

LAB211 Assignment

Type:	Long Assignment
Code:	J1.L.P0027
LOC:	300
Slot(s):	N/A

Title

Mountain Hiking Challenge Registration.

Background

FPT University has launched a movement titled *"Because a Generation of Intellectuals Dares to Think and Do,"* sponsored by Viettel and VNPT. To support this initiative, a program needs to be developed to **manage the registration information** of FPT University students.

The program should enable students from all campuses nationwide to register and select mountains to conquer (a list of mountains is Sprovided in the attached MountainList.csv file). The program's main features should include:

- Allowing new student **registrations**.
- Managing registration data.
- Collecting and displaying statistical information related to the registrations.

Students are required to analyze and design the program using an Object-Oriented Programming (OOP) approach. The development process must incorporate OOP principles, such as abstraction, polymorphism, encapsulation, and inheritance.

Program Specifications

Develop a program to manage the registration information of students at FPT University with the following functionalities:

1. **New Registration:** Add a new student registration.
2. **Update Registration Information:** Modify existing registration details.
3. **Display Registered List:** Show the list of all registered students.
4. **Delete Registration Information:** Remove a student's registration record.
5. **Search Participants by Name:** Find registered students based on their names.
6. **Filter Data by Campus:** Display registrations specific to a campus.
7. **Statistics of Registration Numbers by Location:** Generate statistics on the number of registrations for each location.
8. **Save Data to File:** Store registration data in a file.
9. **Exit the Program:** End the program execution.

Features:

This program contains the following functions:

▪ Function 1: New registration - 50 LOC

Description:

This function allows the creation of a new student registration by collecting necessary details, validating inputs, and calculating the applicable tuition fee.

Input Requirements:

The function requires the following **student details**:

- **Student ID**: A unique 8-character string. The first two characters represent the campus code (SE, HE, DE, QE, CE for FPTU's five campuses), followed by six numeric characters.
- **Name**: A non-empty string between 2 and 20 characters long.
- **Phone Number**: A 10-digit number belonging to a network operator in Vietnam.
- **Email**: A valid email address in standard format.
- **Mountain Code**: A code from the list of mountain peaks provided in the **MountainList.csv** file.
- **Tuition Fee**: Default is 6,000,000 VND. Students using phone numbers managed by Viettel or VNPT are eligible for a 35% discount due to sponsorship.

Validation Rules:

1. Student ID:

- Must start with a valid campus code (SE, HE, DE, QE, CE).
- The remaining six characters must be numeric.
- Must be unique.

2. Name:

- Cannot be empty.
- Length must be between 2 and 20 characters.

3. Phone Number:

- Must contain exactly 10 digits.
- Must belong to a valid Vietnamese network operator.

4. Email:

- Must follow standard email formatting (e.g., example@domain.com).

5. Mountain Code:

- Must match a code from the MountainList.csv file.

6. Tuition Fee Calculation:

- Default: 6,000,000 VND.
- Discount: 35% for phone numbers managed by Viettel or VNPT.

Operation Workflow:

1. Prompt the user to input student details.
2. Validate each input based on the rules above.
3. Calculate the tuition fee based on the phone number.
4. Save the registration record if all inputs are valid.
5. Return to the main menu, after adding new registration information

▪ Function 2: Update registration information – 50 LOC

Description:

This function allows users to update the registration information of a student who has already registered. The information that can be updated includes:

- Name
- Phone number
- Email
- Mountain peak code

If the student code entered does not exist in the registration records, the program will notify the user.

Input Requirements:

1. **Student Code:** The unique identifier of the student (8 characters, as per the validation rules for registration).
2. Fields for update:
 - **Name:** Must adhere to the rules of being non-empty and between 2 and 20 characters long.
 - **Phone Number:** Must contain exactly 10 digits and belong to a Vietnamese network operator.
 - **Email:** Must follow standard email formatting.
 - **Mountain Peak Code:** Must be a valid code from the MountainList.csv file.

Validation Rules:

- Ensure that the student code exists in the registration database before allowing updates.
- Apply the respective validation rules for each field being updated.
- Keep old information, if not enter new data.

