**Design pattern assignment**

## 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.

**Answer**

To compose objects into tree structures as we are using a network card to represent the tree structure and connect other components in the chassis.

Since we are handling multiple aspects of assembling a PC a synchronised assembling and design which is Composite Pattern is an appropriate pattern to be used. The netPrice and powerConsumption are to be written as interface methods.

## Question 2

Clearly, we can have different configurations of PCs, for example a tower PC and a desktop PC. Furthermore, each type of PC may require specific components. We may have, for instance, specific cabinets and chassis for tower PCs and desktop PCs. Consequently, we assume that we will have a family of different products. This is illustrated in the next class hierarchies for cabinet and chassis.



There are actually two concerns here. First, we would like to ensure that a computer configuration always consists of parts that fit together. Second, we would not like to be bothered with details of the creation of computers.

**Answer**

Decorator pattern allows to add new functionality an existing object without altering its structure. Since we do not want to bother with other details we can use the following pattern.

## Question 3

In the future it is expected that class *ComputerEquipment* will be extended with additional operations. The present structure of a computer equipment is assumed not to change. It is also not desired to ‘pollute’ all the existing component classes with new operations.

**Answer**

The Transfer Object pattern is used when we want to pass data with multiple attributes in one shot , as we do not intend to pollute with extra changes to the existing pattern.

## Question 4

A journal on PCs would like to investigate some properties of different PCs, e.g. the price, performance, etc. The results need to be sorted in a list. There are several sorting algorithms for sorting all the PCs depending on specific characteristics. These sorting algorithms behave differently with respect to the time and space performance. Therefore, it is required to select and switch to a different sorting algorithm at run-time.

**Answer**

Iterator pattern is very handy design pattern to used to iterate through the data at run time. This pattern is used to get a way to access the elements of a collection object in sequential manner without any need to know its underlying representation.