**Lab 10: Object Oriented Analysis and Design**

**Topic: Composition, Aggregation, Association, singleton pattern**

**Objective**

Making students familiarize with the concepts of composition/aggregation/association.

**Instructions**

**Whether mentioned or not, make copy constructor and overload assignment operator**

Task 1

Create a class of Student having private data members. Make all constructors along with setter

getter. Also make a print function that can print all the information of a student.

Class Student

{

Private:

char\* name;

char\* reg\_no;

float cgpa;

Date dob;

};

Class Date

{

Private:

int day;

Page 2 of 2

int month;

int year;

};

Task 2

A cup of tea consists of Milk, Tea, Sugar and Water. Code this scenario. Call the overloaded constructors of the composed and aggregated classes from the container class.

Task 3

A personal computer has a CPU, a motherboard, a RAM, a GPU, a HDD, a Display and a Sound, along with other additional components. Each of these components is an object in itself. Thus, a Personal computer is a complex object composed of several smaller level objects. Each of the smaller objects has its own attributes and functions. For example:



Code this scenario and call the overloaded constructors of the composed and aggregated classes from the container class.

Task 4

Given the following classes, find the relationship among them as per your understanding of Composition, Aggregation, Association. You should come up with at least one real life attribute and one function for each of the class as well. Once done, code the scenario in C++:

* Teacher
* Marker
* Whiteboard
* Student
* Classroom
* University

Task 5

A car has an engine, at most 2 ACs, a handle to control the gears and a brake. If the car is manual, it has a clutch, otherwise it does not. Make three constructors of each class; a default and two parameterized. Using constructor initializer list, call the overloaded constructors of the composed and aggregated classes from the container class

Task 6

There is a system in an office which has some info associated with it. One system has single set of information so there must be one instance of SystemInfo class. Apply Singleton design pattern to this scenario so client could read same info of system from any global access point.



You have to implement info of system as shown in UML above.