Echo

Part of the UNIX design philosophy is to develop programs that *do one thing well.* One program found on any UNIX-based system is the command-line tool `echo`. You will write a class called `Echo` that, for the most part, mimics this tool. In essence, the program prints a string fed in as a command-line argument, followed by a newline. For example:

```
[user@notnotbc.org]$ java Echo 'Hello, World!'
Hello, World!
[user@notnotbc.org]$
```

Note: the single-quotes (`'`) are important, due to the behavior of the shell (the program interpreting commands you type into your terminal). You may assume any command-line arguments passed to your program will be wrapped in quotes.

Furthermore, the `Echo` class will not interpret escape seqeunces by default. For example:

```
[user@notnotbc.org]$ java Echo 'Hello,\n\tWorld!'
Hello,\n\tWorld!
[user@notnotbc.org]$
```

However, the traditional `echo` tool provides command-line options to override the default behavior. You will only implement a subset of these command-line options:

```
 -n   :   do not output trailing newline
 -e   :   enable interpretation of backslash escapes

 If -e is in effect, the following sequences are recognized:

 \n   :   newline
 \t   :   horizontal tab
```

These command line options must come *before* the argument to be echoed. For example, the following illustrates the `-n` option:

```
[user@notnotbc.org]$ java Echo -n 'Hello, World!'
Hello, World![user@notnotbc.org]$
```

Next, the `-e` option:

```
[user@notnotbc.org]$ java Echo -e 'Hello,\n\tWorld!'
Hello,
      World!
[user@notnotbc.org]$
```

In conjunction:

```
[user@notnotbc.org]$ java Echo -e -n 'Hello,\n\tWorld!'
Hello,
      World![user@notnotbc.org]$
```

You will find the Java 8 String API useful – in particular, the `replace( )` method.
Hint: `switch` statements are very useful in parsing command-line arguments. Additionally, your environment on `notnotbc.org` is equipped with the `echo` command-line tool. You may

experiment with that in order to get a better understanding of how to implement your own version of the tool.

Remember that the name of the file must be the same as the name of the class with `.java` appended to it. In other words, your file should be named `Echo.java`.

Submit your assignment by copying the `Echo.java` file into your personal submissions directory. There is no need to place the `Echo.class` file in the directory.