1. **Identify where in your code you have used the concepts of method overloading and method overriding.**

- **Method Overloading:**
  - In `Circle.java`, the constructor is overloaded with different parameter lists:
    - `Circle()` - default constructor.
    - `Circle(double r)` - initializes the circle with a radius.
    - `Circle(double r, double ex, double why)` - initializes the circle with radius and coordinates.
  - In `Sphere.java`, the `setCenter` method is overloaded:
    - `setCenter(double ex, double why)` - inherited from `Circle`.
    - `setCenter(double x, double y, double z)` - sets 3D coordinates for the sphere.
- **Method Overriding:**
  - In `Circle.java`, the `toString` and `equals` methods override the methods from `Object` class.
  - In `Sphere.java`, the `getCenter` and `getArea` methods override those from `Circle` to account for 3D properties and sphere surface area.

2. **Explain why the methods in ShapeTester.java are static and what this means.**

- The methods in `ShapeTester.java` are static because they operate on objects passed as parameters rather than requiring an instance of `ShapeTester`. This means these methods belong to the class itself rather than any particular object, allowing them to be called using the class name, e.g., `ShapeTester.isLarger(circle, rectangle)`, without creating an instance of `ShapeTester`.

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 `Sphere` class can access the `radius`, `x`, and `y` variables from `Circle` because they are declared as `protected`, which allows access within the same package and by subclasses.
- Code outside of these two classes cannot directly access these variables if it's outside the package, as `protected` restricts access to the subclass and package-level scope. To access these variables externally, public getter and setter methods must be used.