Lab 07-2: More Control Flow
CSE-020-01, Fall 2019 — Points: 20

Overview

We will continue to get more practice using control statements in this week along with the type of errors you have been encountering.

Run-Time Errors

Eclipse cannot catch these errors as they only appear when the programming is running. When we ask the user to input something, we have been assuming they will enter the correct data. However, it is not always the case and we can detect some of these errors using control statements.

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:

```java
if (a < b)
        System.out.println(a + "is less than " + b);
else
        System.out.println(a + "is greater than or equal to " + 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 greater than or equal to 4`".

Getting Started

After starting Eclipse, create a new project called **Lab20_7**. Import `BooleanIf.java` and `Number.java` files from the Lab 07-2 assignment page into the project and load them.

Part 1: Rearrange – BooleanIf.java

In Lab 6 we discovered that there will always be 3 if statements that will be true for any combination of two numbers (`a` and `b`). `BooleanIf` right now has 6 independent if statements to print out the true relations between the two numbers. It is actually redundant to have 6 checks as some expressions are related – if one is true then the other must be false and vice versa. We have 3 pairs of relations, which can be checked with 3 simple `if-else`. Modify `BooleanIf.java` to use only 3 `if-else` statements while preserving the same output.

Part 2: Number.java

Program asks the user to input a number between 1-26. Put in code to check if the input number is within the valid range (1-26) and print `Character # of the alphabet is *`, where `#` is the input number and `*` is a corresponding character with the following logic:

- If the input number is `1` then your code must output "`Character 1 of the alphabet is A`"
- If the input number is `2` then your code must output "`Character 2 of the alphabet is B`"
- … so on …
- If the input number is `26` then your code must output "`Character 26 of the alphabet is Z`"
- If the number is invalid, then your code must output "`Invalid Input! Number is outside of acceptable range.`"

Part 3: (Assessment) Logic Check

Enter the following values by running `Number.java` after your modification and explain the behavior in a Word document or text file named `Part3`.

- 1

- 20
- 1000
- 500000000000
- -500000000000
- 12,345
- 789.543
- -0

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 `Part3` containing answers to the assessment questions.
- Attached the modified `BooleanIf.java` and `Number.java` files.
- 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.