Two Dimensional Figure Interface

```
public interface TwoDimensionFigure {//Interface declaration
	public int getArea();

	public int getPerimeter();
}
```

Three Dimensional Figure Interface

```
public interface ThreeDimensionFigure {//Interface declaration
	public int getSurfaceArea();

	public int getVolume();
}
```

Abstract 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 are 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 implements TwoDimensionFigure {
//Look at the keywords used in the above line
	int length, breadth;

	public Rectangle(int length, int breadth) {
		this.length = length;
		this.breadth = breadth;
	}
//Look carefully at how the interface methods are overridden here
	@Override
	public int getArea() {
		int area = length * breadth;
		return area;
	}

	@Override
	public int getPerimeter() {
		int perimeter = 2 * (length + breadth);
		return perimeter;
	}

	@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 implements TwoDimensionFigure {
        int side;

        public Square(int length) {
                this.side = length;
        }
/*Look carefully at how the interface methods are overridden here, and also the keywords
used in the first line*/
        @Override
        public int getArea() {
                int area = side * side;
                return area;
        }

        @Override
        public int getPerimeter() {
                int perimeter = 4 * (side);
                return perimeter;
        }

        @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;
        }

}
```

Cube Class

```
public class Cube extends Shape implements ThreeDimensionFigure {
	int side;

	public Cube(int length) {
		this.side = length;
	}
/*Look carefully at how the interface methods are overridden here, and also the keywords
used in the first line*/
	@Override
	public int getSurfaceArea() {
		int surfaceArea = 6 * (side) * (side);
		return surfaceArea;
	}

	@Override
	public int getVolume() {
		int volume = (side) * (side) * (side);
		return volume;
	}

	@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()) {
			Cube cube2 = (Cube) object;
			if (this.side == cube2.side)
				return true;
			else
				return false;
		} else
			return false;
	}

}
```

Cuboid Class

```
public class Cuboid extends Shape implements ThreeDimensionFigure {

	int length, height, breadth;

	public Cuboid(int length, int height, int breadth) {
		this.length = length;
		this.height = height;
		this.breadth = breadth;
	}
/*Look carefully at how the interface methods are overridden here, and also the keywords
```

```
used in the first line*/
	@Override
	public int getSurfaceArea() {
		int surfaceArea = 2 * ((length * height) + (breadth * height) + (length *
breadth));
		return surfaceArea;
	}

	@Override
	public int getVolume() {
		int volume = (length) * (height) * (breadth);
		return volume;
	}

	@Override
	public boolean isSimilar(Shape object) {
		if (this.getClass() == object.getClass()) {
			Cuboid cuboid2 = (Cuboid) object;
			if ((this.length / this.breadth) == (cuboid2.length / cuboid2.breadth)
					&& (this.breadth / this.height) == (cuboid2.breadth /
cuboid2.height)
					&& (this.height / this.length) == (cuboid2.height /
cuboid2.length)) {
				return true;
			} else
				return false;
		} else
			return false;

	}

	@Override
	public boolean isCongruent(Shape object) {
		if (this.getClass() == object.getClass()) {
			Cuboid cuboid2 = (Cuboid) object;
			if (this.length == cuboid2.length && this.breadth == cuboid2.breadth
&& this.height == cuboid2.height)
				return true;
			else
				return false;
		} else
			return false;
	}

}
```