

# Conditionals

## Purpose:

The purpose of this lab is practice using conditionals in implementing methods.

Do either I or II below.

## I. The class *Counter*:

- Copy the files from `~labCourse/Labs/Lab10/counter10a/` into a subdirectory of your *Java* directory named *counter10a*.

We want to create a new *Counter* class with the following functionality:

- A client is able to specify the amount by which the counter will be incremented or decremented, and specify the value to which the counter is reset.
- A client is able to specify the minimum and maximum values of the counter;
- The counter “wraps around” when its value would be greater than the maximum or smaller than the minimum.

For instance, if the maximum count is 9999, the minimum 0, the current count 9998, and the increment amount 2, incrementing the counter will leave the count at 0. if the increment amount is 5, incrementing the counter will leave the count at 3.

The wrapping is with respect to minimum or maximum. Suppose the minimum count is -10, the max 10, the increment and decrement amounts both 2. If a counter with count of 10 is incremented, it ends with a count of -9. If a counter with a count of -10 is decremented, it would end up with a count of 9.

Furthermore, the increment and decrement amounts cannot exceed the counter value range, so there is no possibility of the counter “wrapping more than once” on a single increment or decrement operation. For instance, if the maximum count is 10 and the minimum count is -10, increment and decrement amounts must be greater than or equal to 0, and less than or equal to 21. If the increment is 21, incrementing the counter does not change the count.

The specifications for the class can be found in *here*.

- The file *Counter.java* that you copied into *counter10a* contains part of the implementation. Complete the implementation of the class *Counter* using conditionals. (Don't use the remainder operator, “%”.)
- Create a test plan, and test your implementation by running *counter10a.CounterGUIStart*.
- Submit the source of your *Counter* class, your test plan, and a summary of your test results as directed by your lab instructor.

## II. The class *TextCounter*:

- Copy the files from `~labCourse/Labs/Lab10/counter10b/` into a subdirectory of your *Java* directory named *counter10b*.

We want to create a counter that counts characters. More specifically, we will implement a class *TextCounter* that counts characters and also keeps track of the number of letters and digits it has counted.

The class should provide the following functionality:

- a query *letters* which returns the number of letters counted;
  - a query *digits* which returns the number of digits counted;
  - a query *characters* which returns the number of characters counted;
  - a query *isLetter* which is given a character and returns *true* if the character is a letter, *false* otherwise. For the purpose of this exercise, we limit ourselves to letters in the standard Latin alphabet, 'a' through 'z'. A letter can be upper or lower case.
  - a query *isDigit* which is given a character and returns *true* if the character is a digit, *false* otherwise.
  - a command *increment*, which is given a character and counts it appropriately;
  - a command *increment*, which is given a *String* and appropriately counts all the characters in the *String*.
- The file *TextCounter.java* that you copied into *counter10b* contains part of the definition of the class. Complete the specification.
  - Implement the class in the following way. Rather than simply using an `int` variable to keep the counts, use three instances of the class *counter10b.Counter*. This class is included in the files you copied and is specified *here*.

You should know that the lower case letters, the upper case letters, and the digits are ordered. That is, 'a' < 'b' < 'c' ... < 'z', and no characters besides lower case letters are in the range 'a' to 'z'. Similarly for uppercase characters and digits: 'A' < 'B' < ... < 'Z' and '0' < '1' ... < '9'.

- Create a test plan and test the class. To test the class, run *counter10b.TextCounterTUIStart*. This is a text-based interface rather than a graphical interface. Terminate the test run by keying *Control-D* on a line by itself.

Create a script of your test as follows. When you are ready to conduct your test, type the command `script` at the command line. From this point on, everything that is displayed in the window will also be written to a file named *typescript*. When you are finished with your test, type *Control-D* on a line by itself. This will terminate the script, and close the file *typescript*. You can then print the file.

- Submit the source of your *TextCounter* class, your test plan, and a listing of your *typescript* file showing the results of your test, as directed by your instructor.