Seneca College

Applied Arts & Technology

SCHOOL OF COMPUTER STUDIES

Workshop 3

Due Date: March 2nd, 2025

INSTRUCTIONS

- *This workshop must be completed individually without any outside collaboration. All work must be your own. Copying or reproducing the work done by others (in part or in full) or letting others to copy or reproduce your own work is subject to significant grade reduction or getting no grade at all and/or being treated as academic dishonesty under the College's Academic Dishonesty Policy.*

- *Your goal is to finish the design part (using the Scene Builder is optional if you want to use it) during the lab time, unless otherwise specified by your instructor.*

- *The backend coding for your design can be submitted as a part of DIY.*

- *Your application must compile and run upon download to receive any mark.*

- *To submit the workshop, please follow the Submission Guideline provided at the end of this document.*

- *You must submit your workshop by the due date. Late submissions policy is specified in the Academic Procedures for Evaluations document available through the class plan on Blackboard.*

JavaFX Workshop: Auto Loan Application Development

Workshop Overview:

In this workshop, students will design and develop an Auto Loan Application using JavaFX, Scene Builder, properties, bindings, and event listeners. The goal is to build an application that manages customer information, vehicle details, loan calculations, and amortization schedules, following modular architecture.

Learning Objectives:

- Understand JavaFX layouts (HBox, GridPane, AnchorPane) and controls.
- Implement properties, bindings, and event listeners.
- Apply object-oriented programming principles (inheritance, interfaces, encapsulation).
- Work with multiple controllers for managing UI logic.
- Design and manage user authentication.

Application Features:

1. Login System:
    - Login window with hard-coded credentials (minimum two users).
    - Successful login redirects to the Auto Loan Application.
    - Unsuccessful login triggers an alert with focus reset to the username field.
    - No registration page is required.
2. Customer Information:
    - Fields: Name, Phone, City, Province (drop-down menu).
    - Real-time validation with focus on missing/invalid fields. (User Alerts to pop-up the messages)
3. Vehicle Information:
    - Type: Car, Truck, Family Van (radio buttons).
    - Age: New, Used (radio buttons).
    - Price of Vehicle (input field).
    - Real-time validation with focus on missing/invalid fields. (User Alerts to pop-up the messages)

4. Loan Information:
    - Down Payment (input field).
    - Interest Rate (0.99%, 1.99%, 2.99%, Other with custom input).
    - Loan Duration (slider: 12 to 96 months, tick marks at every 12 months).
    - Payment Frequency: Weekly, Bi-weekly, Monthly (radio buttons).
    - Estimated Fixed Rate Loan Payment (display formatted as currency).
5. Data Handling:
    - Clear Button: Resets all fields.
    - Calculate Button: Calculates loan payments using the LoanCalculation interface.
    - Save Rates Button: Saves current data in-memory (lost upon app closure).
    - Show Saved Rates Button: Displays saved rates in a ListView with Name, Vehicle Type, and Interest Rate. Selecting an item loads data back into the form (Use double click mouse event).
6. Amortization Schedule:
    - Displays payment breakdown (principal, interest, balance) for the loan duration.

Application Architecture (UML-Based):

See the attached UML document with the workshop on Black Board.

Workshop Structure:

Login System

- Design login UI (you can use SceneBuilder or not).
- Implement LoginController for validation.
- Add event handlers for error messages.

UI Design

- Create forms for customer, vehicle, and loan information.
- Utilize HBox, GridPane, and AnchorPane etc. for layout.
- Add radio buttons, sliders, and drop-down menus etc.

Business Logic

- Implement LoanCalculation interface and FixedRateLoan class.
- Calculate loan payments and display results.

- Apply data bindings and change listeners (e.g., slider updates).

Data Handling

- Implement save and load functionality for loan rates.
- Manage in-memory data using appropriate collections.

Amortization Schedule

- Generate and display amortization schedules.
- Implement LoanAmortizationController for detailed view.

Notes:

- Ensure that all required fields are validated with appropriate error messages.
- Saved rates are temporary and will be cleared upon application exit.

Design approval guidelines

Consider I am your client,

- **You must show me your designs (Front end) by Wednesday lab February 19th.**
- **You can also send me your Front end as screen shot on Teams.**
- **If you don't show me or send me your design's by February 19th then I will not accept your design after that and marks will be deducted**

Workshop Header

/***
Workshop #
Course:*<subject type> - Semester*
Last Name:*<student last name>*
First Name:*<student first name>*
ID:*<student ID>*
Section:*<section name>*
This assignment represents my own work in accordance with Seneca Academic Policy.
Signature
Date:*<submission date>*
***/

Code Submission Criteria:

Please note that you should have:

- Appropriate indentation.
- Proper file structure
- Follow java naming convention
- Do Not have any debug/ useless code and/ or files in the assignment

Deliverables and Important Notes:

All these deliverables are supposed to be uploaded on the blackboard once done.

- Design must be shown during the lab. (Read the Design Approval Guidelines above) 15%.
- Complete the code behind as the part of your DIY. 55%
- Submit a **reflect.txt** file with the submission. 10%

  Questions to be answered for the reflection:
  - What challenges did you encounter when implementing the loan calculation logic? How did you resolve them?
  - How did you ensure that the amortization schedule displayed accurate and formatted data? What techniques did you use to format the numbers?
  - What was the most challenging part of the workshop for you, and how did you overcome it? What new skills or concepts did you learn that you found particularly valuable?

        - If you were to extend this application, what additional features would you add? How would you approach designing and implementing these features?

- Video submission explaining core code pointers and showing the full working application (3 – 8 minutes max). 20%
- All submission goes to Black Board.
- Your submission should include
    - Video file with audio
    - Reflect.txt file
    - Complete zipped project.
- Late submissions would result in additional 10% penalties for each day or part of it.

Remember that you are encouraged to talk to each other, to the instructor, or to anyone else about any of the workshops, but the final solution may not be copied from any source.