Operation Workflow:

1. Prompt the user to enter the **Student Code**.
2. Check if the student exists in the registration records:
 - If the student exists:
 - a. Prompt the user to update the desired fields (Name, Phone Number, Email, Mountain Peak Code).
 - b. Validate the inputs based on the specified rules.
 - c. Save the updated information.
 - d. Display a success message.
 - If the student does not exist:
 - Display the message: "**This student has not registered yet.**"
3. After handling (or lack thereof), return to the main menu

Function 3: Display registered list – 20 LOC**Description:**

This function displays a list of all students who have registered for the program. If no registration data is available, the program should notify the user.

Operation Workflow:

1. Check the Registration List:

- Retrieve the current list of registered students from the system.

2. Display Data:

- If the registration list contains entries:
 - Display the following details for each student in a formatted table or list:
 - Student ID
 - Name
 - Phone Number
 - Mountain Peak Code
 - Tuition Fee
- If the registration list is empty:
 - Display the message: **"No students have registered yet."**

3. Return to Main Menu:

- After handling, return to the main menu.

Sample Output:

Case 1: List Contains Data

Registered Students:

Student ID	Name	Phone	Peak Code	Fee
SE123456	John Doe	0987654321	MT01	3,900,000
HE654321	Jane Smith	0902345678	MT05	6,000,000

Case 2: List is Empty

No students have registered yet.

▪ Function 4: Delete registration information – 30 LOC

Description:

This function allows users to delete a registered student's information based on their Student Code. If the Student Code exists, the program will display the relevant details and ask for confirmation before proceeding with the deletion. If the student is not registered, an appropriate notification will be displayed.

Operation Workflow:

1. Input the Student Code:

- Prompt the user to enter the **Student Code**.

2. Check if the Student Exists:

- Search the registration records for the entered Student Code.

3. Handle Cases:

- **If the student exists:**
 - Display the student's details (e.g., Name, Phone Number, Email, Mountain Peak Code, Tuition Fee).
 - Ask the user for confirmation to delete the record:
 - If the user confirms, delete the record and display the message:
"The registration has been successfully deleted."
 - If the user cancels, return to the main menu without making changes.
- **If the student does not exist:**
 - Display the message: **"This student has not registered yet."**

4. Return to Main Menu:

- After handling the deletion (or lack thereof), return to the main menu.

Validation Rules:

- Ensure that the Student Code follows the correct format (e.g., 8 characters, campus prefix, and unique identifier).
- Only delete the record if the student exists and the user confirms the action.

Sample Messages:

Case 1: Student Exists and Confirmation is Given

Student Details:

```
-----  
Student ID: SE123456  
Name      : John Doe  
Phone     : 0987654321  
Mountain  : MT01  
Fee       : 3,900,000  
-----
```

Are you sure you want to delete this registration? (Y/N): Y

The registration has been successfully deleted.

Case 2: Student Does Not Exist

This student has not registered yet.

▪ **Function 5: Search participants by Name – 25 LOC**

Description:

This function allows users to search for students by their name. The user can enter either the full name or a partial name, and the system will display all matching student names. If no students match the search criteria, an appropriate message will be shown.

Operation Workflow:

1. Input the Name:

- Prompt the user to enter the name or partial name of the student they want to search for.

2. Search Logic:

- Perform a search across the student records to find names that match the input.
- If multiple students have the same name, they should all be displayed.

3. Handle Cases:

- **If matching students are found:**
 - Display the list of matching students with details such as student ID, name, phone number, email, and mountain peak.
- **If no matches are found:**
 - Display the message: "No one matches the search criteria!".

4. Return to Main Menu:

- After displaying the results or the message, return to the main menu.

Sample Output:

Case 1: Matching Students Found

Matching Students:

Student ID	Name	Phone	Peak Code	Fee
SE123456	John Doe	0987654321	MT01	4500000
SE654321	Jane Doe	0902345678	MT02	6000000

Case 2: No Matches Found

No one matches the search criteria!

▪ Function 6: Filter data by Campus – 25 LOC

Description:

This function filters and displays a list of students registered under a specific campus. Users can select a campus by its code, and the program will display the corresponding students. If no students are registered for the selected campus, an appropriate message will be shown.

Campus Codes and Locations:

- **CE:** Can Tho
- **DE:** Da Nang
- **HE:** Ha Noi
- **SE:** Ho Chi Minh
- **QE:** Quy Nhon

Operation Workflow:

1. Prompt for Campus Code:

- Ask the user to enter a campus code (e.g., CE, DE, HE, SE, QE).

