OBJECT ORIENTED PROGRAMMING

LABORATORY 9

OBJECTIVES

In this laboratory, you'll get familiar with lambdas, exceptions and streams in C++.
You'll also continue working on your project, and you'll implement the "skeleton" of your application: you'll implement the classes for the three layers (model, controller and presentation), even if some of the functionalities are just stubs for now.

PROPOSED PROBLEMS

1. Modify the stack data structure that you wrote in the previous laboratory to throw an exception in the following cases:

a. When you want to pop an element from the stack, but the stack is empty;
b. When you want the retrieve the element from top of the stack (its first element), but the stack is empty;
c. When you want to push an element on the stack, but the stack reached its maximum capacity (the stack is full).

2. Download the Reddit vaccine myth dataset from kaggle: https://www.kaggle.com/gpreda/reddit-vaccine-myths . This dataset it stored in .csv file format at it contains various posts and comments from Reddit about vaccines. More specifically, it contains the following information:

a. title – the title of the post; only relevant for posts, comments have the title "Comment";
b. score – the score of the post based on impact, number of comments; only relevant for posts;
c. id - unique id;
d. url – the url of post thread;
e. commns_num – the number of comments to this post;
f. created - date of creation;
g. body - relevant for posts/comments - text of the post or comment;
h. timestamp – the timestamp;

Create two classes, one to represent a Date and one to represent a Reddit post (should contain a Date attribute for the creation date).

Parse the Reddit vaccine myths .csv file and store **only** the posts from this file into a std::vector.

Then display this vector into a nice tabular form (use stream manipulators to ensure that the text is properly aligned in the table).

For the Date class, create a method *std::string display(std::string format);* that displays the date in the requested format. Check this post for various date formats that you could use: https://en.wikipedia.org/wiki/Date_format_by_country. Use a *ostringstream* for this: http://www.cplusplus.com/reference/sstream/ostringstream/ostringstream/ .

Finally, use the *find_if* method from the standard library to find **all** the posts that contain the word "coronavirus" in their title. Use a lambda to achieve this.

You can find more details about .csv files here: https://en.wikipedia.org/wiki/Comma-separated_values

PROJECT

Implement the Repository, Controller and UI layers for your application.
By the end of this week, your application should fully implement the operations for adding and item and for displaying items (all the items from the repository and the items with some properties). Write method stubs for removing and editing an item.

At the repository level, you should have two functions to load and save your data from/into a .csv file, respectively. When the application starts, load the data from a .csv file into your repository. When the application ends, dump all the data from the repository into the same .csv file.

Test and document all the classes that you wrote.