**CASE STUDY: SMS Nightmare**



Ms Lim was using her mobile phone while on a recent trip to Tasmania, and accidentally dropped it down a very high cliff (don’t ask, long story). She’s lost the phone and any potential to retrieve the data from it directly, as there is no way she can get to the bottom of the cliff to get her phone. Luckily, she has been using the cloud backup software that was installed on the phone and has a fairly recent backup of all of her data.

When she received her replacement phone, she went to restore all of her data back onto it, but as she didn’t buy a phone with the same operating system, she found there was no easy way to restore all of her information.

Ms Lim is fairly good at computers, though, and she was able to download the backup from the cloud onto her local computer system. She was able to manually restore her contacts and her photos, but not her SMS messages. She did a bit of poking around and found some CSV files within the backups that seemed to look like her text message history. Her new phone only allows her to import XML files, however, so right now the CSV files are useless to her.

Ms Lim needs you to write a Python program that will open up the CSV files that have her SMS history and then re-save them as XML files of the same name. You do not need to write a user interface, as Ms Lim only needs to use this program once – she just needs a stand-alone program that will run in the Python IDLE screen so she can convert all of her files. She’s provided you with a sample exported CSV file, but needs the program to run on all of the CSV files in any directory she puts the program in.

The functional requirements for what you need to write are described in full on the next page.

This task is marked out of 16. Your marks for this task are allocated as follows:

* 4 marks for CSV-related functionality
* 6 marks for XML-related functionality
* 2 marks for iteration-related functionality
* 2 marks for naming conventions
* 2 marks for internal documentation

Your task files must be submitted to Compass (including all CSV and XML files) by the end of the double period. The Learning Outcome is called “Unit 3 Outcome 1: Programming Modules Part 2”

**NAME:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Convert CSV to XML

Explanation:

This function will open up a single CSV file. The CSV file contains an SMS history and will need to be converted to an XML file of the same name. Only the required elements should be saved to the XML file; the rest of the information should be ignored.

Inputs:

* nameOfFile: the name of the file to convert, including extension (as a String)

Sequence of Operation:

* For the given CSV file, open the CSV file and read in all of the information.
* Out of all of the information, the following elements need to be retrieved:

contact\_name, type, address, body, date, read

* These should be obtained using the index values of the items in each line of the CSV file. For example, contact\_name is index value 0. The information required should be stored in a list.
* Each item in the list should be saved into an XML file using the correct names for the elements.
* The first line of the CSV file provides these names so you know which index values they relate to.
* The name of the XML file should be the same name as the CSV file – just a different file extension (.xml instead of .csv). The XML structure should look like this:

<sms\_history>

<sms>

<contact\_name>Lucy D</contact\_name>

<type>1</type>

<address>61415559292</address>

<body>Yeow =p this is my number haha. (it's Lucy)</body>

<date>1356090000000</date>

<read>1</read>

</sms>

<sms>

<contact\_name>Lucy D</contact\_name>

<type>2</type>

<address>61415559292</address>

<body>Haha, thanks Lucy! :)</body>

<date>1356090000000</date>

<read>1</read>

</sms>

</sms\_history>

Outputs:

* A success message if the file was converted.

### Convert All Files

Explanation:

Loops through the directory the script is run in to convert all CSV files found in the folder to XML.

Inputs:

* None.

Sequence of Operation:

* Gets all of the files from the folder the Python file is in.
* Filters the files to find only those ending in .csv
* Runs Functional Requirement 1.1.1 on each CSV file found.

Outputs:

* A completion message once all files are processed.