<u>**Coding Marathon Problem Statement 3**</u>

Create an abstract class <u>**Exam**</u> with the following attributes

- ***examType*** of type enum class which could be either UNIT_TEST, MID-TERM or END_SEM.
- ***examCode*** of type string that stores the code of the exam to be conducted.
- ***timeDuration*** of type short that stores duration for the exam in minutes.
- A pure virtual method ***DisplayAttributes*** that accepts a parameter with no arguments.
- A parameterized constructor that accepts all parameters and creates the object.

Create a child class <u>**OnlineExam**</u> that inherits from **Exam** class and has the following unique attributes

- ***examPlatform*** of type enum class which could be MEET, TEAMS or ZOOM.
- Create an overridden method ***DisplayAttributes*** to print all data members of this class.(Note: Create your own display format.)
- A parameterized constructor that accepts all parameters and creates the object.
- A copy constructor that copies values from an existing object of this class to create the object.
- An overloaded ***+ operator*** that returns the total of ***timeDuration*** of 2 instances of OnlineExam.

Create the following functionalities in <u>functionalities.cpp</u> file

1. A function to create 5 objects of <u>**OnlineExam**</u> type. The objects must be created on the heap.
2. A function that returns the ***examPlatform*** enum value for the object whose examCode is passed as a parameter.
3. A function to print the ***timeDuration*** value of first N objects in the container, where N is passed as a parameter to the function.
4. A function that returns true or false based on whether all instances have ***timeDuration*** over 60 or not.
5. A function to find and return pointers to all instances whose ***timeDuration*** is above a threshold value passed as a parameter. The pointers to instances to be returned must be stored in an array.
6. A function that takes 2 arguments, both pointers to instances of <u>**OnlineExam**</u>. The function must return the result of argument1 + argument2 (by using overloaded + operator in the class).
7. A function to delete all allocations made on the heap.

<u>**Create a Main.cpp file with code to demonstrate each function from the functionalities.cpp file.**</u>