Problem Set 2

(10 points) Cohort Exercise 2:

Given the Enemy Ship example: EnemyShipDemo.java, refactor the code using Factory Design Pattern.

(5 points) Cohort Exercise 3:

Complete the class diagram for the Subscriber-switch example that uses Factory Design patterns.

(10 points) Cohort Exercise 4:

Given the pizza store example: PizzaStore.java, refactor the code using Factory Design Pattern.

(5 points) Cohort Exercise 5:

Complete the class diagram for the Telephone pricing scheme example that uses observer patterns.

(20 points) Cohort Exercise 6:

Assume a social media that displays a number of posts. Anyone who subscribes to a post is notified immediately when the post is edited or some other subscriber comments to the post. Subscribers only receive notification for the posts (s)he was subscribed to. Any subscriber can leave from his/her subscription at any time. Design and implement such a social media using observer patterns. For the design, draw the class diagram. Your implementation should include at least the following function: make a new post, subscribe to a post, edit a post (only allowed for the subscriber who posted initially), comments on a post and leave from a post.

(10 points) Cohort Exercise 7:

Complete DecoratorDemo.java by adding the ham topping.

(10 points) Cohort Exercise 8:

Assume that you are programming a simulator with multiple interacting robots. There are three types of robots: aggressive, defensive and normal. Apply strategy design pattern to complete the code skeleton: RobotGame.java.

(10 points) Cohort Exercise 9:

Given the ugly VisitorOriginal.java, study the Visitor Pattern by yourself, and apply the pattern to improve the code.

(10 points) Homework Question 1:

Write a Java program for a tic-tac-toe game. Two users take turns to make a move. The game is over after a winner is decided. Discuss the design patterns used in your code.

(10 points) Homework Question 2:

Write a Java program for a sequential election vote casting application as follows: There are two candidates A and B contesting an election. There are five electorates and each electorate can cast their vote only once and for only one of the two candidates (A or B). The five electorates cast their votes, i.e. a character 'A' or 'B', one after another. The winner is the candidate who gets the maximum number of votes. Discuss the design patterns used in your code.