Day 01 Assignments:

1. Create a simple hello world EXE application and observe MSIL using ILDASM.exe
2. Create a solution file and add new project(s) into it. Try adding project from different drive or so. See solution file and project file using notepad. Also understand references.
3. Create a console application. Define variable, object into it. Show boxing, un-boxing concept. Understand stack and heap.
4. In console application, create array of integers. Use while, do-while, for, for-each loop to iterate. Break loop if value of 'i' becomes 5 (means checking if loop).
5. Create a console application. Accept input from user. Ask end user to enter some number. Check what number is being entered by end user using switch case. Using switch case, print that number in character format.
6. In console application, create an array which will store anything. i.e. integer, string, customer class object (using object array). Try fetching objects one more time back and print information in it. Understand that array has limited size.
7. Try 6th assignment using array-list object.
8. Create a console application. Create a class Person with Name, Age, Address properties. Understand concept of properties and why properties are being used.
9. In 8th assignment, write down function inside Person class to return concatenated values. Understand importance of string builder.
10. In 9th assignment, create a Customer and Employee class. Inherit them from Person class. Understand inheritance.
11. Create a simple Maths (with add, sub code) library (DLL) and refer in console based application.
12. Create console based application with class A (with Add, Sub methods) and B(with Add, Mult methods). See that both classes can't be inherited in another class C. So check multiple inheritances is not permitted in C#.

13. Now define A, B as interfaces and implement those in Class C. See that .NET permits or allows this. See .NET allows multiple inheritance using interfaces. Is it really multiple inheritance because, interfaces contains nothing like definition which can be inherited! Re-visit concept off interface to understand where it can be used. Understand client server architecture to understand interface at client and implementation of interface at server concept.
14. Understand abstract class concept. Understand it is meant for inheritance only.
15. Add normal method in abstract class. Try adding abstract method inside abstract class. Call abstract method inside normal method defined earlier. See if it is allowed and how it gets called.
16. Try implementing above abstract class inside other class. You will also have to implement abstract method. See how call goes.
17. Can abstract class contain static method? Prove it.
18. Test overloading, overriding, shadowing by defining a class called CMath and with method called Add.
19. Add DLL from 11th assignment into GAC. Make it shared library. See that private copy is not maintained in bin folder of console application referring it.