# CSC-151 Midterm Part 2 Practice

## StringUtils

You are to write a class *StringUtils*. This class will have the following *public static* methods:

### String padLeft(String current, int numOfChars, char paddingChar)

This method returns a *String* that is the value of *current* prepended by a *String* that consists of a series of *paddingChar* characters that has length *numOfChars*.

### String padRight(String current, int numOfChars, char paddingChar)

This method returns a *String* that is the value of *current* appended with a *String* that consists of a series of *paddingChar* characters that has length *numOfChars*.

### String zapRight(String current, int numOfChars)

This method returns the value of *current* with the last *numOfChars* characters removed.

### String zapLeft(String current, int numOfChars)

This method returns the value of *current* with the first *numOfChars* characters removed.

## StringPlay

You are to write a class *StringPlay* that has only a *main* method. This class contains all interaction with the user.

The *main* method

* Uses a *Scanner* to accept user input
* Maintains a “*current*” *String*
* Uses a sentinel-controlled loop to accept multiple user commands, or exit
* The program displays a message asking the user to enter one of the following commands
  + e – enter a new value for *current*
  + pl – *padLeft* asks the user for the number of characters to add to the start of the *current* string, and the character to use for the padding
  + pr – *padRight* asks the user for the number of characters to add to the end of the *current* string, and the character to use for the padding
  + zl – *zapLeft* asks the user for the number of characters to remove from the start of the *current* string
  + zr – *zapRight* asks the user for the number of characters to remove from the end of the *current* string
  + x – exits the program
* The loop body performs its actions using the appropriate method from *StringUtils*.
* After the action is completed, the loop displays the *current* string and prompts for another command.

## Notes

Use *Scanner’s* *next* method for the “e” command; we will limit our user input to have no spaces or other whitespace. Similarly, padding characters will not be whitespace. You may assume valid user input.

## Sample Execution

Enter e(enter current string), pl(pad left), pr(pad right), zl(zap left), zr(zap right), x(exit)

*e*

Enter the new current String: *abcd*

The current string is: abcd

Enter e(enter current string), pl(pad left), pr(pad right), zl(zap left), zr(zap right), x(exit)

*zr*

How many chars to zap? *1*

The current string is: abc

Enter e(enter current string), pl(pad left), pr(pad right), zl(zap left), zr(zap right), x(exit)

*pl*

How many chars to add? *4*

What char to use?

*+*

The current string is: ++++abc

Enter e(enter current string), pl(pad left), pr(pad right), zl(zap left), zr(zap right), x(exit)

*pr*

How many chars to add? *6*

What char to use?

*x*

The current string is: ++++abcxxxxxx

Enter e(enter current string), pl(pad left), pr(pad right), zl(zap left), zr(zap right), x(exit)

*zl*

How many chars to zap? *10*

The current string is: xxx

Enter e(enter current string), pl(pad left), pr(pad right), zl(zap left), zr(zap right), x(exit)

*x*