Homework 2 | Unit2: Data Management
Makers Toolkit - a CLI app

Assignment Goal:

1. **Select an API →** https://github.com/scalemailted/csci2120-apis

Select an API from Homework 1 developed by another student. I suggest you peruse all submissions, and I encourage you to choose the API that both inspires you and would be easy to use. Use the Javadoc documentation to learn how to use the API and the JUnit code to see demo examples of the code in actual use. Please include the original author's name in your project files as they will receive bonus credit for you using their API. Note: They will provide support to you as you need it with any method calls that break their contract via the API documentation

2. **Design an Makers Toolkit App**

Design & Structure an app that allows a user to make and store new custom instances of objects built using the selected API as a command line tool. Depending on the function of the API, the user may create item inventories, monster bestiaries, player options, spellbooks, shop systems, map makers, etc. The three core packages of your app should be: Model (app logic), View (output logic), Controller (input logic). Use classes to group similar methods, and packages to group similar classes. Use unified modeling language (UML) to scaffold your app before implementing.

3. **Implement Makers Toolkit App**

This toolkit must use multiple collections from the Collection framework to manage the user generated data. You must include at least one instance of each collection type: List, Set, and Map. Operations your toolkit should support includes: sorting, filtering, searching, accessing, adding, and removing from the makers system. You may use *Lecture Code* as a reference.

4. **Test Makers Toolkit App**

Test your java app and ensure that it executes and operates based on our directions. Create a readme.md file that defines the operations of your toolkit app. This toolkit will be used as the basis for future homework projects but is not intended to be a game in and of itself.

Learning Objective:

Use external code designed and developed by someone else to build a command-line java application that allows users to make game objects such as items, or spells, or monsters, or skills, or whatever the selected API allows. These user made objects should then be added and managed using the Collections framework. You must use at least one of each Collection type: a List, a Set, and a Map.

- Build an App using another student's API
- Use a List from Collections Framework
- Use a Set from Collections Framework
- Use a Map from Collections Framework
- Separate your java code into Packages: Model, View, Controller

Implementation Advice:

Focus on building an application that builds, searches, and sorts items from the selected API. Start with a simple tool and add complexity to it, using a top-down iterative strategy.

Test Driven Development (TDD) & JUnit Testing

Design a Test Plan for your maker toolkit

You should create a Test Fixture for each class and a Test Case for each class. Make sure your Test plan includes checking routine, challenging, and edge-case instances.

Readme.md & JavaDocs

Full API documentation should be included with your maker toolkit

Your readme.md should include directions for users of your apps. Your javadocs should provide the API of your app for developers to extend its functionality in future revisions.

Resources:

You may use **Lecture code or Lab code** to start!

https://github.com/scalemailted/csci2120-apis ***(APIs from Homework 1)***

Showcases & Demos (JavaScript example):

https://codepen.io/jamescourson/pen/MdrdJe

https://codepen.io/sanchopanza/pen/Bbdyoz

Why Write Your Own App?

1. **Custom Developer toolkits:** Designing CLI developer tools is a valuable mini-project that represents a completed application. Developer tools are used to build out content that can be used to create game content

2. **Portfolio project:** Employers and recruiters commonly prefer candidates who have project portfolios to demonstrate their technical capabilities. Passion and creative projects are often good additions.

Item System Inspirations

- Maker kit for Items, Weapons, etc.
- Maker kit of Food, Cocktails, etc.
- Maker kit for Bestiary of Monsters
- Maker kit for creating Heros with classes and/or skills
- Maker kit of Magic Spells
- Maker kit for creating Quest logs
- Maker kit for generating Combats
- *Hint*: *A crafting or mini-RPG may be a future assignment that extends this one.*

Grading Rubric

```
Part 1: [ DESIGN  ]  select API, UML diagrams using top-down OOP  [  20%  ]
Part 2: [ BUILD   ]  Collections Framework + API                  [  60%  ]
Part 3: [ DELIVER ]  javadocs, junit, executable, readme          [  20%  ]
Part 4: [ BONUS   ]  Outstanding Submission;                      [ 0-10% ]
```

Submission:

Tools

Java, IntelliJ, JavaDocs, JUnit 5

Submission:

Submit your source code to gitlab or github. Publish your HTML file into a doc folder of your project. To receive a grade, you must schedule a zoom audit with me where you walk-through your codebase.