**Java Application Assignment**

You are to create a complete Java application. This application must be approved by your instructor; however, you will decide what this application is about.

# **This application must include at a minimum:**

* Three classes minimum
* At least one class must use inheritance
* At least one class must be abstract
* JavaFX front end – as you will see, JavaFX will allow you to create a GUI user interface.
* The User Interface must respond to events.
* If your application requires a data backend, you can choose to use a database or to use text files.
* Error handling - The application should be able to handle invalid data.

# **There will be multiple assignments associated with this application.**

# **Those assignments include:**

* ***Module 02* - *Semester Project: Proposal*** – This assignment will require that you provide your instructor the proposal for your project. Make sure you read any feedback the instructor provides you in the comments section. You may get full credit but that does not mean the project has been accepted as-is. You may get feedback in the comments section of the graded assignment that may include changes you must make in order for the project to be accepted or you may not get any credit until your proposal is acceptable. Again, read any comments provided as part of this graded assignment. Once your proposal is accepted, you can start working on your Java application. **This assignment is worth *50 points* and is due at the end of module 02.**
* ***Module 05* – Semester Project: UML** – At this point in the course, you have learned about objects, java classes, object-oriented thinking, UML, inheritance, polymorphism, abstract classes, interfaces and unit testing. In this status update, you will provide documentation that documents your classes using UML.

Note: Since the user interface will use JavaFX, these classes should be unrelated to the user interface of your application. These classes should be related to functionality of your project. For example, let's say your application will require the user to enter information about a person (first name, last name, etc.). You would want to create a class that contains all the fields as well as the setters and getters required for the object (with the setters doing validation as required for the associated field). This class might also contain information about which fields are required for the object to be valid as well as an override of the toString method for class to output all the fields using the toString method in some predefined format. The JavaFX user interface could then use the object when trying to validate the user or to output the object to a report, etc. What these classes should not contain is anything related to the user interface for the application. That will come in the JavaFX portion of the project.

**This assignment is worth 50points and is due at the end of module 05**.

* ***Module 06* - *Semester Project:* Classes / Test Classes** – At a minimum, you should have created the classes your application will require. Additionally, you should have created “test” classes that will instantiate your application classes and verify that these classes work. These “test” classes should output to the console test messages that verify your classes perform correctly. **This assignment is worth 50 points and is due at the end of module 06.**

Note: After this assignment is completed, you may find you need some additional classes to complete your project. You may add classes as required to make the app fully functional. However, make sure you get approval for any changes you need to make and remember to update your UML document as you will need to submit this document along with the final project.

* **Module 07 - JavaFX -** In the previous assignment, your non-GUI related code should have been complete.

For this part of the semester project, you should complete the JavaFX GUI widgets (controls / windows, etc.). You should be working on finalizing the events. You should have a minimum of two events completed as well as your complete GUI (i.e. Windows and fields).

**This assignment is worth 50 points and is due at the end of module 14.**

* ***Module 08 – Semester Project: Finalize Project*** - Your application should be complete at this time. All requirements as outlined above and approved for this project should have been met. **This assignment is worth** **180 points and is due at the end of module 08**.

The following is an example app that provides an ordering system for a fictitious restaurant.

This app will allow the user to track orders for individuals who purchase meals from the restaurant. The purchaser will be given an ID which will allow the employee to look up the order which will contain 1 or more food items the purchaser has ordered. When the purchaser saves their order, they will be given an order ID and a total price of the order. When the purchaser initially enters their order, the system will need to validate the food against a list of food choices

The employee can enter the ID provided by the purchaser and then the employee can print out a list of foodstuffs that the purchaser has ordered as well as the total price for the order. The food choices and any other required data will be stored in text files using a delimiter to separate each field within the text file. This delimiter will be used in all text files required for this app.

The text files include at a minimum:

* **Order** – this file will contain the following fields
  + order ID
  + purchaser’s first
  + purchaser’s last name.
* **Order Detail** - This will contain the items that the person has ordered. Each entry in this file will contain the order ID from the above record as well as the Food List ID to output details of the food choices.
* **Valid food choices**
  + ID
  + Food Item
  + Price