Assignment Tutorial Letter 2026

Advanced Programming

COS3711

Year module

Computer Science Department

Assignment 1 Questions

BARCODE

Question 1

For this question you need to ensure that you design the appropriate classes needed to address the specification below.

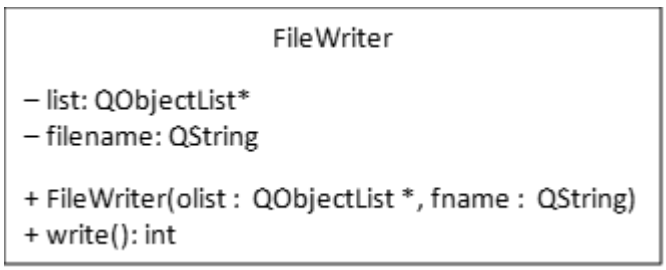
Create a console application to handle vehicle details. Considering the required functionality that is required and OOP design principles (avoiding anti-patterns and having minimal redundant code), create and implement the appropriate classes necessary to achieve the following.

- A vehicle needs to have a model and a year (which should be a reasonable value). There are basically 2 types of vehicles: passenger vehicles (that can carry a specified number of passengers) and transport vehicles (that have a set carrying capacity expressed in kilograms). Vehicles that are created without the necessary details should have default values.
- Ensure all appropriate getters and setters are included.
- Include a function that can output the details of the vehicle.
- Use Qt's parent-child facility to implement a list of vehicles.
- Test your solution by creating some passenger and transport vehicles (at least one of which is created using the default constructor), adding them to the list, and then outputting the values in the list to the console.

Question 2

This question tests the concepts of reflection and meta-objects.

Extend the application you wrote in question 1 by adding a class named FileWriter as described in the UML class diagram below:



The write() function should use the meta-object of the items in list to write each item's type of vehicle, properties and their values to a text file named in filename. You should use reflective programming techniques to do this, and should not use the getters or the toString() functions to extract the type of vehicle and the data from each item in the list. Thus you cannot assume that you know what properties an item holds. The function should return the number of items that were written to file.

From main(), create an instance of the FileWriter class, passing the list of vehicles (as a QObjectList) and a file name, and then write the list to file. Display the number of records printed to the console.

Question 3

This question focuses on model-view programming.

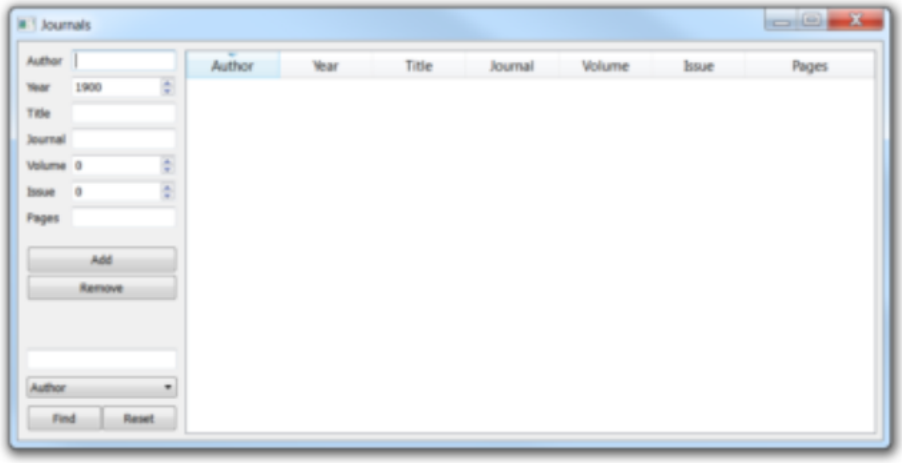
A reference for a journal article needs the following data: the author, year of publication, article title, journal name, volume and issue numbers, and the pages on which the article can be found. Using a QStandardItemModel and an appropriate view, create a database for journal articles. The user should be able to do the following.

- Sort on any column in view.
- Add data to the database.
- Remove data from the database.
- Filter the database on any of the fields (except the page numbers). The user should be able to provide a wildcard filter and select which field to filter the database on. Then only fields that meet the requirements should be displayed. There should also be an option to clear the filter so that all records are displayed again.

Additionally, a year value later than the current year should not be allowed (and you can decide how you will deal with situations where this is attempted). Also, if a reference is older than 10 years, highlight the row in red; if within the last 5 years, highlight in green. Note that you cannot assume that the current year is 2025.

Note further that if the year should be changed in the view, it should not be allowed to be later than the current year, and the highlight colour of the row should still change appropriately.

Below is an example of a possible interface.



Question 4

This question looks at validating user input.

Extend the application developed in question 3 so that only valid input is allowed in the cases where strings have been used to input data. Remember that such validation should take place both when data is input and when it is edited.

Note the following.

- An author's name could be names like "SZ Mbanjwa-Mhlana". You should assume that no full stops will be used with author initials, and that no other special characters should be allowed.
- Assume (for this exercise) that article and journal titles can allow letters of the alphabet ("Journal", for example), numbers, and spaces. However, do not allow a word in the title to contain both alphabetic characters and numbers in the same word – that is, 3D should not be allowed.
- The page numbers should be of the form "12-15" or "121 - 155".