Shape Class

```
abstract class Shape {
	int x, y, z;

	public abstract boolean isSimilar(Shape object);

	public abstract boolean isCongruent(Shape object);

	public void position(int x_coordinate, int y_coordinate, int z_coordinate) {
		this.x = x_coordinate;
		this.y = y_coordinate;
		this.z = z_coordinate;
		System.out.println("The position is updated to (" + x + "," + y + "," + z +
")");
	}
}
```

Circle Class

```
public class Circle extends Shape {
	int radius;
	public Circle(int radius) {
		this.radius = radius;
	}

	@Override
	public boolean isSimilar(Shape object) {
		if (this.getClass() == object.getClass())
			return true;
		else
			return false;
	}
	@Override
	public boolean isCongruent(Shape object) {
		if (this.getClass() == object.getClass()) {
			Circle c2 = (Circle) object;
			if (this.radius == c2.radius)
				return true;
			else
				return false;
		} else
			return false;
	}
}
```

Rectangle Class

```
public class Rectangle extends Shape {
	int length, breadth;
	public Rectangle(int length, int breadth) {
		this.length = length;
		this.breadth = breadth;
	}
```

```
        @Override
        public boolean isSimilar(Shape object) {
                if (this.getClass() == object.getClass()) {
                        Rectangle rect2 = (Rectangle) object;
                        if ((this.length / this.breadth) == (rect2.length / rect2.breadth)) {
                                return true;
                        } else
                                return false;
                } else
                        return false;
        }

        @Override
        public boolean isCongruent(Shape object) {
                if (this.getClass() == object.getClass()) {
                        Rectangle rect2 = (Rectangle) object;
                        if (this.length == rect2.length && this.breadth == rect2.breadth)
                                return true;
                        else
                                return false;
                } else
                        return false;
        }
}
```

Square Class

```
public class Square extends Shape {
        int side;

        public Square(int length) {
                this.side = length;
        }

        @Override
        public boolean isSimilar(Shape object) {
                if (this.getClass() == object.getClass())
                        return true;
                else
                        return false;
        }

        @Override
        public boolean isCongruent(Shape object) {
                if (this.getClass() == object.getClass()) {
                        Square sq2 = (Square) object;
                        if (this.side == sq2.side)
                                return true;
                        else
                                return false;
                } else
                        return false;
        }

}
```

Driver Class

```
public class DriverClass {
	public static void main(String args[]) {
		Rectangle r1 = new Rectangle(10, 5);
		Rectangle r2 = new Rectangle(2, 1);
		Square s1 = new Square(5);
		Square s2 = new Square(10);
		Circle c1 = new Circle(5);
		Circle c2 = new Circle(5);
		System.out.println("r1 and r2 are " + ((r1.isSimilar(r2))? "":"not ") +
"similar");

		System.out.println("s1 and c2 are " + ((s1.isSimilar(c2))? "":"not ") +
"similar");

		System.out.println("c1 and c2 are " + ((c1.isCongruent(c2))? "":"not ") +
"congruent");

		System.out.println("s1 and s2 are " + ((s1.isCongruent(s2))? "":"not ") +
"congruent");

	}
}
```