



Object Oriented Programming

Topic 9: Design Patterns

Resources

The following resources can help you with this topic:

- *Design Patterns - Elements of Reusable Object-oriented Software* by Gamma et al.

Topic Tasks

Before starting to work on the tasks, first read through this entire document to get a sense of the direction in which you are heading. Complete the following tasks and submit your work to the Blackboard for feedback before the deadline (check Blackboard for the submission deadline).

Supplementary Exercise - Conceptual Modelling and C++

Supplementary Exercise - Case Study: Iteration 5

Remember to submit your progress, even if you haven't finished everything.

After you have **discussed** your work with your tutor and **corrected any issues**, it will be signed off as complete.

Supplementary Exercise: Conceptual Modelling and C++

Following are some exercises on conceptual modelling and C++ to prepare you for Pass Task 13 assessment.

For diagrams and programs, clarity is important; Avoid diagrams or programs which are sloppy and hard to read. Correct syntax needed for C++ programming.

Question 1

Consider the following scenario:

The Rhythm one-stop book centre has books, videos, and CDs that it loans to its customers. All loan material has a unique id, a title and a category. In addition, books have one or more authors, videos have one or more actors, while CDs have one or more artists. The book centre maintains one or more copies of each loan item (book, video or CD).

Copies of all loan material can be loaned to customers. The loan material is categorized into two categories: Reference-only material and Normal material. Reference-only material is loaned for 3 hours only and cannot be removed from the book centre. Normal material can be loaned for a weeks. For every loan, the book centre records the customer, the loan date and time, and the return date and time. For customers, the book centre maintains their name, address and phone number.

Draw a class diagram (or two, if this is more convenient) for the description above. Make sure to show attributes, multiplicities and aggregations/compositions, where appropriate. No need to show any operations.

Question 2

Model with a class diagram for the Snacks and Drinks Vending Machine system.

The vending machine sells small, packaged, ready to eat items (snacks, cookies, crackers, etc.) and canned drinks (soft drinks, herbal teas, energy drinks, etc.). Each item has a price and a name. A customer can buy an item, using a touch-and-go card (issued by the vending machine company) to pay for it. No other payment forms are allowed. The touch-and-go card records on it the amount of money available.

The functions supported by the system are:

- Sell an item (choose from a list of items, pay item, dispense item)
- Recharge the machine

- Set up the machine (define items sold and price of items)
- Monitor the machine (number of items sold, number of items sold per type, total revenue)

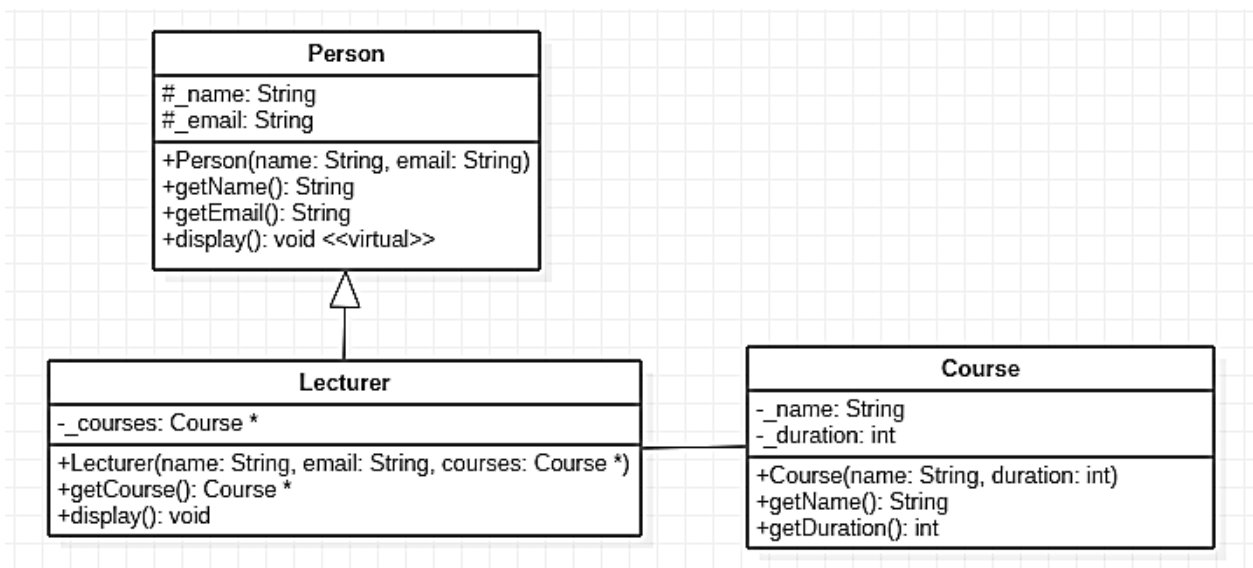
The system can be used by a customer, a maintenance employee (who recharges items in the machines), and an administrator (who sets up the machine).

Question 3

Develop a sequence diagram describing successful transaction procedure for the model of the Snacks and Drinks Vending Machine as given in Question 2.

Question 4

Based on the class diagram given below: -



- Write C++ codes for all the classes given.
- Create an instance of Course class, named it CS and initialize it with its appropriate field values.
- Create an instance of Lecturer class, named it lect1 and initialize it with its appropriate field values. (Note: Use the Course instance created in part (b) as deemed appropriate)
- Execute the `display()` method of the Lecturer instance created in part (c).

Supplementary Exercise - Case Study: Iteration 5

Note: Do not let Credit Tasks delay you in keeping up with unit due dates. If you are behind, skip these tasks and move on to the next Pass Task. You can always come back and have a go at this later if you get time.

Over the next few weeks you will implement a larger object oriented program that demonstrates the use of all of the concepts covered so far. This will help you develop a deeper understanding, and create additional pieces of work to communicate this understanding.

1. For this week complete Iteration 5 - Create a Console Application

Once your programs are working correctly add additional details to your cover page. Remember to relate what you are doing to the unit's learning outcomes.

Tip: Now would be a good point to reflect on anything you did in these iterations that was interesting or challenging. Elaborate on this in the cover page so that you can take advantage of this in your portfolio.