FACTORY & OBSERVER ACTIVITY (5 PTS)

FULL NAMES:

1. Create a UML class diagram for part of the preliminary design of a Maps app.

   The Maps app updates its view whenever the phone's location changes. The Maps app learns of location changes from a LocationMonitor. The app can select one of several LocationMonitors to use: GpsMonitor, WifiLocationMonitor, or CellTowerMonitor. All LocationMonitors can report their current estimate of the phone's location.

   - Since other types of location monitors may be added in the future, use a simple **factory pattern** for creation of concrete location monitors.
   - To reduce coupling, use the **observer pattern** to allow the location monitor to inform the Maps app when the location changes.

2. Create a UML class diagram for part of the preliminary design of a fantasy video game.

The video game user interface is handled by a class called GameView. GameView updates the screen based on the GameModel. The GameModel manages the creation, movement, actions, and destruction of various game Entities.

Entities include dragons, knights, wolves, and zombies. All Entities have a name, a type name, a sprite, a location, and the ability to move or attack.

- Since new entities are likely to be added in the future, use the **simple factory pattern** for entity creation.
- Whenever an Entity is added, moves, attacks, or is removed, the GameModel should inform the GameView via the **observer pattern**.