COMP 2120: Object-Oriented Programming using Java

Lab#3 Tasks

Due: [Feb. 2, 2026] (During labs)

Lab Problem: Working with Multiple Classes

In real-world systems, objects are often composed of other objects. For example, a **Book** *has an* **Author**. Keeping them in separate classes makes the program clearer, reusable, and closer to real-world modelling.

Problem Statement:

1) Create a class **Author** that stores the author's name and email. (5)
   a. It should have a constructor to initialize these values.
   b. It should have a method displayAuthor() to print the author's details.
2) Create a class **Book** that stores the title, price, and an Author object. (5)
   a. It should have a constructor to initialize these values.
   b. It should have a method displayBook() : prints book details along with author details.
   c. It should have a method cheaperThan(Book other) : compares the price of two books.
   d. It should have a method hasSameAuthor(Book other) : checks if two books are written by the same author (both name and email match).
   e. It should have a method withinBudget(double budget) : checks if the book's price is less than or equal to the given budget.
3) In the **Test** class (with the main method): (5)
   a. Input using keyboard details for two books (each with its author).
   b. Display the details of both books.
   c. Compare their prices using cheaperThan().
   d. Check if they have the same author using hasSameAuthor().
   e. Check if each book is within the budget value entered by the user using withinBudget().

Expected Input/Output with expected function call (Inputs must be taken using keyboard):

```
// The following info should be used to create book1 and book2 objects.
// Remember: each Book requires an Author object


Enter details for Book 1 (and author objects as needed):
Author Name: Paulo Coelho
Author Email: paulo@email.com
Book Title: The Alchemist
Book Price: 150

Enter details for Book 2
Author Name: George Orwell
Author Email: orwell@email.com
```

```
Book Title: Animal Farm
Book Price: 85.5

// Calling function book1.displayBook()
Book 1 Details:
Title: The Alchemist
Price: 150.0
Author Name: Paulo Coelho
Author Email: paulo@email.com

// Calling function book2.displayBook()
Book 2 Details:
Title: Animal Farm
Price: 85.0
Author Name: George Orwell
Author Email: orwell@email.com


// Calling function book1.cheaperThan(book2)
Animal Firm is cheaper than The Alchemist.

// Calling function book1.hasSameAuthor(book2)
The two books do not have the same author.

// Calling function book1.withinBudget(100)
The book 'The Alchemist' is not within the budget of 100.

// Calling function book2.withinBudget(90)
The book 'Animal Farm' is within the budget of 90.
```

Evaluation [Total points: 20 (2% of the final grade)]:

The lab evaluation is divided into two parts:

- **Part 1: Attendance and Participation (5 points):**
  - You must attend the **initial discussion session** at the beginning of your lab section.
  - During this time, the GA/TA will walk you through some example problems and explain the concepts and solutions.
  - Full marks (5 points) will be awarded for **being present and actively participating** in this discussion.

- **Part 2: Code Presentation and Explanation (15 points):**
  - You are required to **present and explain your solutions** to **the given problem** to the TA during the lab.
  - The TA may ask any questions related to the solution/code, including code modification, input/output testing, logic validation techniques and conceptual questions (e.g., why do we need two classes in this problem setup?)
  - Partial marks can be given if some parts are correct.

- While students can submit their solutions on Brightspace, the evaluation will be conducted in person during the lab by the TA.

NOTE: Your grades must be uploaded within a week in Brightspace. If you don't see it, please contact the TA who graded the work. After that, the lab instructor /TA will not be responsible for any missing grades.