CM1203/CM1207 Assessment 3

Lecturer	Dr Matt Morgan
Date Set	Week 8
Submission date	Friday 11th April 2013 (Week 11)
Submission arrangements	An electronic copy must be submitted via Learning Central before 17:00
Maximum Mark	32 (This assessment contributes 10% to CM1203 or 20% to CM1207)

NOTE: Source files to begin this assessment are provided on Learning Central. They can be found at Assessment → Java Assessment 3.

Criteria for assessment

Credit will be awarded against the following criteria:

Functionality To what extent does the program perform the task(s) required by the question.

Design How well designed is the code, particularly with respect to the ease with which it may be maintained or extended. In particular, consideration will be given to:

- Use of appropriate types, program control structures and classes from the core API.
- Definition of appropriate classes and methods.

Ease of use

- Formatting of input/output
- Interaction with the user
- How does the code deal with invalid user input? Will the applications crash for certain data?

Documentation and presentation

- Clear and appropriate screenshots.
- Appropriate use of comments.
- Readability of code.

Requirements

This assessment asks you to implement an application that will shorten a message by abbreviating words where possible. (For example, this might be useful to a Twitter user who is trying to get their tweet to fit into 140 characters.) You are expected to first implement a class that handles the shortening of messages (Q1), followed by a command-line shortener application (Q2) and finally a pre-evaluator application (Q3).

Your class and applications are expected to read an abbreviations text file which provides a list of words and their abbreviations. Each line in the abbreviations text file consists of a word and an abbreviation, separated by a comma; for example:

```
you,u

hello,lo

any,ne

one,1

anyone,ne1
```

An abbreviations file named `abbreviations.txt` has been provided. In this assessment you should use `abbreviations.txt` as the default abbreviations file.

Throughout this assessment you should apply what you have learnt so far during this and previous modules, including, but not limited to:

- information hiding
- appropriate error checking and input validation
- elegance and style
- appropriate code re-use
- investigating and using classes of the Java Core API

For simplicity, you may assume that all messages that will be shortened are in lower case.

Q1

Download the file `Shortener.java`. An object of the `Shortener` class represents a message shortener with a particular set of abbreviations. Complete the implementation of this class according to the requirements given in `Shortener.java`. In addition, `Shortener` should be *mutable*; i.e., it should be possible to change the set of abbreviations being used by setting a different file. You should add the fields and methods that are required for this behaviour.

Marks available: 14

Q2

Download the file `ShortenerUtility.java`. Modify the file to implement a command-line application `ShortenerUtility` that takes a message to be shortened as a single command-line argument and prints out the shortened message to standard output. An example use is:

```
> java ShortenerUtility "hello world! anyone out there? it's lonely here." lo
world! ne1 out there? it's lonely here.
```

Marks available: 6

Q3

Download the file `Evaluator.java`, which extends the `Shortener` class and complete the implementation of this class according to the requirements given in `Evaluator.java`.

Finally, download the file `EvaluatorUtility.java`. Modify the file to implement another command-line application `EvaluatorUtility` that takes a message to be shortened as a single command-line argument and prints out the following information to standard output:

```
> java EvaluatorUtility "hello world! anyone out there? it's lonely here."

Length of input: 48

Number of words: 8

Output: lo world! ne1 out there? it's lonely here.

Length of output: 42

Shortened by: 12.5%
```

Marks available: 12

Submission instructions

You should submit your source code via Learning Central at Assessments → Java Assessment 3. At the beginning of each source file you must include your `full name` and `student number` as comments. You must zip your source files together into one zip archive. The zip file should be named with your student number; for example, if your student number is `0000000`, you should upload a zip file named `0000000.zip`.