# CAB201 Programming Principles

## Semester 2, 2021

## Object-Orientated Design and Implementation

## Due Date: 24/10/2021

## Weighting: 50%

## Assessment Type: Individual

### Task

This assignment aims to give you real problem-solving experience, like what you might encounter in the workplace. Your task is to apply object-oriented design and programming techniques to create a well-documented and easy to maintain software solution that achieves the following user stories.

### User Stories

1. As someone interested in buying or selling products I want to register as client with an online auction house so that they have my contact details, including address, and can choose a password that will keep my client records private and secure.
2. As a client I want to authenticate myself with the system in order to access my private records and conduct transactions.
3. As a client I want to be able to log out from the system so that the next person to use the system cannot access my private records or conduct transactions using my account.
4. As a client I want to be able to advertise my product. The advertisement will include the name of the product, the type of product and the initial cost.
5. As a client I want to be able to list all my products that I currently have advertised for sale.
6. As a client I want to be able to search for products for sale using a particular product type.
7. As a client I want to be able to bid a dollar amount for one of the products.
8. As a client I want to be able to view a list of the current bids for my products.
9. As a client I want to be able to sell one of my products to the current highest bidder.
10. As a client I want to be able to say that I would purchase it through “click and collect” or having the product “home delivered”.
11. As a client I would like to know how much the auction house charges me for selling a product. The charge is $10 for a “click and collect” and $20 for having the product “home delivered”.
12. As a client I would like to be told how much sale tax is payable when I purchase a product. The tax is equal to 15 percent of the product plus an additional $5 if it is “home delivered”.

A video on Blackboard gives an example of how the application might achieve each of the above user stories.

## User Interface

The company has plans to eventually develop a graphical user interface for this application, however, this prototype implementation will have a simple text-based interface. To make it easier to switch to a graphical user interface in the future we want to ensure the user interface code is kept separate from the application logic. To help with this an **UserInterface** class has been provided. You may use this class if you wish.

## Database

The company has plans to eventually use a relational database to store and save all customer data. However, for this prototype, all data will just be kept in memory (and will therefore be lost when the application is shutdown).

## Recommended Approach

You are encouraged to design and develop the application in an iterative and incremental fashion – one user story at a time. You can implement the user stories in whatever order you please, however for each new user story you should design and implement only the classes and methods that you require to implement that user story. You are encouraged to make changes to your class diagram using draw.io (https://app.diagrams.net/). You may need to first *refactor* some of the classes and methods that you have already implemented for earlier user stories in order to better facilitate the new user story. After each user story you should have working code that achieves all of the user stories considered so far.

To further encourage this style of iterative and incremental development, we require two separate product increments to be submitted. The first supporting just the first user story that you choose to implement and the second being your final product including all user stories. For each product increment you will need to document your object-oriented design via a class diagram and provide well documented source code packaged as a Visual Studio 2019 solution. Your tutor will provide constructive feedback based on your first submission on Week 11 or Week 12.

## What to Submit

For this assignment, you are required to submit a zip file – via Blackboard – containing the following items:

- The project folder that contains all of your project files.
- Phase 1: Present an image (e.g. BMP, TIFF, JPG) that presents the class diagram of your first user story.
- Phase 2: Present an image (e.g. BMP, TIFF, JPG) that presents the class diagram of all of the user stories.
- A checklist of all the user stories and whether you have competed them or not.

We recommend that your class diagram should be made in draw.io (https://app.diagrams.net/).

Place all your files into a folder named *"CAB201_Assignment_2021_S2_n12345678"* replacing the *n12345678* with your student ID and zip the folder. You should then upload this zip file to Blackboard.

## How to Submit

1. If you are in Brisbane, you must test your program in one of the CAB201 labs prior to submission. Markers will attempt to compile and run your code in that environment only. It is your responsibility to ensure that your code compiles and runs on PCs in the QUT computer labs of CAB201. You can access QUT computers remotely, see the VMWare Horizon Client Tutorial on Blackboard for a guide.

**2.** Follow the submission link on Blackboard to submit your zip file.

**3.** Make incremental submissions. You may make as many submissions as you want, and we will only mark the latest one that came in before the deadline. Make incremental submissions as you complete more functionality. Incremental submissions will protect you against submission delays and save you from issues such as accidentally breaking something in your submission ten minutes before the deadline or system delays due to processing too many assignment submissions.

## Academic Integrity

Please read and follow the guidelines in QUT's Academic Integrity Kit, which is available from the Blackboard site on the Assessment page. Programs submitted for this assignment will be analysed by the MoSS (Measure of Software Similarity) plagiarism detection system (http://theory.stanford.edu/~aiken/moss/).

## Final Comment

Though all care has been taken in the production of this specification and related documentation, there may be a need to notify by email any alterations/clarifications to this specification and related documentation. **So, check your QUT email daily!**