THE AMERICAN UNIVERSITY IN CAIRO
SCHOOL OF SCIENCES AND ENGINEERING

(CSCE 1102)
Fundamentals of Computing II Lab

Session 07 – STL Exercise

Course Application using Vector

Implement a program using **Vectors** to take from user courses till he stops then display the total number of students enrolled in courses. Where each **course** has **Name** and **number of students.**

Sample Run:

Press 1 to Add Course Press 2 to Display Total Students Press 3 to Exit
1
enter course Data

DS 30

Press 1 to Add Course Press 2 to Display Total Students Press 3 to Exit

1
enter course Data

SP 40

Press 1 to Add Course Press 2 to Display Total Students Press 3 to Exit

2

Total =70

Press 1 to Add Course Press 2 to Display Total Students Press 3 to Exit

3

- Course Class Contains:
  - Two private attributes: number of students (int), and name (string).
  - Two Public functions: readData, and getNumOfStd.

- Main Function:
  - Include the STL library for the vector.
  - Define vector of courses.
  - Loop until user enter (3) and add the following inside it:
    - Display the three options to the user (cout).
    - Switch case of if conditions:
      - Case 1: define object course, call readData function, and push_back the object to the vector.
      - Case 2: calculate the total number of students in the vector and display it.
      - Case 3: end the loop.

- **Task Organizer with Queue (Menu-Based)**

Problem Statement:

In this task, you'll build a basic task organizer using a queue in C++. The program should display a menu to the user with options to manage tasks. Each task is represented by a description string, which the user provides.

Requirements:

1. **Use a queue<string> to store tasks.**
2. **The program should repeatedly display a menu with the following options:**
   - **1: Add a new task (the user enters a task description).**
   - **2: Finish the next task and display it as "completed."**
   - **3: View the next task without removing it from the queue.**
3. **After completing each action, the program should return to the menu until the user chooses to exit.**

Example Output:

```
Task Organizer Menu:
1. Add a new task
2. Finish and print the next task
3. View the next task
4. Exit
Enter choice: 1
Enter a new task: Finish homework
Task Organizer Menu:
1. Add a new task
2. Finish and print the next task
3. View the next task
4. Exit
Enter choice: 3
```

(CSCE 1102)
Fundamentals of Computing II Lab

Next task: Finish homework

```
Task Organizer Menu:
1. Add a new task
2. Finish and print the next task
3. View the next task
4. Exit
Enter choice: 2
Completing task: Finish homework
Task Organizer Menu:
1. Add a new task
2. Finish and print the next task
3. View the next task
4. Exit
Enter choice: 4
Exiting...
```

- **Card Organizer with Stack (Menu-Based)**

Problem Statement:

In this task, you'll use a stack to organize and manage cards. Each card is represented by a description string (for example, "Ace of Spades"). The program should display a menu to the user to allow for adding and removing cards, as well as viewing the top card on the stack. This models a "last in, first out" (LIFO) system.

Requirements:

1. **Use a stack<string> to manage the cards.**
2. **The program should repeatedly display a menu with the following options:**
   - **1: Add a new card to the top of the stack (the user enters the card's description).**
   - **2: Remove and display the top card from the stack.**
   - **3: View the top card on the stack without removing it.**
3. **After completing each action, the program should return to the menu until the user chooses to exit.**

(CSCE 1102)
Fundamentals of Computing II Lab

Example Output:

```
Card Organizer Menu:
1. Add a new card
2. Remove and display the top card
3. View the top card
4. Exit
Enter choice: 1
Enter a new card: Ace of Spades
Card Organizer Menu:
1. Add a new card
2. Remove and display the top card
3. View the top card
4. Exit
Enter choice: 3
Top card: Ace of Spades
Card Organizer Menu:
1. Add a new card
2. Remove and display the top card
3. View the top card
4. Exit
Enter choice: 2
Removing card: Ace of Spades
Card Organizer Menu:
1. Add a new card
2. Remove and display the top card
3. View the top card
4. Exit
Enter choice: 4
Exiting...
```

This program allows users to manage a stack of cards, adding, viewing, and removing cards in a last-in, first-out order, similar to a real-life stack of cards,

Good Luck,