Roll No.: 21BCS007 **LAB ASSESSMENT:** #06 **DATE:** 26/09/2023

NAME: Md. Ibrahim Akhtar

PROBLEM 1:

Things Learned:

1. Implementing overloaded subscript operator.
2. Implementing overloaded conversion operator.
3. Operator Overloading in C++.

Problems Faced:

1. How to implement both overloaded subscript & conversion operators.
2. Using same function for both the overloading.

Solutions of Questions in the PDF:

3. Let’s say I have the following code, will it use the type conversion operator or the overloaded subscript operator?

```
#include <iostream>

int main()
{
double arr = {1, 2};
Point p1(arr);
cout << p1[1] << "," << p1[2];

double * parr = p1;
cout << parr[1] << "," << parr[2];
}
```

- Will use the overloaded subscript operator for **p1[1]** and **p1[2]**, and it will use the conversion operator for **double *parr = p1;**.
- The conversion operator is invoked when we assign p1 to double *parr, which returns a pointer to a dynamically allocated array containing the coordinates of the Point object.

PROBLEM 2:

Things Learned:

1. Implementing conversion of point and vector and vice versa.
2. Implementing C++ stream classes structure (ostream class). <Link> <Link>

Problems Faced:

1. Had trouble while implementing the conversion of point to vector and vice versa.

Solutions of Questions in the PDF:

1. .

2. .
3. What is the output of the code? Explain why you get the output that you do.

```
Vector v1;
Point p1 (1,1);
v1 = p1;
cout << v1;
```

- The output of the above code will be: <1,1>.
- This is because the conversion operator from Point to Vector is invoked during the assignment.

PROBLEM 3:

Things Learned:

1. .

Problems Faced:

1. .