Tutorial/Practical Class 6 in Week 7

In this tutorial, you will answer some questions regarding Class Diagram and Design Principles

Relevant Course Materials: Week 6: Lecture and Lectorial (Class Diagram and Design Principles)

Relevant Assignments: Assignment 2

[Individual] Part A: Class Diagram Concepts (25 mins)

1. Answer the following questions
    a. What are objects?
    b. What is the difference between a class and an instance?
    c. What is encapsulation?
    d. What is the difference between Aggregation and Composition?
        i. Provide an example of Aggregation and Composition in the Library System
    e. What is the difference between cohesion and coupling?

[Team-based] Part B: Class Diagram of the Online Bookstore Platform (80 mins)

Note: You need to team up with your group members and work on the following tasks.

Imagine you (students) are a software engineer working in a software development organization and want to develop an Online Bookstore Platform. The Online Bookstore Platform stores and sells different types of books. In the following parts, you are expected to work on the design process of the Online Bookstore Platform. You have freedom and have to add more features, services, requirements, etc. to the Online Bookstore Platform.

1. Use the information that you gathered from previous tutorials about the stakeholders, functional/non-functional requirements, and use case modelling of the Online Bookstore Platform and create a Class Diagram for this system using SOLID and GRASP principles (**45 minutes**).

    Note 1: Your class diagram should have at least **10 classes**: all classes should have two attributes and two operations, multiplicity. Your class diagram should include 4 diverse relationships such as association, generalization, aggregation, and composition relationships

    Note 2: Use the Visual Paradigm[1,2] software tool to create the class diagram

2. Write the skeleton code of the designed class diagram in Java (**35 minutes**).

    Note: Please respect the following conventions when implementing the code:

    - Create a basic Java project in Eclipse.
    - Methods should be empty or return a dummy value (empty String “”, 0, or 0.0 depending

[1] https://ap.visual-paradigm.com/rmit-university/
[2] https://www.rmit.edu.au/students/support-services/it-support-systems/software-apps

on its return value).

    - No need to provide a concrete implementation of methods
- Implement the visibility as presented in the class diagram.
- If required by multiplicity, use java.util.List (e.g. “private List<User> books;”).