



FAST – NU, CFD Campus

National University of Computer and Emerging Sciences



OOP
Assignment # 2

Department of Computer Science



Objectives

1. Pointers
2. DMA
3. Structures

Note: Carefully read the following instructions.

1. Make a word document with the convention “ROLLNO_SECTION_” and put all your C++ source code in it.
2. After every question, paste a screenshot of your working code below the source code in the document file.
3. At the end, when you are done with your tasks, make your submission on Google Classroom.

Problem 1:Pointers

Write a program in C++ to Calculate the length of the string using a pointer.

Problem 2:DMA

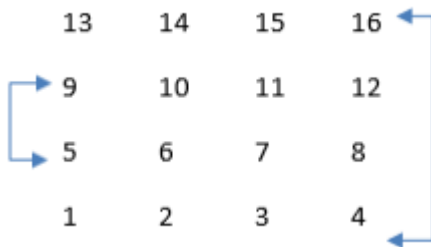
Write a C++ program that will take input of a 1D dynamic array named Matrix. Now you will treat this array as a 2D Matrix. User will input the size of the array. Make sure it should be a square matrix otherwise take the input again. Let's Say we have the following array of size=16 with row_size=column_size=4. Matrix: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 Its matrix representation will be



Matrix:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Now your task is to call a function `SwapIt(Matrix)` to swap its first row with last row, 2nd row with 2nd last row (and so on for matrix with greater size). After this function call `DisplayMatrix()`.



Write your program for integer array named `Matrix` of `size=16` with `row_size=4` but your program should be generic enough that if user enter the `size=25` with `row_size=5` or any other size it should work fine. No other array should be used for this task. Your main should be in following flow.

```
Int* Matrix;
```

```
Input(int* Matrix, int size);
```

```
MatrixDisplay(int* Matrix, int size); // this function will display the Matrix array in matrix form
```

```
SwapIt(int* Matrix, int size); // this function will swap the rows
```

```
MatrixDisplay(int* Matrix, int size); // Now this function will display the Matrix array in matrix form with swapped row
```



Problem 3: Structure

It's time to help your academic officer to automate their Registration Process.

Create a struct student having all essential attributes to store the record of a student. The struct should maintain a record of total number of students enrolled.

o The registration no. of a student, once assigned, cannot be changed. (Assigned on the time of registration (object creation))

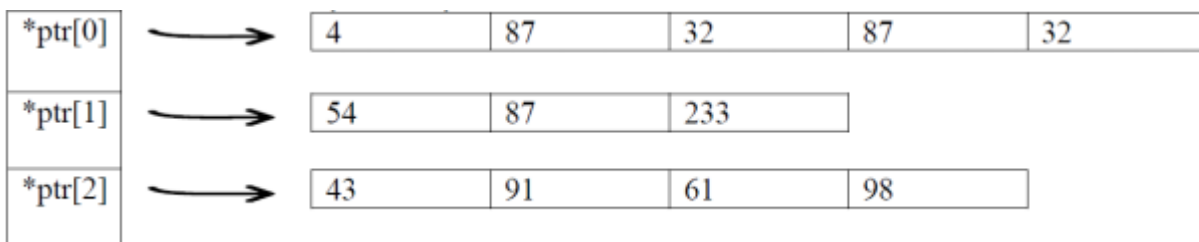
- Each student can register at most 5 and at least 3 courses.
- Registered courses can be dropped if that course has been registered for 5 days.
- Students record (e.g. CNIC, address etc.) can be updated.
- Your program should be menu driven having following options
- Register a Student
- Register Courses
- Drop Courses
- Update Record
- Delete a Student
- Search a Student (Should return an object of Student Class)
- Get a list of students (Studying a specific course together)
- Total count of Students Enrolled
- Write separate functions for each of the above operations.

**Problem 4: DMA**

Design a C++ program that define a 2-Dimensional array with different number of values in each row. You need to fulfill all the requirement as given below,

- You must use double pointer to define and access the elements of array
- The system asks how many rows n the user want to define
- The system asks how many elements should have for each row [store those numbers in 1D array (DMA)]
- Define that rows with required amount of values in each row (already stored in 1D array name cols)
- Input values for each row one by one in 2D array
- Print all the values row wise from 2D array
- Delete the 2D array
- Delete the pointer array

For example, if user enter n=3 then you should have 3 rows in total, then enter cols = [5, 3 and 4]. The structure of your array should be like

**Problem 5: Nested structure**

You are creating a program to manage the information of a company's employees, which includes their names, ages, addresses, salaries, and benefits. Define a nested structure called "Benefit" that contains the following fields:

- the name of the benefit (string)
- the amount of the benefit (float)
-

Define another nested structure called "Employee" that contains the following fields:

- name (string)
- age (integer)
- address (string)
- salary (float)



- an array of Benefit structures (dynamic array)
-

Write a C++ program that asks the user to input the information of all the employees (name, age, address, salary, and the information of up to 5 benefits). The program should then display the information of all the employees, including their total compensation (salary plus the total amount of all their benefits).

Your program should meet the following requirements:

- The user should be able to input up to 50 employees.
- The program should use nested structures to organize the information of the employees and their benefits.
- The program should display the information of all the employees, including their total compensation.

Problem 6:Structure

Write a C++ program that asks the user to input the information of up to 50 books (title, author, publication year, and price) and passes an array of book structures to a function that sorts the books by price in descending order using the selection sort algorithm. The function should then return the sorted array of book structures. The program should then display the information of all the books in the sorted array, including the total cost of all the books.

Your program should meet the following requirements:

- The program should use a function to sort the array of book structures.
- The function should take the array of book structures as an argument by reference.
- The program should display the sorted array of book structures, including the total cost of all the books.



Problem 7: Structure and arrays

You are creating a program to manage the information of a company's employees, which includes their names, ages, addresses, and salaries. Define a structure called "Employee" that contains the following fields:

- name (string)
- age (integer)
- address (string)
- salary (float)

Write a C++ program that asks the user to input the information of an employee (name, age, address, and salary) and passes the employee structure to a function that calculates the employee's tax withholding based on the following formula:

- If the employee's salary is less than \$50,000, the tax withholding is 10% of the salary.
- If the employee's salary is between \$50,000 and \$100,000, the tax withholding is 15% of the salary.
- If the employee's salary is greater than \$100,000, the tax withholding is 20% of the salary.

The function should update the employee's salary field to reflect the tax withholding and return the updated employee structure. The program should then display the information of the employee, including the tax withholding amount.

Your program should meet the following requirements:

- The program should use a function to calculate the tax withholding.
- The function should take the employee structure as an argument by reference.
- The program should display the updated employee structure, including the tax withholding amount.



Problem 8: Structure and arrays

You are creating a program to manage the information of a fleet of ships, which includes the names, types, lengths, widths, and top speeds of each ship. Define a structure called "Ship" that contains the following fields:

- name (string)
- type (string)
- length (float)
- width (float)
- top speed (float)
-

Write a C++ program that asks the user to input the information of up to 10 ships (name, type, length, width, and top speed) and stores the ships in an array of ship structures. The program should then pass the array of ship structures to a function that calculates the average length and width of the ships in the array and returns the average values as a structure called "ShipStats". The program should then display the information of the ships, including the average length and width. Your program should meet the following requirements:

- The program should use an array of ship structures to store the information of the ships.
- The program should use a function to calculate the average length and width of the ships.
- The function should take the array of ship structures as an argument by reference.
- The program should display the information of the ships, including the average length and width.