**ASSIGNMENT SET 6**

**Date: Thursday, 10/04/2020**

**Submission Deadline: Sunday 17/04/2020**

**Objective(s): Learn to implement the following features…..**

* **Binary and Unary Operator Overloading**
* **Cascaded calls to overloaded I/O (streaming) operators**
* **Overloading new and delete operator**
* **Composition (has-a) relation among classes [Ex: linked\_list class which in turn uses the node class]**

***Assignments 1 (complex class) & 5 (linked list) has been partially discussed in the lectures; you may follow the guidelines as discussed there***

**==================================================================**

1. Complex class: overload Add (+), Subtract (-), Multiply (\*), Divide (/), Conjugate (!), Compare (==, !=), Copy (=), Subscript ([ ]) – returns real for [0] and img for [1]; Input-Output (>>,<<)
2. Fraction class : Add (+), Subtract (-), Multiply (\*), Divide (/), Normalize ( unary \*), Compare (==, !=, <, >), Copy (=), Subscript ([ ]) – returns numerator for [0] and denominator for [1]; Input-Output (>>,<<)
3. Matrix class : (Don’t forget to add copy constructor) Add (+), Subtract (-), Multiply (\*), Divide (/), Invert ( ! ), Compare (==), **Copy (=), Subscript ([ ]) – check and display message for out of bound access,** Allocation/Deallocation (new, delete), Input-Output (>>,<<)
4. Set class : Don’t forget to add copy constructor) Union (+, Difference (-), Intersection (\*), Subset (<, <=), Superset (>, >=), Compare (==, !=), Input-Output (>>,<<)
5. Linked List class : Don’t forget to add copy constructor) Concatenate (+), Reverse ( ! ), Compare (==), **Copy (=), Subscript ([ ]) – check and display message when index is more than the size of the list,** **Allocation/Deallocation (new, delete) – using avail list,** Input-Output (>>,<<)
6. Convert the Assignments 4 & 5 solved above into class templates