Practical Test, Autumn Semester 2023
Royal University of Bhutan
Gyalpozhing College of Information Technology
CSA201: Applied Data Structures and Algorithms
Year II & III, Semester I

Time: 1.5 Hours **Max. Marks: 25**

*Before you begin working on the practical test, rename your Java program file with your unique student number as the filename. For example, if your student number is "123456789," your file should be named "**123456789.java.**" All questions in this practical test are compulsory, and marks are given at the end of each question.*

Task: Create a Java application that manages assets (or College resources) for the GCIT. Follow the detailed requirements below to complete this task:

Requirements:

1. Create a hierarchy for different types of college items. You should include a base class named **Item**. Under this hierarchy, implement at least two subclasses (e.g., Equipment and Books). Each subclass should have specific attributes and methods. **[7 M]**
   - Implement the base class **Item** with the following attributes:
     - itemID: An integer to store a unique identifier for the item.
     - itemName: A string to store the name of the item.
     - description: A string to provide a brief description of the item.
     - quantity: An integer to store the current quantity of the item in stock.
     - price: A double to store the price of a single unit of the item.
   - Implement a constructor to initialize these attributes and getter methods to retrieve the attribute values.
   - Create subclasses with unique attributes and methods for different item types, such as:
     - **Equipment**: Include attributes like manufacturer, condition, and methods specific to equipment.
     - **Books**: Include attributes like author, edition, and methods specific to books.

2. Develop a class named "**ItemManager**" responsible for handling the collection of college items (catalog). This class should include the following methods: **[10 M]**
   - addItem(Item item): Add an item to the college catalog/items.
   - removeItem(int itemID): Remove an item from the catalog based on its unique ID.
   - updateItem(int itemID, int quantity): Update the quantity of an item.
   - calculateTotalValue(): Calculate and return the total value of all items in the college catalog.

3. Design a user-friendly text-based interface tailored for college staff to interact with the application. The interface should include a menu with options to: **[5 M]**
    - Add items to the college catalog.
    - Remove items from the catalog.
    - Update item quantities.
    - Calculate and display the total value.
    - Quit the program.

4. Create a **Main** class to serve as the entry point for the program. Implement a loop that continuously displays the menu and executes the selected option until the user chooses to quit the program. Ensure that the program correctly calls the methods of the **ItemManager** class to perform college assets management tasks based on user input. **[3 M]**

Note: Submit your Java program as a single .java file, which includes the **ItemManager**, **Item**, and **Main** classes.

******END*****