**Discussion Questions to Java Encapsulation Sarah Redmon**

**INF 260**

1. Encapsulation is the ability to control or restrict an object’s accessibility to both class data and methods. This is to keep the code from any external harm and keep it all together.
2. The four different access modifiers in Java are public, private, protected, and no modifier.

* **Public**: This is accessible by the current class, same package, & by both subclasses and classes in other packages. This is the least restrictive.
* **Protected**: This is accessible by all except for classes located in other packages.
* **No Modifier**: This is accessible by all except for subclasses and classes located in other packages.
* **Private**: This is only accessible by the current class. This is the most restrictive.

1. Method parameters should be validated before they are used in a method to prevent any out of bounds errors and to validate that the values are correct and useful to the program.
2. Private data members are more secure than public ones because private is only accessible by the current class, unlike public where the data members would be accessible by all the classes.
3. Private data members are more secure than default ones because, as stated above, private is only accessible by the current class, unlike default (no modifier) where the data members are accessible by subclasses and classes.
4. An accessor should not be routinely created for each instance variable because vulnerabilities can be created as that data member can be exposed to an outside class.
5. A mutator should not be routinely created for each instance variable because, as stated above, vulnerabilities can be created as that data member can be exposed to an outside class.
6. There are many changes I decided to do to Card.java & CardGame.java. First, I decided to change the data members from default to private because private is the most secure of the two. Second, I decided to add in accessor methods; since the suit and face are now private, they must be accessed through a public method. This is an important concept to Encapsulation. Third, I decided to delete the mutator methods since they are not necessary and could create vulnerabilities if left in. Lastly, I added in input validation to ensure there are no out of bounds errors and leave the data less vulnerable to external harm. I changed the methods in CardGame.java to add in the accessor methods since suit and face are private and need an accessor to access them publicly.