Data Classures and Algorithms — Lab I

Objective

- Basic concepts of 2-D array
- Revision of clasess
- Introduction to *Attributes* and *Methods*

2-D Array

You’ve worked one-dimensional array in Programming Fundamental courses. As an array keeps track of multiple pieces of similar information in linear order. However, data associated with some applications like images, statistical survey are represented in two dimensions. A two-dimensional array is nothing more than the *array of arrays.*

Applications

As stated above, data in some applications can only be represented in 2-D arrays. In elementary classes school, you’ve studied *Matrices* (singular: Matrix). A matrix is a rectangular array of mathematical objects like whole numbers, real numbers and complex numbers etc.

Lab Task

Students are required to perform the following lab task during lab timings. Create *C++ Class* for attributes/methods and check their functionality in `main()` by creating a *menu*. Please note that attributes of a Class are the variables and methods of a Class are the functions written in it. Make your Class according to the template given below:

```
class TwoDArray
{
private:
      //write your attributes (varibales) here
public:
      //write your methods (functions) here
};
```

Mr. Donald teaches Mathematics courses to grade 8 in University Pattern School. He wants to provide simple console based application to his students, which can calculate addition, subtraction, constant multiplication, constant addiction, constant subtraction, multiplication etc. to check their answers during practice. He asks you to give him a favor to accomplish this task.

The application/program should be generic, and ask the student to give the number of rows and columns of two matrices. It’ll create two matrices and then it will ask the operation to perform.

Sample Output:

```
Rows of First Matrix: 3
Columns of First Matrix: 4
Rows of Second Matrix: 3
Columns of Second Matrix: 4

Options
1: To Input Matrix
```

```
2: To Display Matrix
3: Set Specific element using single number as index
4: Get Specific element using single number as index
5: To Add Matrices
6: To Subtract Matrices
7: To add Constant to Matrix
8: To Multiply Matrices
9: To Exit
```

Note: Matrices Multiplication is optional. Try it yourself at home after completing other functions.