

OOP Lab 2

Topic: Classes, Data members, Member Functions (Constructors, Destructors)

Marks: (18*5 + 10 = 100) each member function carries 5 marks, menu 10 marks

Release Date: 1-Oct-20 Thursday

Submission Date & Time: 03-Oct-20, 11:59 PM, Saturday

You may send your lab queries at: asktoknow42@gmail.com

Write a class **Matrix** providing functionalities of a mathematical matrix. Using this class you will implement a menu based program in main that acts like an application for user to work with matrices. The structure of class is given below.

Data Members (Attributes/Variables):

private:

int row; //row variable, representing row in a matrix

int col; //col variable, representing columns in a matrix

int ** ptr; //a double pointer that is used for creating 2D array (matrix) dynamically

Constructors:

public:

Matrix (); //default constructor

Matrix (int r , int c); // r is rows, c is columns

Matrix (Matrix &obj); //A special constructor called copy constructor, Read about it
//before you implement it and see why it is important while
//declaring dynamic memory for some variables

Destructor:

~Matrix ();

Getter Setter:

int get(int i, int j) // returns value of (i,j)th element of the matrix

void set(int i, int j, int val) // sets value of (i,j)th element of the matrix

Methods:

bool isSquare (); //returns true if the matrix is a square matrix

bool isIdentity (); //returns true if the matrix is an identity matrix

bool isDiagonal (); //returns true if the matrix is a diagonal matrix

bool isNullMatrix (); //returns true if the matrix is a Null matrix

bool isUpperTriangular (); //returns true if the matrix is an upper triangular matrix

bool isLowerTriangular (); //returns true if the matrix is a lower triangular matrix

bool isEqual (Matrix& m); //returns true if the size and values of **calling matrix object** and the
//**parameter matrix 'm'** are same

Matrix transpose (); //Returns a matrix object that is transpose of the **calling matrix object**

Matrix addition (Matrix &m); //returns a matrix that is sum of the **calling matrix object** and
//the **parameter matrix 'm'**. Check compatibility

Matrix subtraction (Matrix &obj); //returns a matrix that is difference of the **calling matrix**
//**object** and the **parameter matrix 'm'**. Check compatibility

Matrix multiplication (Matrix &obj); //returns a matrix that is product of the **calling matrix**
//**object** and the **parameter matrix 'm'**. Check compatibility

//You must make functions for input and display output of matrices.i.e

Void inputMatrix (); // takes row-wise input from user for all cells of the matrix

Void outputMatrix (); // It prints a matrix in row-wise manner.

Main Program

- You main function must allow a user to enter two matrices and their values.
- It should display a menu that with all the functionalities available for user to select from and perform on the matrices he/she entered.
- Also menu should have an option to change the two matrices user entered earlier
- Menu should run eternally till user quits by selecting the quit option also given in the menu.

Following part is not included in lab, we shall discuss it in class. REMIND ME

Single Access Function (using return by reference)

int& access(int i, int j) //use this single function to get and set the (i,j)th element of the matrix