2. Filter Data:

- Retrieve all registered students whose Student ID begins with the entered campus code.

3. Handle Cases:

- **If students are found:**
 - Display the filtered list of students, showing details like Name, Phone Number, Email, Mountain Peak Code, and Tuition Fee.
- **If no students are found:**
 - Display the message: "**No students have registered under this campus.**"

4. Return to Main Menu:

- After displaying the results or the message, return to the main menu.

Sample Output:

Case 1: Students Registered Under a Campus

Input: **HE**

Registered Students Under Ha Noi Campus (HE):

Student ID	Name	Phone	Mountain	Fee
HE123456	John Doe	0987654321	MT01	3,900,000
HE654321	Jane Smith	0902345678	MT05	6,000,000

Case 2: No Students Registered Under a Campus

Input: **QE**

No students have registered under this campus.

▪ **Function 7: Statistics of registration number by Location – 50 LOC**

Description:

This function displays statistics on the number of students registered for each mountain peak they wish to conquer. It includes details like the peak name, the number of participants, and the total tuition fee to be collected. Peaks with no registered participants will not be shown.

Operation Workflow:

1. Retrieve Data:

- Collect all registration records to determine which mountains students are interested in conquering.

2. Calculate Statistics:

- For each mountain peak:
 - Count the number of participants.
 - Calculate the total tuition fee based on the number of participants.
- Exclude peaks with no registered participants from the display.

3. Display the Results:

- Show the statistics in the following format:
 - **Peak Name:** The name of the mountain peak.
 - **Number of Participants:** The count of students registered.
 - **Total Cost:** The total tuition fee collected.
- Ensure that only peaks with participants are displayed.
- 4. **Return to Main Menu:**
 - After displaying the statistics, return to the main menu.

Sample Output:

Statistics of Registration by Mountain Peak:

Peak Name	Number of Participants	Total Cost
MT01	5	19,500,000
MT05	3	18,000,000

▪ **Function 8: Save data to registration file – 30 LOC**

Description:

This function saves all the registration data to a file in an object format. This allows for persistent storage and easy retrieval of student registration records. The data should be saved in a structured format that preserves the relationships between student details and registration information.

Operation Workflow:

1. **Data Collection:**
 - Gather all current registration data from the program, including student details (ID, name, phone, email, mountain peak, and tuition fee).
2. **Serialization:**
 - Convert the registration data into an object format suitable for file storage as a binary object file.
3. **Save to File:**
 - Write the serialized data to a file. The file should be named appropriately (e.g., registrations.dat).
4. **Confirmation Message:**
 - Display a confirmation message once the data is successfully saved.
5. **Return to Main Menu:**
 - After saving the data, prompt the user to return to the main menu.

Sample Confirmation Message:

Registration data has been successfully saved to `registrations.dat`.

▪ **Function 9: Exit the program – 20 LOC**

Description:

This function exits the program. The program will only exit when the user presses the designated exit key (9). If any other function key is pressed, a message will be displayed indicating that the function is not

available. If the user chooses to exit without saving the data, a reminder will prompt the user to save the information before exiting.

Operation Workflow:

1. User Input:

- Display a menu or prompt where the user can select a function key.
- If the user selects function keys other than those listed on the menu:
 - Display the message: **"This function is not available."**

2. Exit Confirmation:

- If the user selects the exit key (9):
 - If there are unsaved changes:
 - Prompt the user with the message:
"Do you want to save the changes before exiting? (Y/N)"
 - If the user responds with 'Y':
 - Call the function to save the data to the file.
 - If the user responds with 'N':
 - Exit the program directly.
 - If there are no changes to save, exit the program immediately.

3. Return to Main Menu:

- If the user selects a function key other than the exit key:
 - Display the message: **"This function is not available."** and remain in the program.

4. Reminder Message:

- If the user exits without saving data, display a reminder message:
You have unsaved changes. Are you sure you want to exit without saving? (Y/N)

The above specifications **provide basic information**. You are required to *conduct a detailed requirements analysis and build the application based on the real-world requirements*.

The lecturer will explain the **full set of requirements only once during the initial slot of the assignment**.