



CL-217

Object Oriented Programming

Lab No 3

Objectives:

- File handling
- Pointers
- Pointer Variable Declarations and Initialization
- Referencing/Dereferencing & Functions
- DMA

Note: Carefully read the following instructions (*Each instruction contains a weightage*)

1. There must be a block of comments at start of every question's code by students; the block should contain brief description about functionality of code.
2. Comment on every function about its functionality.
3. Use understandable name of variables.
4. Proper indentation of code is essential
5. Write a C++ statement(s) for each of the following task one after the other, in the same order.
6. Make a Microsoft Word file and paste all of your C++ code with all possible screenshots of every **task output in MS word and submit .cpp file with word file.**
7. Make separate .cpp files for all tasks and use this format **22F-1234_Task1.cpp.**
8. First think about statement problems and then write/draw your logic on copy.
9. After copy pencil work, code the problem statement on MS Studio C++ compiler.
10. At the end when you done your tasks, attached C++ created files in MS word file and make your submission on Google classroom. (Make sure your submission is completed).
11. Please submit your file in this format **22F-1234_L1.**
12. Do not submit your assignment **after deadline.**
- 13. Do not copy code from any source otherwise you will be penalized with negative marks.**



Problem 1: | Pointer with 2d array

Write a code to input a 2 by 2 matrix using pointers. Then add value 100 in all values and print it in matrix form.

Problem 2: | Dynamic Memory Allocation

Define an `int*` pointer variable `a`. Then:

1. Use **new** to make `a` point to a dynamic array of 5 cells of type `int`.
2. Write a loop to fill `a` with values 3, 7, 11, 15, 19.
3. Using **Print function** to print values stored in `a`.

Problem 3: | Dynamic Memory Allocation

Suppose you are developing a program that works with arrays of integers, and you find that you frequently need to duplicate the arrays. Rather than rewriting the array-duplicating code each time you need it, you decide to write a function that accepts an array and its size as arguments, creates a new array that is a copy of the argument array, and returns a pointer to the new array. The function will work as follows:

1. Accept an array and its size as arguments.
2. Dynamically allocate a new array that is the same size as the argument array.
3. Copy the elements of the argument array to the new array.
4. Return a pointer to the new array.
5. Release memory to memory heap(use delete in main)
6. `Int * copyFun(int arr[], int SIZE)`

Program Output Here are the

Original array contents:

100 200 300 400 500 10 20 30 40 50 60

Here are the duplicate arrays:

100 200 300 400 500 10 20 30 40 50 60

Note Do not accept negative numbers for test scores.

Problem 4: | (Double pointers, DMA, Passing pointers as parameter, Returning pointers from function)

Write a C++ program to add 2 matrices A and B. Matrices should have user defined size. Use double pointers and dynamic memory allocation to create matrices. If both are equal size, then



result of A+B should be saved in C matrix otherwise terminate the program. Write the following functions

- Void Input(int** p, int row, int col) // this function will input the p matrix
- Void Display(int** p, int row, int col) // this function will output the p matrix to console
- Int** Sum(int** p, int row, int col, int ** q , int row2, int col2) // this function will take sum of two matrices and return the resultant matrix.
- Print the resultant matrix in main using Display() function.
- Display A , B and C in matrix on console.

Example

A = 1 2

3 4

B = 6 5

7 1

C = 7 6

10 5

Problem 5: (Double pointers, Dynamic Memory Allocation)

Write a C++ program to build a matrix that have different number of elements in each row (different number of column in each row) using two-dimensional dynamic array. For Example

```
Enter the number of rows=3
Enter the number of col in row 1 =3
Enter 3 elements in row 1 =1 2 3
Enter the number of col in row 2 =5
Enter 5 elements in row 2 =4 5 6 7 8
Enter the number of col in row 3 =2
Enter 2 elements in row 3 =1 2

Matrix is
1 2 3
4 5 6 7 8
1 2
```

Your program must contain two functions. One for filling the elements into your two dimensional array and other for printing that array or matrix.



Problem 6: (Double pointers, Dynamic Memory Allocation)

Input a sentence from the user. Use full stop, space and comma as word separators. Each word should be stored in a 2D array whose columns vary in size and each row stores one word as a NULL terminated string. For example, if the user inputs:

Hello how are you?

It should be stored as:

| | | | | | |
|---|---|---|------|------|------|
| H | e | l | l | o | NULL |
| h | o | w | NULL | | |
| a | r | e | NULL | | |
| y | o | u | ? | NULL | |

Proper code indentation will hold extra marks !

Best of luck 😊

You are done with your exercise, submit on Teams at given time.