National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	OOP - LAB	Course Code:	CL1004
Program:	BCS-2E	Semester:	Spring 2024
Duration:	2 Hours	Total Marks:	100
Paper Date:		Weight:	%
Section:		Pages:	2
Exam:	Midterm		

Student Name:_____ **Roll No.**_____ **Section:**_____

Instruction/Notes:

- We will check your code for plagiarism. If plagiarism is found, it will result in an F grade in the lab.
- In case of any ambiguity make a suitable assumption.
- No cell phones are allowed. Sharing of USBs or any other items is not allowed.
- You are not allowed to have any helping code with you.
- The path for submission is: **\\cactus1\Xeon**

Question 1: University Grading System using Classes and Pointers marks: 60

Universities often use grading systems to evaluate students' performance in courses. These grading systems assign letter grades based on students' scores in assignments, exams, and other assessments. In this programming exercise, you are tasked with designing a university grading system using classes and pointers.

Your task is to create a program that models a university grading system. The system should be able to handle multiple courses and students, calculate grades, and display student transcripts. Each course may have different grading criteria, such as weightage for assignments, quizzes, exams, etc.

Here are the requirements for the university grading system:

1. Create a class named Course to represent a course. Each Course object should store the course name, maximum score for each assessment component (assignments, quizzes, exams), and the weightage of each component in the final grade calculation.
2. Implement a class named Student to represent a student. Each Student object should store the student's name, ID, enrolled courses, and scores for each assessment component.
3. Implement necessary member functions in the Course class to calculate the final grade for a student enrolled in the course. The final grade should be calculated based on the scores of assessment components and their weightage.
4. Implement necessary member functions in the Student class to enroll in courses, submit scores for assessment components, and generate transcripts.
5. Use dynamic arrays to manage multiple courses and students efficiently.
6. Use copy constructors to initialize values for Course and Student objects.

Ensure error handling and validation for input data. Test your program with multiple courses and students to verify its functionality.

Question 2: Sales Data Analysis Tool with Dynamic 2D Arrays marks:40

You have embarked on the development of a sophisticated sales data analysis tool aimed at providing insights into monthly sales performance for various products. The core functionality of this tool revolves around dynamically allocating memory for a 2D array to store the monthly sales data, followed by calculating the total sales for each product and displaying the results in a clear and informative manner. Here's a detailed breakdown:

1. The foundation of the sales data analysis tool is a dynamic 2D array designed to store the monthly sales data for different products.
2. Each row of the array represents a product, and each column represents a month sale.
3. The number of rows (products) and columns (months) in the array will be determined dynamically based on user input as which product have the highest sale in every month like if coke was highest sale in Sep but in Nov honey has the highest sale.
4. Proper memory allocation and deallocation will be implemented to prevent memory leaks and ensure efficient use of system resources.
5. After calculating the total sales and displaying the results, the dynamically allocated memory for the 2D array will be freed to release memory.