National University of Computer and Emerging Sciences

Islamabad Campus

CS1004		**Object Oriented Programming**	
Assignment	**1**	**Time: 180 minutes**	**Marks: 100**
Date: 26-Sep-2024			
Roll Number:		**Name:**	

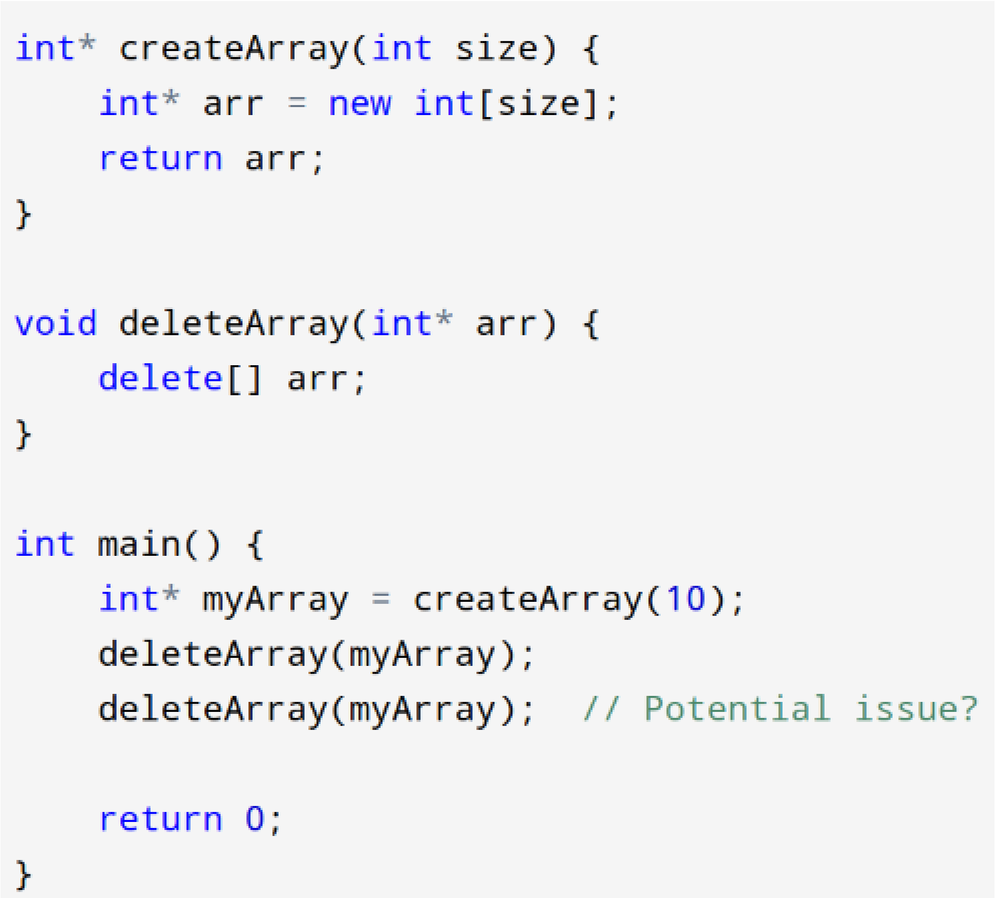
Pointers
Question No. 1 [Marks: 10]

Write a program to dynamically allocate a 2D array using pointers, where the size of each row may vary (a jagged array).

A) The program should allow the user to input the size of each row and initialize the array.
B) Implement a function to calculate the sum of each row and print it.
C) Write a deletion function to deallocate the whole memory.

Pointers
Question No. 2 [Marks: 10]

Analyze the following code and identify any memory management issues, such as memory leaks or double-free errors. Provide corrections to the code.



```cpp
int* createArray(int size) {
    int* arr = new int[size];
    return arr;
}

void deleteArray(int* arr) {
    delete[] arr;
}

int main() {
    int* myArray = createArray(10);
    deleteArray(myArray);
    deleteArray(myArray);  // Potential issue?

    return 0;
}
```

Pointers	
Question No. 3	**[Marks: 10]**

Write a test function that simulates various scenarios where dynamic memory allocation could cause problems, such as:

A. Memory leaks.
B. Dangling pointers.
C. Double-free errors

Pointers	
Question No. 4	**[Marks: 10]**

Write a program that simulates **memory fragmentation**. Your program should:

- Allocate and deallocate blocks of memory of random sizes (using new and delete) in a loop. The random size should be from 0 to roll_number%100 bytes.
- Keep track of memory usage and simulate fragmentation by randomly choosing blocks to deallocate and allocate.

- Provide a report showing the memory usage and fragmentation level after several iterations.

Recursion

Question No. 5 **[Marks: 10]**

Implement a recursive solution to solve the **N-Queens** problem. The goal is to place N queens on an N x N chessboard such that no two queens attack each other. Write a function that prints one possible solution.

The queen attack means there are two queens in same row, same column or in same diagonal.

Recursion

Question No. 6 **[Marks: 10]**

Given a 2D array representing a maze where 0 is a walkable path and 1 is a wall, write a recursive function to determine if there is a path from the top-left corner (0, 0) to the bottom-right corner (N-1, N-1). The function should return a boolean indicating if a path exists, and if one does, it should print the path.

Recursion

Question No. 7 **[Marks: 10]**

Using recursion, implement a function to calculate the binomial coefficient C(n, k), which represents the number of ways to choose k elements from n elements (combinations). You should use the following recursive formula:

C(n, k) = C(n-1, k-1) + C(n-1, k)

C(n,n)=1 where n and k are same

C(n,0)=1 where k is zero

Structures

Question No. 8 **[Marks: 10]**

Define a structure Book that holds information about a book, including:

- Title (string)
- Author (string)
- Year of publication (int)
- Price (double)

Write a program to dynamically create an array of Book structures, allow the user to input details for each book, and then sort the books by price in ascending order.

Structures	
Question No. 9	**[Marks: 10]**

Create a structure Course that has the following fields:

- Course name (string)
- Number of students enrolled (int)
- A dynamically allocated array of structures Student, where each Student structure contains:
  - Name (string)
  - Age (int)
  - GPA (double)

Write functions to:

- Initialize the Course structure, including the array of students.
- Input student information.
- Calculate and print the average GPA of all students in a course.
- Ensure proper memory management (i.e., allocate and deallocate memory for the students array inside the Course structure).

Structures	
Question No. 10	**[Marks: 10]**

Design a structure Employee to store information about an employee:

- Employee ID (int)
- Name (string)
- Department (string)
- Salary (double)

Additionally, create another structure Department to store information about a department:

- Department name (string)
- Number of employees (int)
- A dynamically allocated array of Employee structures

Write a program to:

- Dynamically allocate memory for the employees in each department.
- Input and store the details of multiple departments and employees.

- Print the details of employees in a specific department with a salary greater than a user-defined threshold.