CSC 207 Introduction to Software Design
Winter 2015 – Project Phase I

Logistics

- **Due date:** 9:00pm Monday 23 February 2015
- **Group size:** Pair. Submitting this alone is not an option; if you do, you will receive a mark of 0.

Overview and Learning Goals

In Phase I of the project, you will review a client request for a software program and create an object-oriented design of a solution. By the end of this phase, you should have:

- practised performing analysis of software specifications,
- practised designing an object-oriented software system,
- used CRC cards as an aid to the object-oriented software design process,
- used a shared version control repository to support collaborative software development,
- experienced some of the joys (and frustrations) of working with others and consequently developed some strategies to facilitate productive team interaction.

The Client Request

A company has contacted you about building a program for them. Here is the description.

> We need an Android application for booking airline tickets. The application should provide the functionality to store, retrieve, and update flight information, as well as information about our clients. A user should be able to create a trip itinerary by providing origin, destination, and travel dates information. The application should calculate and display itinerary costs and create new bookings. A user should be able to display the list of suggested itineraries by the itinerary cost and by total travel time.

Feature List

After consulting with your requirements engineering experts and after further communication with the client, you arrived at the following feature list:

- Clients can launch the flight booking application, which loads saved data, if it exists.
- Clients can search available itineraries by entering a departure date, and travel origin and destination. An itinerary should include, per flight: (1) flight number, (2) departure date and time, (3) arrival date and time, (4) airline, (5) origin, and (6) destination, plus an overall itinerary cost and travel time.
- Clients can display search results sorted by total travel time or by total cost.
- Clients can select an itinerary from the displayed list for booking.
- Clients can book an itinerary.
- Clients can view booked itineraries.
- Clients can enter personal and billing information to be stored in the system.
- Clients can edit personal and billing information stored in the system.
- Administrators can view and edit all stored client information.
- Administrators can enter flights information into the system.
- Administrators can upload flights information into the system in a file.

What to Submit

0.1 Task 1 — CRC Model

With your partner, create a CRC Model for the problem described. Instead of handing in your index cards, create a `pdf` file named `crc.pdf` in your **team's** subversion repository in folder `PI`. This file should contain a collection of CRC cards, and nothing else. Please, make sure your file is easy to read, the figures are not too small, not too big, etc. You may want to look at a sample `pdf` file linked from the Assignments and Project page of the course website.

Keep in mind that you are designing the **backend**, not GUI screens. None of the CRC cards should mention buttons, text fields, or other graphical components. In later phases of the project, you will implement both the backend and a GUI that will invoke the backend methods, but for now you are just designing the backend.

0.2 Task 2 — Walkthrough

In the file `walkthrough.txt` submit a walkthrough for your CRC model for the scenario "search all itineraries from Toronto to Venice on May 3, 2015". You may want to look at a sample walkthrough file linked from the Assignments and Project page of the course website.

0.3 Task 3 — Data structures and Files used

Describe all data structures used by your classes. In addition, describe the format of the file(s) in which the application stores its data.

You may present your description in paragraphs or bullet points, as long as you use complete sentences. Imagine that the person reading the description is a colleague or manager who has technical knowledge but is not intimately familiar with your project. Your description should be no more than one page in length, using single spacing.

(Of course, we realize that in a real application you would use a database to store the information, would connect to a server, etc. To keep the project within the scope of the course, we will store all information locally, in files.)

Submit your descriptions in a plain text file named `data.txt`.

Meeting with your TA

Between February 23rd-26th, you will have a 15 minute long meeting with your TA. During this meeting, you will need to (a) present the submitted walkthrough, (b) present two walkthroughs of the TA's choice, (c) explain the choice of data structures and files used by the program.

Your TA will contact you to schedule the meeting.

Marking

All of these items affect your grade:

- The modularity of the design, and the degree to which it is reusable and extensible.
- The degree to which the design meets the requirements.
- The use of OO concepts in your design.
- The correctness of your walkthrough.
- The clarity of the presentation of your walkthrough.
- The appropriate use of files and data structures.
- The quality of the written description of data structures and files used by the system.
- Your performance during the meeting with the TA.

Checklist

Have you…

- used your team repository and not your individual repository to submit your work?
- submitted `crc.pdf`?
- submitted `walkthrough.txt`?
- submitted `data.txt`?
- verified that your changes were committed using `svn list` and `svn status`?