1. Week 12 Review --OO Design Principles: Cohesion & Coupling
2. Name some good OO design principles.

1. What is coupling?

Coupling is the degree to which one class knows about another class. OO designs dealing with multiple classes (such as thru shared interfaces, inheritance situations, etc.) can be therefore viewed defined as either “loosely” coupled classes or “tightly” coupled classes.

An example of a loosely coupled scenario would be: If the only knowledge that class A has about class B, is what class B has exposed through its interface, then class A and class B are said to be loosely coupled.

Give an example of a tightly coupled scenario.

1. What is Cohesion?

The term cohesion is used to indicate the degree to which a class has a single, well-focused purpose. The more focused a class is, the higher its cohesiveness. A pseudo-code example

for an app of this follows:

class SalesReport {  
void connectToDb(){ }  
void generateSalesReport() { }  
void saveAsFile() { }  
void print() { }  
}

What would be some key benefits of having a cohesive class?

How would you take a more cohesive role in your coding of the app where you would break up the above pseudo-code into parts (classes) to satisfy a manager who may want take get more involved into different aspects of the application such as inventory reporting, revenue projection reports, printer selectivity, database selectivity, file saving options, etc.?