

LAB 211 Assignment

Type:	Long Assignment
Code:	J1.L.P0026
LOC:	500
Slot(s):	N/A

Title:

Courses Program Management

Background:

FU's short-term informatics training center is training informatics courses. Short-term topics such as Photoshop, Java, Web, ...The management of courses, students, transcripts, and revenue is done through Excel. Currently, the center is facing difficulties when the number of learners is increasing day by day. The data is getting bigger and bigger, so it is very difficult to manage Excel without security. The center wishes to build software to solve the above difficulties.

Requirements for professional functions:

- Manage Topics
- Manage courses created from topic

Program Specifications:

Build a courses management program. With the following **basic functions**:

1. Manage the Topics

- 1.1. Add Topics to catalog
- 1.2. Update Topic
- 1.3. Delete Topic
- 1.4. Display all Topics

2. Manage the Course

- 2.1. Add Course
- 2.2. Update Course
- 2.3. Delete Course
- 2.4. Display Course information
- 2.5. Display all Course

3. Manage the Learner

- 3.1. Add Learner to Course
- 3.2. Enter scores for learners
- 3.3. Display Learner information

4. Search information

- 4.1. Search Topic

- 4.2. Search Course
 - 4.2.1. By Topic
 - 4.2.2. By Name
- 5. **Save** Topics, Courses and Learner to file.
- 6. Others- **Quit**

Features:

This system contains the following functions:

✚ Function 0 – 50 LOC: Build your data structure

- **Topic** information includes Topic *code or ID, name, type (long/short term), title, duration, etc....* **Topic ID is unique.**
- **Course** information includes Course *code or ID, name, type (online/offline), title, begin date, end date, tuition fee and topic, etc....* **Course ID is unique.**
- **Learner** information includes *code or ID, name, date of birth, score, course, etc...*
- **Applying OOP principles, design with inheritance, polymorphism, interface or abstract class.**
- For simplicity, a learner is considered to be taking only one course.

✚ Function 1 – 100 LOC: Manager the Topics

1.1.Add Topics to catalog

- Create a submenu.
- **Remember that the constraints must be checked: code can not duplicate**
- **Add the new topic to the collection.**
- Ask continuously create new topic or go back to the main menu.

1.2.Update Topic

- Create a submenu.
- **Require to enter the topic code**
- If a code **does NOT EXIST**, the notification “**The topic does not exist**”. **Otherwise**, we can start inputting new information about topic and update it.
- If new information is **blank**, then not change old information.
- New information **must be validated**.
- The system must **print out the result of the update**.
- After updating, the program returns to the main screen.

1.3.Delete Topic

- Create a submenu
- **Require to enter the topic code**
- **Before the delete system must show, confirmed message.**
- **Show the result: success or fail.**
- After deleting, the program **returns to the main screen**.

1.4.Display all Topics

- Show all the data in the topics' collection on the screen.
- **Sorted in ascending order by name**

Function 2 – 150 LOC: Manager the Course

2.1.Add Course

- Create a submenu.
- **Remember that the constraints must be checked: code can not duplicate**
- **Add the new course to the collection.**
- Ask continuously create new course or go back to the main menu.

2.2.Update Course

- Create a submenu.
- **Require to enter the course code**
- If a code **does NOT EXIST**, the notification “**The course does not exist**”. **Otherwise**, we can start inputting new information about course and update.
- If new information is **blank**, then not change old information.
- New information must be **validated**.
- The system must **print out the result of the update**.
- After updating, the program returns to the main screen.

2.3.Delete Course

- Create a submenu
- **Require to enter the topic code**
- **Before the delete system must show, confirmed message.**
- **Show the result: success or fail.**
- After deleting, the program **returns to the main screen**.

2.4.Display Course information

- **Show all** the data in the course' collection on the screen.
- The information includes **status of course, number of learner pass or fail, incomes**.
- **Sorted in ascending order by begin date**

Function 3 – 100 LOC: Manager the Learner

3.1.Add Learner to Course

- Create a submenu.
- **Remember that the constraints must be checked by the size of class.**
- **Add the new Learner to the Course information.**
- Ask continuously create new course or go back to the main menu.

3.2. Enter scores for learners

- Create a submenu.
- **Require to enter the learner code**
- If a code does **NOT EXIST**, the notification “**The learner does not exist**”. Otherwise, we can start inputting scores of learner and update.
- Remember new information must be **validated**.
- The system must **print out the result of the update**.
- After **updating**, the program **returns to the main screen**.

3.3. Display Learner information

- **Show all** the data of the learner on the screen.
- The information includes status of learner: **pass or fail, gpa**.

Function 4 – 50 LOC: Search information

4.1. Search Topic

- **Display all** topics on the screen whose **name contains the search information**.

4.2. Search Course

- The information includes **status of course, number of learner pass or fail, tuition fees**.
- **By Topic**: Display all courses on the screen that have the same topic code as the search information
- **By Name**: Display all courses on the screen whose name contains the search information

Function 5 – 50 LOC: Save Topics, Courses and Learner to file.

- **Store** the data of Topics, Courses and Learner to **file**.
- Reload data when the program starts.

- ❖ The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.
- ❖ The lecturer will explain the requirement only once on the first slot of the assignment.