An-Najah Nation University
Faculty of Engineering

جامعة النجاح الوطنية
كلية الهندسة

Computer Engineering Department
Data Structures and Algorithms (10636211)

HW 1

ILOs [3]	**Due to 19/10/2021**	**15 points**

In this assignment, you will implement the following class.

Point3D: Represents the three-dimension coordinate systems.

Private data:

x,y, and **z**: coordinate (type **double**)
name: the name of the point (type **char***).

Required Functions:

- A constructor that takes the default values: (0.0,0.0,0.0,” ”).
- Destructor.
- get_x( ): returns x.
- get_y(): returns y.
- get_z(): return z.
- get_name(): returns name.
- set_x(double): sets the value of x.
- set_y(double): sets the value of y.
- set_z(double): sets the value of z.
- set_name(const char*): sets the value of name.
- Dist_From(Point3D): returns distance (double).

- **Operator overloading for:**
  - Addition, Add two points. Example: A=B+C (regular add the corresponding axis. For **name** you must concatenate the two names of B then C).
  - Compound assignment for addition. Example A+=B (same above for **name**).
  - Assignment operator. A=B. (allow chaining assignment).
  - Decrement. Example: A-- (No change on the name) ***friend function***.
  - cout << A // Prints the point with format as: “name: (x,y,z)”. *Note: print max two decimal places for each axis.*
  - cin>>A// Reads values of x, y, z, and name from the user.
  - Operators: >, <, ==. The comparison is based on the distance of the points from the origin (0,0,0).

In main file for testing your code:

Don't submit the same main.cpp file as in the following, another code will be tested by the teaching assistant. Make sure the following is working with your code.

You must read the data from the file that contains several points with their names.
Example:

```
# of points
#Xvalue Yvalue Zvalue TextName

3
1.0 2.0 3.0 P1
-3.0 3.0 3.0 P2
4.0 5.0 6.0 P3
```

Show the following in your code for any file with the same order and a different number of points:

- Declare an Array `Point3D my_Array[size];`
- Read the points from the given file and save them in the array.
- Print the points in the array with distance from origin greater than *d*. Read *d* value from the user.

    Ex: **d= 5**; the result:

    P2: (-3.00,3.00,3.00)

    P3: (4.00,5.00,6.00)

-
  ```
  Point3D A; // declare variable A
  A= my_Array[0] + my_Array[1];
  cout<<A; // result: P1P2: (-2.00,5.00,6.00)
  ```

-
  ```
  my_Array[0]  += my_Array[2];
  cout<<my_Array[0]; // result: P1P3: (5.00,7.00,9.00)
  ```

-
  ```
   Point3D B= myArray[0];
  A=B;
  A--;
  cout<<B<<endl;// result: P1P3: (5.00,7.00,9.00)
  cout<<A<< endl;// result: P1P2: (4.00,6.00,8.00)
  ```

-
  ```
  if(A<B)
  cout<< "Yes" << endl;
  else
  cout<< "No"<< endl;
  ```
  Result: prints Yes.