## Instructions for Subjective Assignment

# Common Instructions

The subjective assignment will consist of One problem statement for design and implementation for total 100 marks and total time given will be 3hours 30mins.

# Assignment Implementation Specific Instruction

1. Visual Studio IDE will be available on your machine.
2. Please implement the problem statement in a CPP project already provided to you.
3. Object Oriented Design and Programming is mandatory for the solution implementation.
4. **Adhere to the following instructions in Code implementation:**
   * The CPP project (assignment code) provided to participant will contain **.h and .cpp** files which have only method declarations (in **.h** files) and non-implemented function definitions (in **.cpp** files), those function definitions should be implement by participant.
   * Do not change the workspace name, project name, file name, class names and method names provided to you.
   * You need to take care of what a method takes as argument and what it must return. You have to strictly adhere to the method signatures as provided. Please do not change method names or signatures. If any discrepancy of names is present, that specific class and/or method will not be graded.
   * Some Pre-defined / User Defined classes, exception classes will already be provided in the project. You have to use the same classes and DO NOT re-create them.
   * You are free to add more classes and methods if required.
5. Sample input data/files will be provided to test your code.
6. **You have to make sure your code is compiling successfully before submitting.**

**Un-compiled code will not be accepted and graded.**

# Good Programming Practice/Specific Information

**All Programming best practices are to be followed**.

1. Usage of data structures (class, struct, enum, typedef, #define, STL containers)
2. Modular Programming (One function per task) is followed.
3. Proper handling of dynamic memory (new/ delete) for the data structures created in the program (if it is used) resulting in no memory leaks.
4. Error handling (for functionality and memory)
5. Proper indentation, readability and commenting.
6. No hardcoded values, following variable naming conventions, minimum use of global variables.
7. Putting header guards in .h files
8. Creating separate .h and .cpp files for proper declaration and definition

# Evaluation of Assignments

1. Total marks for the assignment are 100 distributed across the methods to be implemented.
2. Every method must be implemented correctly and completely as per the given specifications. **No marks will be given for partial implementation of a method.**
3. Marks will be given method wise and 100% correctness is expected along with required validations and business rules.
4. Passing criteria is **50%** for this subjective assignment.
5. Any out of the box thinking or exceptional answer (very relevant to the question) will warrant bonus marks. Final decision for the same will be with assessment center only.

Important Note:

A small part of assignment named “**PhoneBook.docs**” is shared for your reference along with the **.h** and **.cpp** files including method signatures and definitions.

**This is an incomplete problem statement. The Actual Challenge Assignment will require a lot of other features like, method / Operator Overloading, file I/O, exception handling, file handling, STL container etc…**

**For example in this problem statement, functionalities like deleting phonebook records, update phonebook records, read and write phonebook records etc. will not be included. Corresponding method signatures will also be provided which you will have to use as it is.**