CO3095 / CO7095 Assignment 2

This coursework component accounts for 15% of your module mark.

Task

The task is to generate a test set, using the Category Partition Method, to test the functionality within the Java SDK of parsing textual representation of dates and times according to the RFC1123 standard representation.

Within Java, this implementation is invoked by obtaining the static class:
`java.time.format.DateTimeFormatter.RFC_1123_DATE_TIME`.

For our testing, we will focus on the `DateTimeFormatter.parse(CharSequence text)` method.

Start off by consulting the Java documentation for the respective `java.time.format.DateTimeFormatter` methods here:
`http://docs.oracle.com/javase/8/docs/api/java/time/format/DateTimeFormatter.html`

In particular, focus on the specification of the `RFC_1123_DATE_TIME` field description, and any other relevant documentation in the JavaDoc.

Task

Use the category partition method to construct test sets for the this method. This should result in at least 10 categories. Each category should be accompanied by a brief 1-2 sentence justification.

Identify any constraints between the categories, and systematically identify the various test cases that can be derived from the categories.

Use the test cases you have produced to generate a JUnit test class. This should contain 10-15 test cases that, between them, exercise as much of the functionality you have identified within your categories as possible. Each test case should contain a comment highlighting the combination of categories that is being tested, along with an assertion that checks that the output is as expected.

Generate a second JUnit test class. Choose one of the test cases that you produced by hand. For that test case, write a JUnit test method that will automatically generate random inputs (where the inputs satisfy the categories belonging to the test case in question). By default, the test method should generate and execute 100 test cases. This test class does not need to include any assertions or checks on the output.

Submission instructions

Please put the following items into a Zip file (not tar.gz, bzip2, rar, or anything else):

- A PDF file containing details of the categories, and how they can be combined.

- Two Java source code files, containing JUnit tests described above (using version 4 of the JUnit libraries).

Assessment

Assessment will be carried out according to the following factors:

- The appropriateness and justification for the chosen categories (where possible backed up by relevant reference to documentation).
- The appropriateness of the constraints between the categories, and the ability to derive potential test cases that obey these constraints.
- The choice and implementation of the test cases.
- The implementation of the random test setup.