Assignment Tutorial Letter 2023

Advanced Programming

COS3711

Year module

Computer Science Department

Assignment 1 Questions

BARCODE

Assignment 1

1. Introduction

There are two parts to this assignment.

- Part A is to be submitted for marking.
- Part B is for extra practice of examinable concepts and should not be submitted.

Note:

- Please use `CMake` (and not `qmake`) when setting up your assignment projects.
- Qt Designer should not be used to design user interfaces, and you are expected to manually set up GUIs to ensure that you properly handle memory using Qt's parent-child functionality.

Good programming practices should be followed.

- Follow standard naming conventions: class names start with a capital letter, variable and function names start with a lowercase letter, using camelCase for names made up of multiple words.
- Ensure consistent code layout and use of blank lines.
- Use forward class declarations in header files.
- Use initialiser lists in constructors.
- Have proper GUI management: setting cursor focus, sequential tabbing, clearing input widgets (like text input fields being cleared and spin boxes being returned to some default value), and enabling and disabling buttons as appropriate.
- Provide appropriate user feedback.
- Your code should build and run without any warnings.

2. Part A (To be submitted)

Question 1

Write an application that can be used to keep a list of staff member details. Each staff member data should include the following:

- A name,
- A birthdate, and
- A staff appointment type (one of permanent, part-time, or contract, stored and handled as an `enum`).

Include all necessary getter and setter functions. An additional getter can be used to return the appointment type as a `QString`.

The application should allow a user to enter details via a GUI. Also, once a staff member's details are entered, they should be saved to a list/container in the application. The user should be able to write this list to file (in any acceptable, human-readable format) when needed.

Ensure that you adhere to basic design principles and avoid any anti-patterns in the design of your solution.

Question 2

Taking Question 1 a step further, make all the getters and setters (including any `toString()` functions) that give access to the data members name, birthdate and staff type `private`. Then make the necessary changes so that your application uses reflective programming techniques to access the data members in the object that holds the staff data when writing to file.

Note that you cannot assume that you know beforehand how many properties there are, what they are called, or of what type they are. You will need to find a way of handling the `enum`, though as you (and in this case you can accept that you know that there is an `enum`, but you need to find its values in some way); also, you will have to move away from using a `QString` for the `enum` here.

HINT: You will need to use `QVariant`.

Question 3

Use Qt's model/view framework to store and display music album information. The following information should be stored in your own custom model (that inherits from `QAbstractTableModel`):

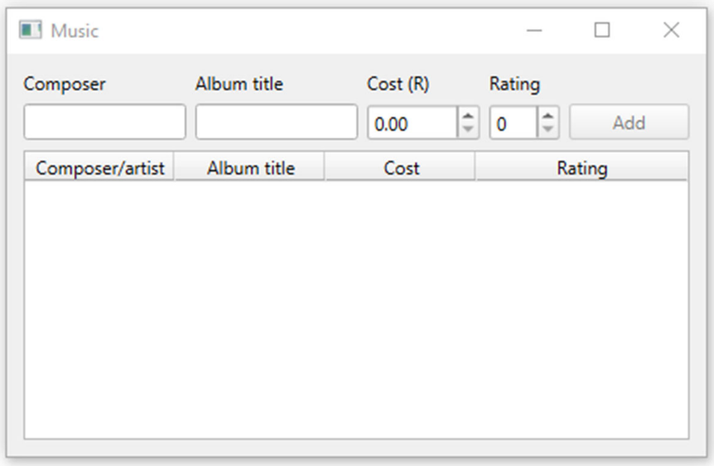
- composer,
- album name,
- cost (stored as a `double`, and
- a rating (out of 100, store as an `int`).

Your sub-classed model should be a well behaved, editable model (and you will need to add all the necessary functions to your model). The information should be displayed in a standard `QTableView`.

The following functionality should be included.

- There should be a header row.
- The user should be able to add data to the model.
- The replacement value should always display the cents (even as .00).

Below is a possible example of the interface.



3. PART B (For self-assessment; not to be submitted)

Question 4

Extend Question 2 by allowing for the addition of dynamic properties.

To use the application, you will first need to add some objects to the container. Then, when an appropriate button is clicked, the new application should ask the user which object to use, what the property name should be, and its value.

When the user selects to write the data to file, the dynamic properties should also be written to the file.

You can see from the screenshots below of the file contents that a dynamic property has been added to only one staff member.

```
name: Donald Duck; date: 2023-01-01; type: Permanent;
name: Mickey Mouse; date: 2022-12-25; type: Contract;
```

```
name: Donald Duck; date: 2023-01-01; type: Permanent;
name: Mickey Mouse; date: 2022-12-25; type: Contract; Creator: Walt Disney;
```

Question 5

Extend the application in Question 3 so that it can do the following as well.

- Use a delegate to display the rating as a horizontal bar. This column should take up all the remaining space available.
- The user should be able to delete a row of data.