ECE 651

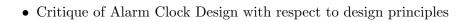
Lecture 6: Design Patterns Notes Outline

• OO Design Review

• UML

```
public class Time {
  private int hour;
  private int minute;
  private int second;
  public void addSecond() {
      second++;
                                                        public class Clock {
                                                          private Time now;
  public void set(int hour, int minute, int second) {
     this.hour = hour;
                                                          public Time getTime() {
     this.minute = minute;
                                                            return now;
     this.second = second;
                                                          public void tick() {
  public int getHour() {
                                                             now.addSecond();
     return hour;
                                                          }
                                                        }
  public int getMinute() {
     return minute;
  public int getSecond() {
     return second;
   }
}
```

•	Aggregation, Composition, Association
•	Potential problems/pitfalls with current $\operatorname{Clock}/\operatorname{Time}$ relationships?
•	Restricting actions through an interface
•	Think, Pair, Share: Design an Alarm Clock



• Alarm Clock Design Improvements

•	Observer Pattern
•	Design Patterns: "Design Patterns: Elements of Reusable Object-Oriented Software" by Gamma, Helm, Johnson, and Vlissides ("Gang of Four").
•	More features for Alarm Clock: Smart Toaster
•	Adapter Pattern