# Assignment: Design a Battery Management System

In this programming assignment, you are tasked with designing a Battery Management System (BMS) for a robot. Your design should demonstrate your proficiency in C++ and your understanding of battery management concepts.

# Requirements:

The system will consist of two main classes: `***Battery***` and `***BMS***`. The ability to accept inputs from either the console or a file will be considered a bonus.

## 1. Battery Class

Create a `***Battery***` class that models a real-world battery. It should be instantiated with a specified capacity, internal resistance, and minimum charge threshold. The class should keep track of voltage and temperature, but it should privately keep track of the current charge level. It should not provide a function to get the current charge level.

If the battery charge level goes below the minimum charge threshold, the battery should no longer accept the `***charge***` command.

The `***Battery***` class should include the following methods:

* **charge(int power)**: This function should increase the battery's charge by a certain amount of power and should appropriately affect the temperature of the battery.
* **discharge(int power)**: This function should decrease the battery's charge by a certain amount of power and should appropriately affect the temperature of the battery.
* **cool()**: This function should decrease the temperature of the battery.
* **getVoltage()**: This function should return the current voltage of the battery.
* **getTemperature()**: This function should return the current temperature of the battery.

Each command sent to the battery should be assumed to last for one second.

## 2. Battery Management System (BMS) Class

The `***BMS***` class should use an instance of the `***Battery***` class to perform operations and monitor the battery. It should include the following methods:

* **chargeTo(int percentage)**: This function should charge the battery to a certain percentage of its total capacity.
* **driveForward(int power)**: This function should move the robot forward using a certain amount of power.
* **turnLeft(int power)**: This function should turn the robot left using a certain amount of power.
* **turnRight(int power)**: This function should turn the robot right using a certain amount of power.

The `***BMS***` class should also monitor the battery's temperature to make sure it does not overheat, and it should estimate the State of Charge (SOC) to make sure the battery does not get below the minimum energy level. It should print logs for all important actions and state changes, including SOC and faults.

**Bonus:** Expand the list of commands to include additional functions.

## 3. Main Program

Your main program should instantiate both the `***Battery***` and `***BMS***` classes and provide the input commands to the `***BMS***`. Your program should be able to accept commands from the console.

**Bonus:** Design your program to also accept command inputs from a file.

# Evaluation Criteria:

Your assignment will be evaluated on the following:

* **Correctness:** Your code should correctly implement the assignment requirements.
* **Code Quality:** Your code should follow good coding practices, including readability, maintainability, and efficiency.
* **Error Handling:** Your code should handle edge cases and potential errors.
* **Documentation:** Your code should include comments explaining your logic and decisions where necessary. You should also document any design decisions or explanations – either in the code or separately. For example, how did you determine the battery values for minimum charge level or internal resistance?

# Submission:

Please submit solution as a single ZIP file including your source code (as a `.cpp` files) along with any appropriate documentation/readme files including instructions for building & running your submission. If your program accepts inputs from a file, also submit a sample input file.

Good luck, and happy coding!