

National University of Computer and Emerging Sciences



Lab Manual 07 **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	21-03-2023

Department of Computer Science
FAST-NU, Lahore, Pakistan

Lab Manual 07– Class and Operator Overloading II

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.**
- **22LXXX_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: You have created the following Matrix Class in LAB 04 - LAB 06

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

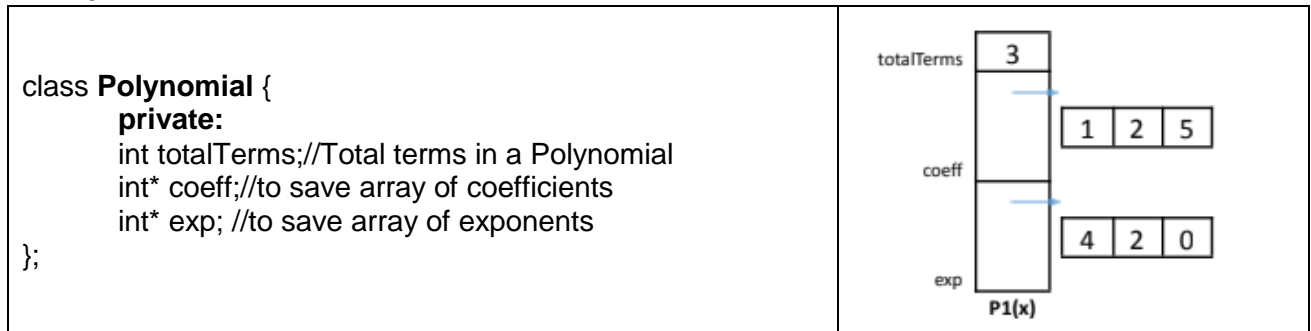
1. **Matrix(int rows, int cols): allocate memory for the matrix**
2. **~Matrix():** Destructor to deallocate memory for the matrix
3. **void input()** to initialize the matrix with user input
4. **void print()** to print the matrix
5. **void transpose()** to transpose the matrix.
6. **A Deep Copy Constructor**
7. Overloaded assignment operator (a member function)
8. A non-member overloaded plus operator to add two matrixes.
9. A non-member overloaded plus operator to add a scalar to a matrix.

Add following new public member functions in LAB 07:

1. **const int* operator[](int row) const:** Overloaded subscript operator
2. **Matrix& operator++():** Overloaded pre-increment operator
3. **Matrix operator++(int):** Overloaded post-increment operator

Also Add Usage Example of each function of your class in main() and **SUBMIT COMPLETE MATRIX CLASS with all functions from lab 04 to lab 07**

2. Exercise: A polynomial $P1(x) = x^4 + 2x^2 + 5$ have three terms: x^4 , $2x^2$ and 5. Coefficients of these terms are 1, 2 and 5 respectively while exponents are 4, 2 and 0 respectively. To work with Polynomials, a definition of class Polynomial is given below and memory configuration for P1 is shown as follows:



Your task is to complete the definition of Polynomial class such that the following main program runs successfully. Make sure that your program doesn't consume extra memory space and it should not leak any memory.

```
void main() {
    int coeff_P1[] = {1,2,5}; //Coefficients for Polynomial P1
    int exp_P1[] = {4,2,0}; //Exponents for Polynomial P1
    int coeff_P2[] = {4,3}; //Coefficients for Polynomial P2
    int exp_P2[] = {6,2}; //Exponents for Polynomial P2

    Polynomial P1(3, coeff_P1, exp_P1); //Creates P1 with 3 terms (P1 = 1x^4 + 2x^2 + 5x^0 )
    Polynomial P2(2, coeff_P2, exp_P2); //Creates P2 with 2 terms (P2 = 4x^6 + 3x^2)
    cout<<"P1 = "<<P1<<endl; //Prints P1 = x^4+2x^2+5
    cout<<"P2 = "<<P2<<endl; //Prints P2 = 4x^6+3x^2

    if(!P1)
        cout<<"P1 is zero"<<endl; // if polynomial has only 1 term and its coeff and exp are zero i.e. if p1 = 0

    if(P1 != P2)
        cout<<"P1 is Not Equal to P2"<<endl;

    //Adds P1 and P2 and saves result in P3.You may consume extra space for resultant Polynomial in Add
    function
    Polynomial P3 = P1+P2;

    cout<<"P3 = "<<P3<<endl; //Prints P3 = 4x^6+x^4+5x^2+5

    P3 = 2 + P1; //Assume P1 already has a constant term, add 2 in it.

    cout<<"P3 = "<<P3<<endl;

    .

    cout<<"+P1<<endl; //adds 1 in all the coefficient.
    cout<<P1<<endl;
    cout<<P1++<<endl; //adds 1 in all the coefficient.
    cout<<P1<<endl;
}
```