C# Labs:

TRY ALL WHAT YOU HAVE LEARNED IN THE LECTURE

1. Design 3D Point Class and Include the basic Constructor(s) [use chaining in constructors]

2. Override the ToString Function to produce this output

Point3D P = new Point3D(10,10,10);

Console.WriteLine(Point3D.ToString());

**🡺 Point Coordinates: (10, 10, 10)**

**Try to Cast Point3D to string type**

3. Read from the User the Coordinates for 2 point P1, P2

(Check the input, tryPares , Parse , Convert )

3-1 Try Array of Three Points Read and Write

4. Try to use ==

If (P1 ==P2)

Does it work properly?

**Try to override the Equals Function (from base Object)**

5. Define array of points:

Sort this array based on X & Y coordinates

6. **Implement IClonable interface to be able to clone the object.**

**To implement more than one interface:**

**class Point3D:IComparable ,ICloneable**

**7. Write a program with a Math class that has four methods: Add, Subtract, Multiply, and Divide, each of which takes two parameters. Call each method from Main ( ).**

**8. Modify the program from Exercise 7 so that you do not have to create an instance of Math to call the four methods**

9. SingleTon **Write a class that will be used by an FTP client Project**

**Your class is needed to fully describe the Network Card [Network Interface Controller (NIC)] for your machine [your machine have one and only one NIC card].**

**Prevent the other classes from declaring more than one object from NIC class.**

**NIC card must have these data: Manufacture, MAC Address, Type [Ethernet or token ring – use Enumeration here]…**

10- Define Class **Duration**

To include Three Attributes (**Hours**, **Minutes**, **Seconds**)

Override All **System.Object** Members ( ToString, Equals,GetHasCode) .

Override Equals to Work as **Value** Equality

Output from **ToString** Should follow this pattern

Hours: 1, Minutes :30, Seconds :20

Support All Required Constructors to Produce this output

Duration D1 =new Duration (1,10,15);

D.ToString();

Output: Hours: 1, Minutes :10 , Seconds :15

Duration D1 =new Duration (3600);

D.ToString();

Output: Hours: 1, Minutes :0 , Seconds :0

Duration D2 =new Duration (7800);

D.ToString();

Output: Hours: 2, Minutes :10 , Seconds :0

Duration D3 =new Duration (666);

D.ToString();

Output: Minutes :11 , Seconds :6

**Implement All required Operators overloading’s to enable this Code**

D3=D1+D2

D3=D1 + 7800

D3=666+D3

D3=D1++ (Increase One Minute)

D3 =--D2; (Decrease One Minute)

If ( D1>D2);

If ( D1<=D2);

DateTime Obj = (DateTime) D1