POLYMORHISM

You will continue to develop the Computer API that you developed in the coding exercise of previous Inheritance section. You will keep the class hierarchy as same as in the previous section. You will add new features to existing classes and a new class TestComputer. You can find what will be added under the "New Development Requirements" section in this document.

This API is expected to have the following class hierarchy with the specified members.

Computer Class
This is the top most super class in the hierarchy.
Desktop Class
This is a sub-class of the Computer class.
Laptop Class
This is a sub-class of the Computer class and no body can extend this class.

Common Properties Of Three Classes

serialNumber: It must be given automatically to each created Computer object starting from 10001 and it can never be changed by the Computer object user. It can be read by the object user. It can not be accessed directly from other classes.

cpuFrequency: It can be modified and read by the object user. It can not be accessed directly from other classes. It can store decimal numbers between 1 and 10. In the case of attempting to assign other values it should be set to 1. This value indicates the cpu frequency in terms of GHz
color: This field can be black, white. It can be modified and read by the object user. If the object user attempts to set other colors than black, white it must be set to default color which is white. This color field must be stored in uppercase letters even if the object user attempts to set in lowercase. It can not be accessed directly from other classes.
memory: It can be modified and read by the object user. It can not be accessed directly from other classes. It can store digital numbers that are multiple of 4 between 4 and 32, otherwise it should be set to 4. This value indicates the memory amount in terms of GB.

Common Methods Of Three Classes
Getters And Setters: All the getters and setters must be accessible and overrideable from **only** classes in the same package and sub-classes in different package.
start: This method should start the computer and it should print different messages according to the type of computer.
If the computer is a laptop it should print "Laptop is booting up"
If the computer is a desktop it should print "Desktop is booting up"

shutdown: This method should shutdowns the computer and it should print different messages according to the type of computer.
If the computer is a laptop it should print "Laptop is shutting down"
If the computer is a desktop it should print "Desktop is shutting down"
Rule: The shutdown method of the subclasses Desktop and Laptop can not be overridden by the subclasses of the Desktop and Laptop classes.

Properties That Are Peculiar to Only Laptop
batteryLife: It can be modified and read by the object user. It can not be accessed directly from other classes. It can store digital numbers between 1 and 5. Attempting to assign other values it should be set to 2. This value indicates the battery life in terms of hours.

Properties That Are Peculiar to Only Desktop
caseType: This field can be FULL TOWER, MID TOWER and MINI TOWER. It can be modified and read by the object user. If the object user attempts to set other than FULL TOWER, MID TOWER and MINI TOWER it should be set to default case type which is MID TOWER. This case type field must be stored in uppercase letters even if the object user attempts to set in lowercase. It can not be accessed directly from other classes.

Rules
Also there should be a logical order of calling methods.

1. If the computer is already started you should not call start again. Otherwise it will give message "Computer is already started ".

2. To call shutdown, the computer should be started otherwise it will give message "Computer is already shut down"

Constructors
There should be only one constructor per class like below.

Computer(String color, float cpuFrequency, int memory)
Laptop(String color, float cpuFrequency, int memory, int batteryLife)
Desktop(String color, float cpuFrequency, int memory, String caseType)

The constructors of the sub-classes should invoke the super class constructor. There should not be lines that repeat the lines in the super class constructor.

New Development Requirements

1. Add a new method to the Laptop class. This method has the signature as below and just print "*Battery is being charged*". public void *chargeBattery()*

2. Add a toString() method to the appropriate class that returns seraial number, color, cpu frequency and memory amount for all types of object in the following format.
   Computer{*Serial Number:<Serial Number>, Color:<color>, Frequency:<cpuFrequency>, Memory:<memory>}*
   Example: Computer{Serial Number:100, Color:WHITE, Frequency:3.2, Memory:16}

3. **TestComputer Class:** This test class has only one method as defined below
   void startAll(Computer[] computers): This method takes array that can have Computer, Desktop and Laptop types of elements. Method should call toString() and start() method of each element. If the element is a laptop it should call chargeBattery() method before calling start() method.