



COMP-1502

# ASSIGNMENT #3

Dept. of Mathematics & Computing

Mount Royal University

Khosro Salmani



## ***The Problem***

In this assignment you will use JavaFX to create a graphical user interface for the toy store software you wrote in assignment 2. This means you'll primarily be replacing the view components of your assignment 2 project with graphical interfaces. Some corresponding changes will be needed in the controller components too (the severity of these changes will depend partly on your attention to modularity and object-oriented design during assignment 2)

## ***Requirements***

Your project must have the following features and specifications:

### **1. Project Directory Structure**

- README.md (containing project title, authors' names and contributions)
- bin/ (compiled java files)
- src/ (Java source files)
  - application/ (classes housing the "main" program)
  - controller/ (classes used to coordinate the "model"s and "view"s)
  - model/ (classes related to data objects, like toys)
  - view/ (classes related to user interface)
  - exceptions/ (custom exception classes)
- doc/ generated JavaDoc (Ensure the private option is checked and everything is included in the generated documentation.)
- lib/ any 3rd-party libraries. This folder may be empty
- res/ any resources or data files
- test/ unit tests

### **2. Exceptions:** The program should still throw the custom exceptions you implemented in Assignment

- The input price is negative when the user creates a new toy
- The minimum number of players is greater than the maximum, when adding a new board game

### **3. Testing:** We expect to see *\*further\** JUnit tests for your project when compared to the reasonable tests implemented for Assignment 2. Isolate some of the key functions and write tests for them that cover all possible situations.

### **4. Logging:** The project should create and update a log file while the user is running the application.

### **5. GUI Components:** The exact layout is up to you, but at a minimum your GUI must have the components shown in these sample screenshots.

## Sample Screenshots:

Home Add Toy Remove Toy

Welcome To Toy Store Company

Search Inventory :

☒ Serial Number ☐ Name ☐ Type

Serial Number(SN) Enter SN Here.

Name

Type

Search Clear

Buy

List View

Home Add Toy Remove Toy

Enter New Toy Information:

Category Figure

Serial Number

Name

Brand

Price

Available-count

Age-appropriate

Save

Figure

Classification

Animal

Material

Size

Puzzle

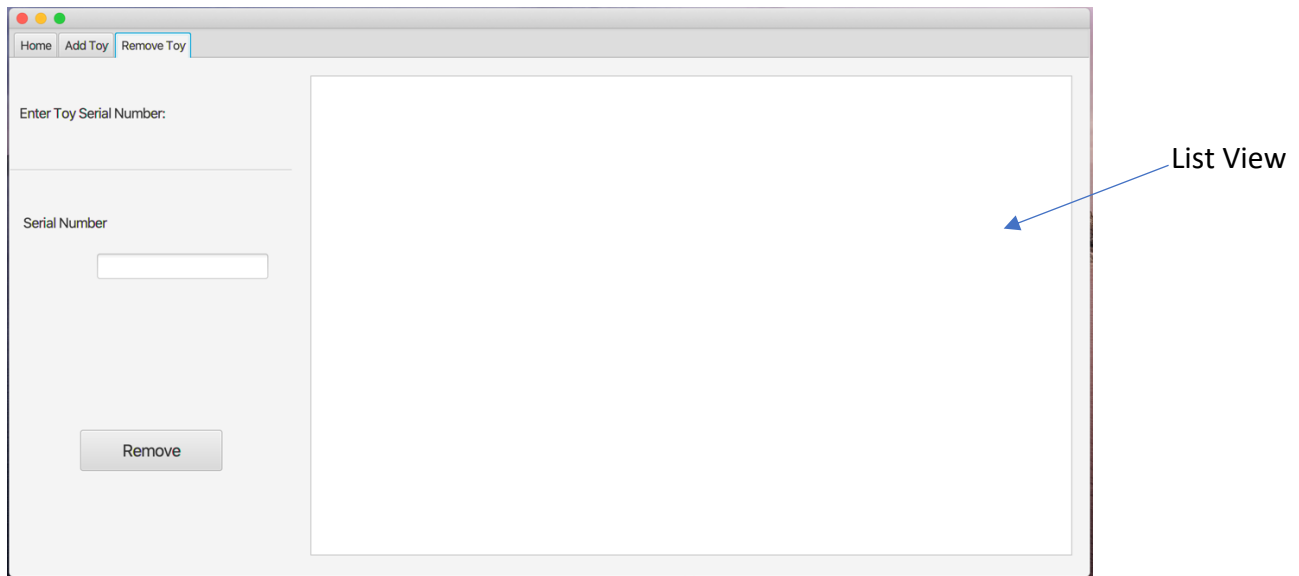
Type

Board Game

Minimum Number of Players

Maximum Number of Players

Designers



### ***Team Work***

- Before starting, understand the requirements and meet with your project partner to divide tasks equally. Describe each team member's contributions in your readme file.
- Both team members must contribute to the git project. The contribution of each member must be visible in the git commit history.
- Both members of a team must be knowledgeable about the whole project. This must be evident during the assignment demo. Your instructor may ask questions or make requests of either team member.

### ***Submission instructions***

- Push all git commits before the due date. The contribution of each team member must be visible in Git history. Paste the link to your GitHub repo into the D2L assignment to finalize your submission.
- You must demo your project to your instructor during a demonstration to be scheduled during tutorial/lab time. **This demo is mandatory. No demo = a grade of 0**

### ***Grading***

The following items are prerequisite to having your project graded. Failure to satisfy any of these points could result in a grade of zero:

- Code must compile and run. Include a jar file in your finished project.
- Basic project structure follows the folder structure outlined above
- Readme file in repository outlines each team member's contribution
- Demo with instructor

Projects that satisfy the above criteria will be graded according to the following rubric:

### ***Marking Rubric***

Item	Points
Program requirements met (Search/Add/Remove tabs work as expected, exceptions, inputs validated, etc)	40
Object-oriented structure (Problem broken down into classes/methods. Correct use of Events)	20
Documentation & comments (Javadoc, comments, appropriate/conventional names, etc)	20
Unit testing (tests work properly, good test coverage)	20

\* The maximum mark for this assignment is 100 and the bonus can boost your mark if you lost any marks in other items **only** in this assignment.