CSCI 3302 Programming Assignment 02 (100 Points)

Due: Sep 19, 12:00 AM

OBJECTIVES:

- Demonstrate a basic understanding of Java programming.
- Demonstrate a basic understanding of recursion.

ASSIGNMENT ASSISTANCE:

- This homework assignment is due before the date and time specified above.
- This assignment is restricted to individual effort. As per our syllabus, the use of AI is prohibited. You may not receive help from any other person except the instructor or the AARC (help from the AARC must be well documented!).
- Any resource used (other than Dr. Becnel or the course text) must be documented in the code (as comments) detailing the source and describing exactly what was learned and how that information was used. Submissions will be severely penalized if copied in part or in whole from any source.

PROBLEM DESCRIPTION:

In this assignment, you will solve problems using recursion.

1. Create a Java class named `Compressor` in a class file called `Compressor.java`.

2. The `Compressor` class has no fields.

3. The class contains a **`public`** method called `repeatString` that returns a value of type `String`. This method has two parameters: one of type `int` and one of type `String`.

    a. The method retursn the given string repeated the given number of times.
    Examples:
    ```
    repeat("dog", 0), returns ""
    repeat("cat",  1), returns "cat"
    repeat("hat", 2), returns "hathat"
    ```
    b. **You are not allowed to use iteration within this method in your submission. You must use recursion.**

    c. Throw an illegal argument exception if the integer is negative (< 0).

4. The class contains two **`public`** methods called `compress` and `decompress`. Each method returns a value of type `String`. Each method has one parameter of type `String` called `text`.

    a. The String for `decompress` will be the format
    char, int, char, int, char, int, car, int,… char, int
    where the integer tells us the number of times to repeat the preceding character.

Example:

```
decompress("w4e3") returns "wwwweee"
decompress("w1t2f3") returns "wttfff"
decompress("o2m3g0") returns "oommm"
decompress("s10") returns "ssssssssss"
```

The `decompress` method must use recursion, i.e. call itself. Hint: process one letter and number combination, then recursively call the method on the substring with the letter and number combination removed.

Note: You may use iteration to find the number after a character, as you will not know the number of digits in the number ahead of time. This may be a good place for a helper method.

b. The String for `compress` will simply be a string of letters. The method takes the string and compresses it by counting repetitions of letters. If a letter x is repeated k times, the Compressor will denote this by xk.
Example:

```
compress("oommm") returns o2m3
compress("EEEEK") returns E4k1
compress("helloooo") returns h1e1l2o4
```

The `compress` method can use either iteration or recursion. It can use only iteration if you prefer.

c. You are NOT responsible for throwing an exception for incorrect input.

5. Your file `Compressor.java` should NOT contain a main method or any extraneous testing code. You can include files with testing code; however, these will not be considered for grading. If you wish to include non-working code for insight into your thought process, make sure to contain it within comment blocks and ensure that the submission successfully compiles.

6. Your program should work in the GitHub codespace (Linux environment) and locally (Windows environment).

HINTS:

1. You may define any additional helper methods that you need. Make sure to declare them as `private`.

2. You may use either head recursion or tail recursion.

3. The `substring` method is the Java String class may be helpful. Look up how to use it in the Oracle docs: https://docs.oracle.com/javase/8/docs/api/java/lang/String.html

4. `Integer.valueOf` will convert a String to an int.

5. The Java Character has useful static methods. In particular, `Character.isDigit` and `Character.getNumericValue` may prove useful. Look these up in the Oracle docs: https://docs.oracle.com/javase/8/docs/api/java/lang/Character.html

SUBMISSION:

- Review the Evaluation below to ensure you have met all the requirements.
- Commit `Compressor.java` to GitHub. Upload a backup copy to D2L.

EVALUATION

Remember to consult Program Requirements.docx

Automatic Deductions:	
Late/Not Submitted	-100
Code not submitted to GitHub	-30
Code does not run/compile	-50
Earn Points for the following:	
Code has a comment header with name, section, date	5 pts
Code organization, structure, and indention are appropriate (SHFT + ALT + F in VS Code)	5 pts
Code is well and meaningfully commented.	5 pts
Appropriate variable and method names that follow Java conventions	5 pts
Instructions correctly followed for fields, class, methods	10 pts
repeatString uses only recursion to provide correct solution	20 pts
compress provides correct solution (partial credit given)	20 pts
decompress uses recursion and provides correct solution	30 pts