Q2

1)

a) Attach additional responsibilities to an object dynamically. Decorators provide a flexible alternative to subclassing for extending functionality.

b) the pattern is right for this situation because we want to add to the original profile more information dynamically and print it in the end.

c)

Diagram

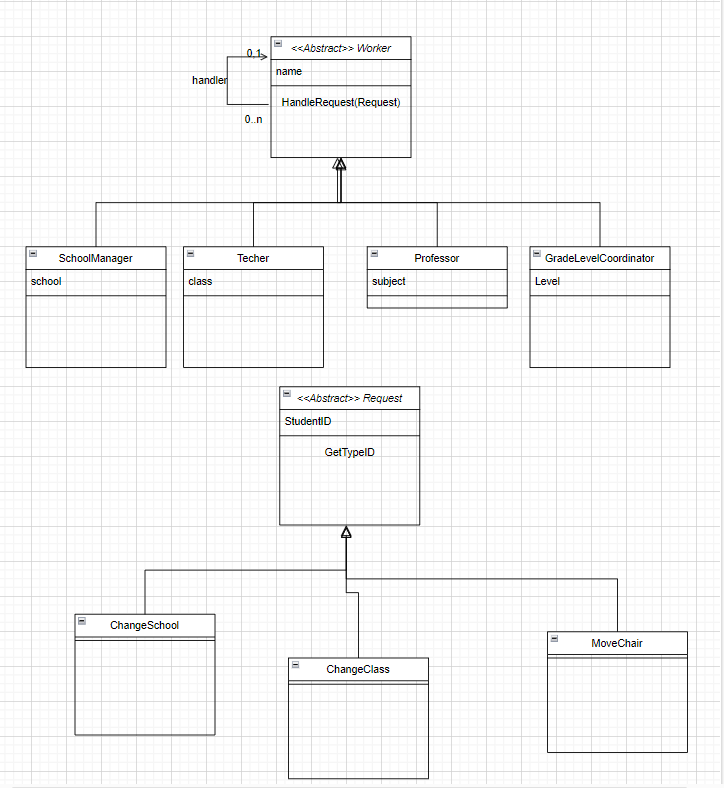
Description automatically generated

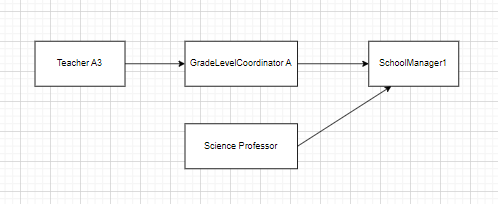
2)

A. lets imagine a school system, where the client may be a student, and the handlers are workers (teacher, ect’).

And we want the student to send request only to his teacher, and if the teacher can handle it, she will otherwise she will pass it on and on…

B. our system may look this way :





Similarity :

1. In Both the “main” class, extends and hold itself
2. Both pattern are meant to solve our problems only at runtime (cannot know the concrete object)

Difference:

1. Decorator meant to add functionality to an object, while chain meant to pass responsibility between objects
2. In chain pattern every object can hold another chain, while in decorator, the “original” classes can not hold a different occurrence of themselves