## **OOAD Assignment 6**

## **Problem 6: Questions: Decide which Pattern**

What pattern best meets each of these situations below?

1. The model stores a collection of blocks. Blocks can be medal or wood; can be painted, sanded, or chrome plated; and can sometimes be radioactive or magnetized. What design pattern would allow the system to easily add new types of blocks without changing existing code?
   * Explain how two design *principles* apply to this design.
2. The model for a game stores robot. The robot navigates a maze that has obstacles. While playing the game, the robot can be upgraded with new parts that change its abilities like speed, weapons, and shields. Which design principle allows the robot object to change its behaviour at runtime in flexible ways.
   * Explain how two design *principles* apply to this design.
3. The model stores a phone number, and the UI (a keypad) allows the user to enter digits one at a time. A different part of the UI wants to respond to each key being entered; however a) the two parts of the UI should be decoupled, and b) the model should be decoupled from the UI (model knows nothing about the UI). Which pattern would allow this?
   * Explain how two design *principles* apply to this design.
4. A library supports recursively searching a directory for files. It allows the client code to provide it an object to filter the results. For each file which the library finds, it will ask the filter object if that file should be accepted or rejected. (See Java's [FileFilter](https://docs.oracle.com/javase/8/docs/api/java/io/FileFilter.html))
   * Explain how two design *principles* apply to this design.

A)

1) Factory

\* Open close principle.

\* Single responsibility principle.

2) Decorator (in run time we can type cast and change the program)

\* Decoupling and Separation of Concerns

\* Single Responsibility Principle

3) Observer Pattern (PubSub)

\* Dependency inversion

\* Liscop substation

4) Strategy Pattern

\* Open close principle.

\* Dependency Inversion Principle.