Roll No.: 21BCS007 **LAB ASSESSMENT:** #04 **DATE:** 05/09/2023

NAME: Md. Ibrahim Akhtar

PROBLEM 1:

Things Learned:

1. Copy Constructor concept becoming clearer now. <Link> <Link>
2. Concept of Operator Overloading. <Link>
3. Overloading Arithmetic Operators as Member Functions.
4. Arithmetic Operator Overloading. <Link>

Problems Faced:

1. Sending const data in operator overloading function.
2. Using 2 const's in declaration of + operator overloading.
3. Operator Overloading Concept as a whole but with more practice and examples, I'd be able to grasp its functionality more.

Solutions of Questions in the PDF:

1. Identify the overloaded operator declaration inside the class definition.
   - The overloaded operator declaration is: "**Point operator + (const Point & p) const;**"
2. Identify the return type and argument type.
   - Return Type: Point
   - Argument Type: const Point &p
3. What do the two const keywords mean? What restrictions do they impose on the functionality of the operator?
   - The **1st const keyword** indicates that the argument "p" is a constant reference, which means that the function would not be able to modify the passed object.
   - The **2nd const keyword** indicates that the function does not modifies the state of the object on which it is called. That is, it does not modifies any member variables of the current instance.
4. Implement the operator by defining an appropriate function outside the class definition.
   - <written in code – arithmetic_operator_overloading.cpp>
5. Test your implementation by adding 2 points. Which one is that "this" object and which one is "that" object?
   - <written in code – arithmetic_operator_overloading.cpp>
6. Chain the operator and try to add more than two points. How many points can you chain?
   - < written in code – arithmetic_operator_overloading.cpp >
   - Each addition operation takes the result of the previous addition and adds it to the next point. Hence, there is **no inherent limit** to the number of points we can chain in this way.

PROBLEM 2:

Things Learned:

1. Relational Operator Overloading. <Link> <Link>

Problems Faced:

1. Relational Operator Overloading and its functionality.

PROBLEM 3:

Things Learned:

1. -

Problems Faced:

1. -

PROBLEM 4:

Things Learned:

1. Binary Operator Overloading. <Link>
2. Scaling (*) Operator Overloading.

Problems Faced:

1. Operator overloading as whole is a bit confusing topic. Need more practice.
2. Use of const arguments in operator overloading.
3. Use of const member functions i.e. using const at the last during operator overloading declaration.