C++ Assignment 2

1. Write a C++ program to declare pointers to various data types (int, double, char, float). Allocate memory dynamically for each type using new, assign values, and print the addresses and values.

2. Create a C++ program that shows the dangers of wild pointers. Declare an uninitialized pointer, assign it to a local variable, and attempt to print the pointer's value.

3. Write a program to dynamically allocate memory for an integer using a pointer, assign a value through the pointer, and print the value and address.

4. Develop a program that initializes pointers to NULL and demonstrates checking for NULL before dereferencing. Use this approach in a function that safely handles pointer inputs and processes them only if they are not NULL.

5. Write a program that demonstrates the use of const with pointers. Create scenarios for:

a. A constant pointer to a non-constant integer.
b. A pointer to a constant integer.
c. A constant pointer to a constant integer.

6. Write a C++ program that defines a structure to store information about a student (name, roll number, marks in three subjects). Create functions to initialize, calculate the average marks, and print the structure members.

7. Write a program to create an array of student structures dynamically. Initialize the array with data, calculate averages and grades for each student, and print their details.

8. Write a C++ program to demonstrate the use of namespaces. Create two namespaces with overlapping function names and variables. Use namespace aliases and the using directive to resolve conflicts and call the appropriate functions.

9. Develop a C++ program for an employee management system using structures. Implement functionalities to add new employees and display employee details.

Requirements:

1. Define a structure Employee with attributes: name (string), ID (integer), and salary (float).
2. Implement a function to add a new employee to the system, accepting input for name, ID, and salary.
3. Implement a function to display details of all employees in the system.

10. Develop a C++ program for a simple book inventory management system using structures. Implement functionalities to add new books, display book details, and search for books by id.

Requirements:

1. Define a structure Book with attributes: title (string), author (string), id (integer), and price (float).
2. Implement a function to add a new book to the inventory, accepting input for title, author, id, and price.
3. Implement a function to display details of all books in the inventory.
4. Implement a function to search for a book by id and display its details.