Question 1:

Assume that we would like to model a personal computer (PC). A PC consists of a cabinet, which includes a chassis. A chassis on its turn is composed of a bus, a floppy disk drive, a memory unit, a CPU, and a power supply. A bus incorporates a network card. We would like to treat all the equipment components in a uniform way. The basic operations needed are netPrice and powerConsumption, which return the net price and the power consumption of each component respectively. In addition, each component may have its own specific semantics.

Which design pattern would be suitable for this problem?

Solution:

We've been assigned the task to create a PC consisting of various modules which in turn hold other modules. To begin with, we can consider the modules as objects representing a class. We also require to "*treat all the equipment components in a uniform way*". This implies all components must inherit from the same class/interface.

This modular design calls for the **builder pattern**. It allows us to create complex data structures with each member object having the ability to "*have its own specific semantics*".