|  |  |  |
| --- | --- | --- |
| **PRACTICAL EXAMINATION** | **Code:** | **PRO192 – Su 2024** |
| **Duration:** | **85 minutes** |

**Title**

Develop Candidate Management OOP application with the following components and requirements.

**Requirements**

*Guidelines for program design*: following the MVC pattern is required.

* The Candidate and CandidateList classes act as models, the CandidateManager class contains the main method which acts as the controller, and the Menu and CandidateListView act as the view.
* Other classes can be added optionally.

**1.** (**2 marks**) Define a class Candidate with the following attributes and methods

**Attributes:**

* id: Stores a unique ID to identify the candidate.
* name: Stores the name of the candidate.
* dateOfBirthday: Stores the age of the candidate.
* grades: Stores the grade of the candidate.
* email: Stores the email of the candidate.

**Methods:**

• Constructors, getters, and setters.

• override toString.

**2.** Define the CandidateList class manages the list of candidates and CandidateListView manages inputting and outputting information in the candidate list.

The CandidateList class needs to contain the following methods:

a**. (2 mark)** Loads the list of candidates from a file "candidates\_input.txt" and stores in List <Candidate>.

The format of the file "candidates\_input.txt" is as follows:

|  |
| --- |
| *C1, John Doe, 01-01-2024, 7.5, johndoe@gmail.com*  *C2, Jane Smith, 05-01-2024, 9.0, janesmith@gmail.com*  *C3, Robert Johnson, 10-01-2024, 5.0, robertjohnson@gmail.com*  *C4, Emily Brown, 15-01-2024, 7.15, emilybrown@gmail.com*  *C5, Michael Wilson, 20-01-2024, 9.1, michaelwilson@gmail.com* |

b**. (2 mark)**

* (1 marks) Adds a new candidate to the list, check unique id before adding to list is required.
* (1 marks) Prompt the user to re-enter data if there are input errors,
* incorrect real numbers for the grades attribute.
* incorrect date format for dateOfBirthday attributes. The correct date format is: "dd-mm-yyyy".

c**. (1 mark)** Search candidates based on candidate name.

d. **(1 mark)** Delete a candidate based on candidate id.

e. **(1 mark)** Display all candidates sorted descending by grades.

**3. (1 mark)** Displays the menu of available functionalities for the user in question 2a, 2b, 2c, 2d, 2e.

A screen shot of a computer

Description automatically generated

***Hint****:* An abstract Menu class is provided for use in the development this application.

Notice: Students must strictly follow all requirements as described above for marking. Any modification of the requirement details is not allowed and will receive no marks!