Practical Object oriented design (Strategy & Observer)

Solve the following problems.

Problem1: Computing Average Score

We want to implement a class to represent the student scores from a class they are enrolled in. The class must provide at least the following methods:

```
void addAssignmentScore (double as); //0 or more assignments
void addExamScore (double es); //0 or more exams
double getAverage(); //the final class average
```

The algorithm to compute the average can be selected at runtime. It also must be possible to add new algorithms to compute the average to the program without modifying the above class. Your task is to design such class that satisfies the above requirements.

Use the following two algorithms for computing the average in your implementation:
A. The Assignment average contributes 40%, and the Exam average contributes 60% to the final class average.
B. Use the same percentages as the first algorithm, but first drop the lowest Assignment score.

Problem2: Facebook

In principle a social network service focuses on building online communities of people who share interests and/or activities, or who are interested in exploring the interests and activities of others. Facebook support groups that people can join. Each group has a title, administrative members, a group type (open/ closed), and a list of related groups. If somebody writes on the wall page of the group, the information is broadcasted to all the members and it is visualized in the news feeds of the members. Users should be able to join a group as well as leave a group if they get bored. Once a user has joined a group it will automatically receive any updates that are published on the wall. Which design pattern is the most appropriate to handle this basic functionality of such a Facebook group? Provide the Design as well. The following test code provides details about the group operation:

```
public static void main(String[] args) {

        System.out.println("Testing the Facebook Application");
        //Create a group
        FacebookGroup dp = new FacebookGroup();

        //Create users
        FacebookUser user1 = new FacebookUser(dp, "XYZ", 23);
        FacebookUser user2 = new FacebookUser(dp, "ABC", 20);
        FacebookUser user3 = new FacebookUser(dp, "AXY", 25);

        //Add users to the newly created group
        dp.addUser(user1);
        dp.addUser(user2);
        dp.addUser(user3);

        //write something on the wall
        dp.setState("Hello World");

        //users can also write on the wall
        user1.writeOnTheWall("Hi");
}
```

Problem3: Groupon

Groupon webpage shows deeply discounted offers from different business located in the city where you are living. As an example, it might show a beauty salon offer that has a 50% discount on women’s haircut. You do nothing if you are not interested. If you like the Groupon deal, you select “buy” before the expiration date and your credit card is charged. If enough people select the deal, you will be sent a link for your rebate coupon at the end of the offer. Once you receive your coupon you can claim it at the specified store, usually within a period of time before it expires as noted on the website. If enough people do not join for that deal, the Groupon is cancelled and no one gets it and in this case you will get a refund for your money. In case that you have changed your mind you can also cancel an already made order and you will get a full refund. Provide a design for the specified functionality.