/*
THIS CODE WAS MY OWN WORK, IT WAS WRITTEN WITHOUT CONSULTING
CODE WRITTEN BY OTHER STUDENTS OR COPIED FROM ONLINE RESOURCES.
Philip Cardozo
*/

In a file submitted as DesignQuestions.pdf include the answers to the following questions:
1. Identify where in your code you have used the concepts of method overloading and method overriding.
overloading:
ShapeTester:
longerPerim(Circle, Rectangle) - lines 14; longerPerim(Rectangle, Circle) - line 21;
largerArea(Circle, Rectangle) - line 26; largerArea(Rectangle, Circle) - line 31

Sphere:
setCenter(double x, double y, double z) (line 28)

Circle :
the constructors: Circle() - line 16; Circle(double r) - line 24; Circle(double r, double x, double y) - line 31

overriding:
Rectangle :
equals(Object obj) overrides Object.equals()

Circle:
toString() overrides Object.toString() and equals(Object obj) overrides Object.equals()

Sphere :
getCenter() overrides Circle.getCenter() and getArea() overrides Circle.getArea()

2. Explain why the methods in ShapeTester.java are static and what this means.
Static refers to the class not to an individual instance. When a method is static, it can be used wihout coding an object of the class.

They are static because they support functions of Circle and Rectangle and are not connected to a specific object of ShapeTester. In other words, they do not store information, they are tools to be called using ShapeTester.#####() without having to code an object first. They also cannot access variables or methods that are not static, because they work with the inputs given to them.

3. Describe which instance variables of Circle are accessible in Sphere and why. Can code outside of these two classes directly access any of these variables?
The protected doubles (radius, x, and y): protected variable are accesible in Sphere becasuse of the protected characteristic of accessing subclasses or what's in the same package, that's why these variables can't be accessed directly by outside code.