King Saud University **Procedural Programming** **CSC215**
College of Computer and Information Sciences **2023 - Semester 451**
Computer Science Department **LAB 08 Structures**

Lab overview:

We would like to create a Student Course Management System using user-defined structures. It is a simple command-line program that allows users to manage students and their enrollment in courses.

Objectives:

- Creating user-defined structures
- Creating an instance of structures and an array of structures
- How to initialize and access structures members
- How to pass structures to functions

Exercise 1. Defining Student Structure: (1 point)

The Student struct represents an individual student in the Student Course Management System. Here's a description of the fields within the Student struct:

Id (int): This field stores a unique identifier for each student. It serves as a primary key to distinguish one student from another.
Name (char[50]): The name field is a character array that stores the student's full name. It is designed to hold up to 50 characters, allowing for both first and last names.
GPA (float): The GPA field is of the float data type and stores the GPA of the student.

Create a student struct based on the description given above. Afterward, create an array of 10 students, read their info as an input, and then print their details in a neat way.

Exercise 2. Defining Course Structure: (1 point)

The Course struct represents a specific academic course within the Student Course Management System. Here's a description of the fields within the Course struct:

Course Number (int): This field serves as a unique identifier for each course, allowing the system to distinguish one course from another.
Title (char[50]): The title field is a character array that stores the name or title of the course. It can accommodate up to 50 characters, allowing for a descriptive name of the course.
Students (struct Student[MAX_STUDENTS_PER_COURSE]): This field is an array of struct Student and is used to store information about the students who are currently enrolled in the course. The array allows for a maximum number of students (specified by MAX_STUDENTS_PER_COURSE) to be enrolled in the course.
Number of Students (int): The numStudents field is an integer that keeps track of the current number of students enrolled in the course. It is used to manage the capacity of the course and ensure that the maximum number of students is not exceeded.

Extend the previous exercise and create a course struct based on the description given above.

Exercise 3. Functions of Student Course Management System: (3 points)

The program has the following functions:

```
void addStudentToCourse(struct Student student, struct Course *course);
```

This function takes a Student struct, representing a specific student, and a pointer to a Course struct, representing the course in which the student will be enrolled. It is used to add a student to a particular course.

```
void withdrawStudent(struct Course *course, int studentId);
```

This function takes a pointer to a Course struct, representing a specific academic course, and the studentId of the student to be withdrawn. It is used to remove a particular student from the specified course.

```
void displayStudentsInCourse(struct Course course);
```

This function takes a Course struct, representing a specific academic course. It displays a list of students who are currently enrolled in the specified course.

Implement each of these functions and then write a main function test to your program capabilities. Your program should display a menu showing a list of the services available.

Lab Assignment: (5 points)

Write a function to display a student's schedule. The function prototype should look like this:

```
void displaySchedule(struct Course courses[], int numCourses, int studentId);
```

Bonus exercise: (5 points)

Write a function to find and return course with highest number of enrolled students

```
struct Course findCourseWithMostStudents(struct Course courses[], int numCourses);
```