Roll No.: 21BCS007 **LAB ASSESSMENT:** #03 **DATE:** 29/08/2023

NAME: Md. Ibrahim Akhtar

PROBLEM 1:

Things Learned:

1. Access Modifiers in C++ and their roles in class (through data members and member functions). <Link> <Link> <Link>

	Derived Class	Derived Class	Derived Class
Base class	**Public Mode**	**Private Mode**	**Protected Mode**
Private	Not Inherited	Not Inherited	Not Inherited
Protected	Protected	Private	Protected
Public	Public	Private	Protected

2. Basic Concept of Inheritance in C++. <Link>
3. Basics of const & static const data members in C++. <Link> <Sir's Notes>
4. Note: Learned about cmath functions (equivalent to math.h in C). <Link>

Problems Faced:

1. Access Modifiers and their scope.
2. Inheritance and its relation to objects of classes.
3. const data members was entirely new topic for me, was a bit hard to grasp.

Solutions of Questions in the PDF:

1. Access Modifiers in the class declaration are:
   - 'private' - Applied to _x and _y, making them private data members.
   - 'public' - Applied to the member functions, allowing them to be accessed from outside the class.
2. Scope of data attributes '_x' & '_y' are **private**, so their scope is limited to the class itself. They can be accessed directly from outside the class.
3. Scope of member functions is **public** during declaration. They can be accessed and called from outside the class.
4. <Written in the code / access_modifier.cpp file>
5. The member of a class can access the private members of all objects of that class.
   - Is a **TRUE** statement. Each member function has access to the private members of the specific object on which it is called. The output of the access_modifier.cpp vouches for it.

PROBLEM 2:

Things Learned:

1. Using const member functions in C++ programs.

2. Using static data members to ensure each value of the respective data members remain same throughout the program.
3. Declaring static data members inside the class, but defining them outside.
4. Declaring and defining each member functions inside the class.

Problems Faced:

1. Faced issues with declaring const member functions.
2. Use of static data members in a class and declaring them inside the class but defining them outside the class.
3. Still, having a little bit trouble with const and static data members. (Still need a little bit more practice, I guess)

Solutions of Questions in the PDF:

1. Static data members must be defined and initialized outside the class definition.
   - **TRUE**
   - This is because the static member is shared among all instances of the class and exists independently of any particular object.
2. Memory is allocated for static data members once they are defined even though no objects of that class have been instantiated.
   - **TRUE**
   - They are associated with class itself rather than with the instances of class.
3. Static member functions cannot access non-static data members.
   - **TRUE**
   - In C++, static member functions can only directly access static data members and other static member functions of the class. They do not have access to non-static (instance) data members or member functions without an explicit object instance.
4. static const data members can be initialized within the class definition but not outside it.
   - **TRUE**
5. const member functions cannot change the state of an object, i.e., they cannot change the values of any of the data members.
   - **TRUE**
   - In C++, declaring a member function as const, indicates that the function does not modify the state of the object on which it is called.
   - A const member function ensures that the object it is called on remains unmodified, preventing any changes to its data members.

PROBLEM 3:

Things Learned:

1. Using cmath, ctime, cstdlib, vector and queue libraries.
2. Using vectors.
3. Using queue/priority_queue

Problems Faced:

1. Faced issues in all of the above things.