the correct expression.

Part 2: (Assessment) Logic Check

Create a Word document or text file named `Part2` that contains answers to the following:

1. What is the maximum number of relations that can be true at once in `BooleanIf` for some `a` and `b` input?
2. What is the minimum number of relations that can be true at once in `BooleanIf` for some `a` and `b` input?
3. What does Dead code mean?
4. What is the maximum number of output statements showing the relation between `a` and `b` for `BooleanIfElse`?
5. What is the value of `a` and `b` where `BooleanIfElse` will print out "`is less than or equal to`"?

Part 3: Fill-in Code

`Division.java` – Enter 0 for denominator and look at the output of the program. Now make the change so you check for the case where denominator is equal 0 and don't perform the division operation. Instead print out the message "`Dividing by zero!`". If the denominator is a nonzero number, then print out the result of the division as the original code.

`Manipulatev2.java` – The program is asking for 4 inputs from the user. First two inputs are the two numbers to use in calculation. These two numbers are defined as `double`s so all values can be used. Then it further asks the user to select the operation they want by choosing the appropriate number. User will press 1 for addition, 2 for subtraction, 3 for multiplication. 4 for division and 5 for remainder. Finally, it asks the user to pick the data type to use for printing the result of the operation. You want to replace all the `true` with the right expression. Also, for all the parts with `// fill in`, you need to put in the executable statement that belongs there.

What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

Before you submit, make sure you have done the following:

- Attached the file named `Part5` containing answers to the assessment questions.
- Attached the replaced `BooleanIf.java`, and `BooleanIfElse.java` files.
- Attached the filled in `Division.java` and `Manipulatev2.java` file.
- Filled in your collaborator's name (if any) in the "Comments…" text-box at the submission page.

Also, remember to demonstrate your code to the TA or instructor before the end of the grace period.