**ITSC 1213 – Assignment 3 – Project & Class Design**

The main idea in this assignment is for you to design your own set of classes that work together to solve a problem. You’re getting to the point where you should be able to take more responsibility for the solution and depend less on us for detailed instructions.

**Prep Work**

Choose a topic from the list of topic descriptions provided at the end of this document.

**Proposal (5pts) – Due Apr 12th at 11:55pm. (No late extensions on this)**

Complete Part 1 (below) and submit a draft of your UML diagram. This should be as close to complete as possible. The more complete this is, the more useful feedback we will be able to give you as you move forward with the rest of the assignment. You will have the opportunity to improve your UML class diagram and submit it again as part of Assignment 3.

**Assignment 3 (95%) – Due at Apr 19th at11:55pm. (Automatic extension to Apr 22th at 11:55 pm with a 10% penalty – NOT ACCEPTED AFTER THE 22nd)**

**Part 1 – For the topic that you have chosen, create a UML diagram that shows the different classes, fields, and methods and how they work together (20%)**

1. You must submit a formal diagram. *No hand-written submissions will be accepted.* You may use lucidchart, draw.io, Visio, or any diagramming software that you choose.
2. Your diagram must show:
   1. All the classes, fields, and methods. Remember that you will need methods that accomplish appropriate functionality, not just getters and setters.
   2. The classes must be connected in an appropriate manner (with ‘uses’ lines and ‘extends’ arrows)
   3. You must include visibility (i.e., public or private), data types, return types, and any other pertinent information.
   4. Show the data structures (i.e. arrays or ArrayLists) your program needs to hold the important data.

**Part 2 – Create the project, and the classes (along with fields and methods in each) (15%)**

1. In NetBeans, create the project that corresponds to your UML diagram.
2. Create all of the classes included in your UML diagram.
3. Add all fields for each class. Note that every field referred to in your UML diagram must be present and must have characteristics matching those described in the UML diagram.
4. Add a stub method for every method in your UML diagram.
5. Implement all of the getters and setters.

**Part 3 –Implementation (30%)**

1. Choose at least five major functions that your solution needs and implement them with working code and include appropriate comments. (Getters and setters are not major functions.)
2. If there are additional major functions that these five DO NOT rely on, you can leave the method a stub, but you must add a one or two sentences description to explain what the method is supposed to do. Use complete sentences.

**Part 4 – Create a test harness program that will prove the project works (20%)**

1. Create a test harness program that tests the working parts of your program. This should simulate the functionality of your program with test data. This program should accept user data via the console and use that data to do something with your program. (Look at the burger ordering system lab as an example of what we mean by this. Like the burger ordering system, your program should create some amount of initial data we can work with and add to)
2. Add a comment to the top of your test harness program that states which are the five major functions and what they do. This will allow us to focus on them when we do the grading.

**Coding Style – (15%)**

This grade is awarded for proper coding style. This includes:

* Your name in comments at the top of every Java file
* Appropriate method, variable, field, object and class names
* Proper indentation
* Good JavaDoc and within method commenting (explains what code is doing)
* Well-organized, elegant solutions for each part

**Files to submit:**

* A pdf of your UML diagram. Make sure that it is readable otherwise it will not be graded.
* A ZIP file of your NetBeans project.

**Topic Descriptions**

## Community Theater Audition Management System

Design a system for a community theater that helps the various directors manage the audition process. This system must include user accounts for directors and actors. It also must include a scheduling component for scheduling auditions. An audition is related to a specific show and a specific part. There needs to be a way to add a new show and cast members. There needs to be a way for a director to give ratings to each actor who auditions for a part. Inheritance: 1. User account is sub-classed into directors and actors.

## Community Food Bank Inventory System

Design a system for a local food bank to keep track of food inventories. The system should have food items that are both perishable and non-perishable, as they accept items from community gardens. The system should have users who are either employees, volunteers or donors. The system should allow food items to be added and removed as they are given out. The system also needs to alert if perishable food is beyond it’s safe date for consumption. Inheritance: 1. User account is sub-classed into employees, volunteers and donors. 2. Food item is sub-classed into perishable and non-perishable.

## Bookstore Management System

Design a system for a book store that allows the owner to keep track of the store’s inventory and members. The store sells two types of products: books and CDs. The store offers two types memberships to customers: regular memberships and premium memberships. The regular membership is free, while the premium members pay a fee every month. For this reason, the store keeps track of payment method and whether the fee is paid on time for the premium members. The system should keep track of the members and how much money each has spent at the store. The system also keeps track of the inventory of each product. Inheritance: 1. Member is extended by premium members. 2. Product is sub-classes into books and CDs.

## Laboratory Management System

Design a system for a research laboratory which keeps track of the experiments the laboratory is conducting, what reagents (materials, chemicals, etc) are used in the experiments, and the researchers in charge of the experiments. The system should have hazardous and non-hazardous reagents, and keep track of inventory of each. When a new experiment is created it should deduct whatever materials it needs from the inventory, and a warning about the hazardous materials it uses should be given. When an experiment completes, the amount of hazardous waste it used should be collected, and after this has reached a threshold, all hazardous materials should be disposed of. The laboratory should have two types of employees, scientists and interns, and at least one of each should be the ones conducting an experiment. Inheritance: 1. Employee is sub-classed into scientist and intern. 2. Reagent is extended by hazardous reagants.