

Project #3 (50 points)

Due Date

Friday, March 12, by 11:59pm.

Submission

- You must include both team members' names in the comment block on top of EVERY Java file.
- Your project folder must include the following **subfolders/files for grading**.
 - Source folder, including all Java files [35 points]
 - *.java from Project #2
 - 3 JavaFX GUI files – Main.java, *Controller*.java, *View*.fxml.
 - Test design document [10 points]
 - Javadoc [5 points]

Project Description

In this project, you will develop a GUI (graphical user interface) with JavaFX for the Payroll Processing System you created for Project 2. The GUI shall replace the PayrollProcessing class (console user interface) in Project 2. In other words, this project is a GUI version of Project 2. In addition, you must add the text file import and export functionality to the GUI. The file import will load (read) the employee database from a text file to the array (instance of Company class), and the file export will write the employee database from the array to a text file.

Requirements

1. This is a **group assignment**. You **MUST** work in pair in order to get the credit for this project.
2. You **MUST** follow the software development ground rules, or **you will lose points** for not having a good programming style.
3. Each Java class must go in a separate file. **-2 points** if you put more than one Java class into a file.
4. You **MUST** include all the classes from Project 2 and use them in this project, EXCEPT PayrollProcessing class and RunProject2 class. **-2 points** for each class not used.
5. Project 3 uses the Model-View-Controller (MVC) design pattern. You must use ONE JavaFX fxml file for the “view”, ONE Controller class for the “control”, and ALL the entity classes from Project 2 for the “model”. The Main.java, fxml and controller files constitute the GUI version of Payroll Processing System. **You will get 0 points** if you don't follow the MVC structure.
6. Your GUI must include the following JavaFX components.
 - (a) Use at least 3 different Layout Panes, such as BorderPane, GridPane, VBox, Hbox,, or **-5 points**.
 - (b) Use a TextArea to display output, or **-5 points**. All output **MUST** be appended to the TextArea. **You are NOT ALLOWED to use System.out in ALL CLASSES, or you will lose 10 points**. This means you must modify all the print() methods in the Company class to return a string, or define a toString() method for the Company class.
 - (c) Use a RadioButton group for single-selection items, or **-2 points**; for example, employee types, admin roles, or department, etc.
 - (d) You must disable not applicable items, **-2 points for each violation**; for example,
 - Only a part-time has hours worked and hourly pay rate.
 - Only a management has the role of either manager, department head, or director.
 - Only a full-time and a management has annual salary.
 - (e) You **MUST** set the title of the primaryStage (title for the window.) or **-2 points**.
 - (f) Use the **javafx.stage.FileChooser** class to select the file you wanted to import/export. A text file “database.txt” is provided for you to test your program. You must add an **exportDatabase()** method to the

Company class for the file export. Exporting the database does not remove the employees from the array.

you will lose 10 points if import/export functionality is not implemented.

7. You are required to **generate the Javadoc** after you properly commented your code. Your Javadoc must include the documentations for the constructors, private methods and public methods of all Java classes (*.java files.) You **must comment the Main.java and Controller class** and include them in the Javadoc. DO NOT include the *.fxml file, which is NOT a java file. Generate the Javadoc in a single folder and include it in your project folder to be submitted to Canvas. You are responsible to double check your Javadoc after you generated them. The grader will navigate the Javadoc with the “index.html”. You will **lose 5 points** for not including the Javadoc, OR, the grader cannot navigate your Javadoc through the “index.html”.

System Testing

1. You **MUST** create a test document and design the test cases for testing Project 3. Use the test cases in Project 2 sample input as a reference to design your test cases. For example, adding and removing employees, computing payments, set the hours worked by a part time, import and export database. In addition, you must design the test cases to test all possible invalid/exception conditions. For example, type mismatch or not entering data on GUI, etc. The test document is worth 10 points.
2. Use your test cases to manually test your GUI. All invalid data should be rejected by the GUI. Proper error messages must be displayed in the TextArea. **You will lose 2 points** for each invalid condition not rejected, or error message not properly displayed in the TextArea.
3. Your program must always run in a sane state and **should not crash in any circumstances**. The graders will try to produce exceptions while running your GUI. You must catch all Java Exceptions. Your program shall continue to run until the user stops the program execution or closes the window. **You will lose 2 points** for each exception not caught.