C++ and OOP – Final Project

1. Create a class – Device with the following members:
    - Attributes:
        - Id – generated automatically for every device
        - Name
        - IRQ number
        - Base address
    - Operations:
        - Constructor
        - Init – to initialize the hardware (dummy implementation)
        - Log – to write a message/error to log
2. Derive RTC class from Device
    - Attributes:
        - Match time - integer
        - Enable/disable – Boolean
    - Operations
        - Constructor
        - Init
        - Set_time (dummy or not)
        - Get_time (dummy or not)
3. Derive UART class from Device
    - Attributes:
        - Baud rate – enum with options
        - Parity – Boolean
        - Data bits – integer
        - Stop bit – enum with options
    - Operations:
        - Constructor
        - Init
        - Send (dummy)
        - Receive(dummy)
4. Create a class DeviceManager
    - Add an array of pointers to devices
    - Add a function CreateDevice
        - Input: enum devicetype
        - Output: pointer to the created device (create the object dynamically and add it to the array)
    - Add Init function to initialize all the devices
5. Add a main function and test your work

- Notes:
    - Use const for functions and parameters
    - Use explicit for conversion constructors
    - Divide the project to cpp/hpp files for each class

Part 2

1. make sure that the only possible way to create devices is using the DeviceManager class
2. Add another device class for Timer and change/add the code to support it
3. Change the DeviceManager class to singleton
4. Create a class Logger as abstract class with abstract method Log(string s)
5. Derive 2 classes from logger:
    a. FileLogger – write the log to file
    b. NetLogger – write the log to network
6. Implement the functions as dummy operations
7. Add the device classes a logging support