National University of Computer and Emerging Sciences

Lab Manual 04
Object Oriented Programming – CL1004

Course Instructor	Dr. Saira Karim
Lab Instructor(s)	Ms. Amna Zulfiqar Mr. Muhammad Adeel
Section	BCS-2B
Semester	Spring 2023
Date	21-02-2023

Department of Computer Science
FAST-NU, Lahore, Pakistan

Lab Manual 04 –2D arrays and Intro to Class

Important Note:

- **You may find the syntax to accomplish these exercises from lecture demo.**
- **Add Necessary Comments in you code to justify your logic.**
- **Comment exercise number or statement at the start of your code**
- **Save each exercise in .cpp file with your roll no, ex and lab number e.g.**
- **22LXXXX_EX01_Lab01.cpp**
- **Place all of your exercises in a folder a Zip it (Do not create .rar file) with roll no and lab no. e.g. 22LXXX_Lab01.zip**
- **Make sure that the interface of your program is user friendly i.e. properly display information.**
- **Properly follow the coding standards.**

1. Exercise

Create a class "Employee" with private members "name", "id", "position", and "salary". Add two public member functions, one function **inputEmployeeInfo()** to input the employee's name, id, position, and salary and the other **printEmployeeInfo()** to output the Employee information. Drive the Program in main().

What will happen if you call **printEmployeeInfo()** on an Employee object without calling **inputEmployeeInfo()?**

2. Exercise

Create a class "Rectangle" with private members width and height. Add Following public member functions:

1. **void setWidth(float w)**
2. **void setHeight(float h)**
3. **float getArea()**
4. **void printDimensions()**
5. **void printArea()**

3. Exercise

Create a class "Matrix" that represents a 2D matrix with private member rows, column, and ****data** (a pointer to a dynamically allocated 2D integer array).

Add public member functions

1. **Matrix(int rows, int cols): allocate memory for the matrix**
2. **~Matrix():** Destructor to deallocate memory for the matrix
3. **void input()** to initialize the matrix with user input
4. **void print()** to print the matrix
5. **void transpose()** to transpose the matrix.

Sample output:

```
Enter number of rows: 2
Enter number of columns: 3
Enter value at [0][0]: 1
Enter value at [0][1]: 2
Enter value at [0][2]: 3
Enter value at [1][0]: 4
Enter value at [1][1]: 5
Enter value at [1][2]: 6
Matrix:
1 2 3
4 5 6
Matrix:
1 4
2 5
3  6
```