BLG252E - Object Oriented Programming
Homework-1

Assignment Date : 15.10.2018
Due Date : 05.11.2018 at 18:00

Write a C++ program to do the followings.

MATRIX CLASS

Define the class named **Matrix** with public members below.

- **Two-dimensional integer array** : Sizes are NxN. The N is a constant symbol with value of 3.
- **Default constructor:** Does nothing.
- **Parameterized constructor:** Takes a two-dimensional integer array as parameter.
  Function prototype: Matrix(int data[N][N]);
- **Overloaded operator+ :** Takes another Matrix object as parameter, returns a Matrix.
  Function should add the values of itself with the values of given Matrix, and should assign the results to a new Matrix object, and should return the new Matrix.
  Function prototype: Matrix operator+ (Matrix Other);
- **Transpose() function :** Takes no parameters, returns a Matrix.
  Function should perform transpose operation for the values of itself, and should assign the results to a new Matrix object,and should return the new Matrix. (Row and column values should be exchanged.)
  Function prototype: Matrix transpose();

OVERLOADED operator<<

Write the non-member **operator<<** function whose prototype is given below.

Function prototype: ostream& operator<< (ostream& ekran, Matrix Mat);

Function takes two parameters and returns the resulting stream output.
The given matrix object should be displayed on screen.
Function enables cascaded usage like cout << x << y << z.

MAIN PROGRAM

In main program, perform the tasks described below.

- Declare a two-dimensional integer array (**data1**), and initialize with data below.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- Declare another two-dimensional integer array (**data2**), and initialize with data below.

$$\begin{bmatrix} 10 & 20 & 30 \\ 40 & 50 & 60 \\ 70 & 80 & 90 \end{bmatrix}$$

- Declare a Matrix object named **A**, whose constructor parameter is **data1**.
- Declare a Matrix object named **B**, whose constructor parameter is **data2**.
- Declare two Matrix objects named **C** and **T** , without any constructor parameters.
- Call the overloaded **operator<<** function to display A and B matrices on screen.
- Call the overloaded **operator+** function of A, to add A with B, and assign the result to C.
- Call the overloaded **operator<<** function to display C matrix on screen.
- Call the **transpose()** function of A, and assign the result to T.
- Call the overloaded **operator<<** function to display T matrix on screen.

EXAMPLE SCREEN OUTPUT

```
A matrix =
1  2  3
4  5  6
7  8  9

B matrix =
10  20  30
40  50  60
70  80  90

C = A+B
11  22  33
44  55  66
77  88  99

T = A'
1  4  7
2  5  8
3  6  9

Press any key to continue . . .
```